

**UNIVERSITÉ DE LILLE**

École doctorale **Sciences de l'Homme et de la Société**

Centres de recherche

Centre Interuniversitaire de Recherche en Éducation de Lille

(CIREL - ULR 4354)

Centre de Recherche en Informatique, Signal et Informatique de Lille

(CRISTAL - UMR 9189)

Thèse présentée par

**Marielle LÉONARD**

Soutenue le 13 novembre 2024

Discipline **Sciences de l'éducation et de la formation**

Spécialité **Didactique de l'informatique**

**Approche didactique et instrumentale  
de la pensée informatique :  
focus sur le concept de motif  
- Annexes -**

**Composition du jury**

<i>Rapporteurs</i>	Béatrice DROT-DELANGE	professeure à l'université Clermont Auvergne
	Sébastien GEORGE	professeur à l'université du Mans
<i>Présidente</i>	Clarisse DHAENENS	professeure à l'université de Lille
<i>Examinatrices</i>	Mariam HASPEKIAN	professeure à l'université Paris Cité
	Sandra NOGRY	MCF HDR au Cergy-Paris Université
<i>Invitée</i>	Janine ROGALSKI	université Paris Diderot
<i>Directeurs de thèse</i>	Cédric FLUCKIGER	professeur à l'université de Lille
	Yvan PETER	professeur à l'université de Lille
	Yann SECQ	MCF à l'université de Lille



# **Annexe 1 : description de la base de données**

## Annexe 1 : description de l'extraction de la base de donnée

Après requêtes SQL et export au format .csv, une extraction de données de la base bebras.chticode comporte une table avec les champs suivants (Figure 1) :

- ID : n° d'enregistrement dans la base de donnée
- password : identifiant de la session de l'utilisateur
- groupID : identifiant du code de groupe, qui permet de retrouver le parcours réalisé et la classe si un groupe est créé par classe, et qui permet la correspondance avec la base de donnée srl.
- questionID : identifiant du problème
- name : intitulé du problème
- type : type d'évènement enregistré
  - submission : validation d'un problème
  - attempt : soumission erronée
  - extra : chargement d'une version d'un problème
  - load : chargement de la page d'accueil
  - proxyload : évènement lié à la communication avec le serveur
- answer : programme soumis par l'utilisateur, collecté au format xml
- score : nombre de points obtenus sur le problème, qui dépend de la version (basic, easy, medium, hard). Ce score n'est enregistré que pour les évènements de type submission.
- date : date d'enregistrement de l'évènement

L'extraction brute pour l'ensemble des données utilisées dans la thèse comporte 141 901 enregistrements.



ID	password	groupID	questionID	name	type	answer	score	date	
63420	250365	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	load	NaN	NaN	2020-12-13 16:07:08
63421	250366	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	extra	"loadLevel;basic"	0.0	2020-12-13 16:07:10
63422	250367	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:08:40
63423	250368	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	submission	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	10.0	2020-12-13 16:08:40
63424	250369	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	extra	"loadLevel;hard"	0.0	2020-12-13 16:08:41
63425	250370	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	extra	"loadLevel;easy"	0.0	2020-12-13 16:08:45
63426	250371	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:09:36
63427	250372	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:11:05
63428	250373	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:11:31
63429	250374	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:12:01
63430	250375	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:12:16
63431	250376	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:13:11
63432	250377	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:14:00
63433	250378	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:15:39
63434	250379	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:16:34
63435	250380	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:17:34
63436	250381	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:17:52
63437	250382	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	attempt	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	NaN	2020-12-13 16:18:08
63438	250383	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	submission	"{\\"basic\\";{\\"blockly\\";{\\"<xml xmlns=\\\"htt...	20.0	2020-12-13 16:18:08
63439	250384	qpqcet78	546641506633946078	603617216933141191	1 - Planter des fleurs [white 2021]	extra	"loadLevel;easy"	0.0	2020-12-13 16:18:09

FIGURE 1 – Extraction de la base de données bebras.chticode (extrait)

## **Annexe 2 : autorisation parentale scolaire**

# **Autorisation de participation**

## **Recherche doctorale en didactique de l'informatique**

### Présentation des travaux de recherche

Marielle Léonard est ingénieur d'étude pour les laboratoires CRISAL et CIREL, université de Lille. Dans le cadre du projet Interreg Teach Transition, elle réalise une recherche doctorale intitulée « Approche didactique et instrumentale des notions de bases en algorithmique ». Cette recherche vise à mieux comprendre l'apprentissage et l'enseignement des notions de base en algorithmique.

Pour ces travaux de recherche, Marielle Léonard collecte les traces d'activités lors du passage des épreuves du concours Algorea organisé par l'association France-ioi. Les élèves participent au concours Algorea dans les mêmes conditions que celles du concours officiel. Seule la collecte de données diffère.

### Collecte et traitement des données

L'activité se déroule sur la plateforme [chticode.algorea.org](http://chticode.algorea.org) gérée par l'association France-ioi et dédiée aux activités de recherche en informatique du laboratoire CRISAL, université de Lille. Des traces de l'activité sont collectées et stockées sur des serveurs sécurisés de AWS, localisés en Europe. La plateforme [chticode.algorea.org](http://chticode.algorea.org) est en conformité avec les lois sur la protection des données personnelles (RGPD). Le détail des informations légales est disponible sur [chticode.algorea.org](http://chticode.algorea.org).

Données collectées dans le cadre de cette étude :

- Nom et prénom
- Classe
- Traces de l'activité : réponses aux exercices, score obtenu, dates d'enregistrement, interactions avec l'interface (clics, mouvements de souris)

À l'issue de la collecte, toutes les données sont anonymisées par Marielle Léonard. Les données sont utilisées exclusivement dans le cadre des travaux de recherche décrits.

### Consentement de participation à l'étude

J'ai pris connaissance des modalités de la collecte de données pour la recherche doctorale « Approche didactique et instrumentale des notions de bases en algorithmique » menée par Marielle Léonard.

J'autorise mon enfant à participer aux tests des parcours Algorea en lien avec cette recherche dans les conditions précisées par ce document.

J'accepte que les données enregistrées à l'occasion de cette recherche puissent faire l'objet d'un traitement informatisé. J'ai bien noté que le droit d'accès prévu par la loi "informatique et libertés" s'exerce à tout moment auprès des responsables de l'étude. Je pourrai exercer mon droit de rectification auprès de ceux-ci.

Nom du participant :

Date, nom et signature d'un représentant légal :

## **Annexe 3 : autorisation parentale individuelle**

# **Autorisation de participation**

## **Recherche doctorale en didactique de l'informatique**

### Présentation des travaux de recherche

Marielle Léonard est ingénieur d'étude pour les laboratoires CRISStAL et CIREL, université de Lille. Dans le cadre du projet Interreg Teach Transition, elle réalise une recherche doctorale intitulée « Approche didactique et instrumentale des notions de bases en algorithmique ». Cette recherche vise à mieux comprendre l'apprentissage et l'enseignement des notions de base en algorithmique.

Pour ces travaux de recherche, Marielle Léonard organise des tests en ligne des parcours du concours Algorea organisé par l'association France-ioi. Les tests se déroulent à distance, en partage d'écran via le logiciel Zoom. Un test dure 45 minutes. Il consiste à participer à une épreuve du concours Algorea, et à échanger avec Marielle Léonard au cours de l'activité. Pour certaines épreuves, un questionnaire complémentaire à l'activité pourra être soumis au participant à l'issue de celle-ci.

### Collecte et traitement des données

L'activité se déroule sur la plateforme [chticode.algorea.org](https://chticode.algorea.org) gérée par l'association France-ioi et dédiée aux activités de recherche en informatique du laboratoire CRISStAL, université de Lille. Des traces de l'activité sont collectées et stockées sur des serveurs sécurisés de AWS, localisés en Europe. La plateforme [chticode.algorea.org](https://chticode.algorea.org) est en conformité avec les lois sur la protection des données personnelles (RGPD). Le détail des informations légales est disponible sur [chticode.algorea.org](https://chticode.algorea.org).

Données collectées :

- Nom et prénom
- Classe
- Traces de l'activité : réponses aux exercices, score obtenu, dates d'enregistrement, interactions avec l'interface (clics, mouvements de souris)
- Enregistrement vidéo de la participation
- Réponses au questionnaire

À l'issue de la collecte, toutes les données sont anonymisées par Marielle Léonard. Les données sont utilisées exclusivement dans le cadre des travaux de recherche décrits.

### Consentement de participation à l'étude

J'ai pris connaissance des modalités de la collecte de données pour la recherche doctorale « Approche didactique et instrumentale des notions de bases en algorithmique » menée par Marielle Léonard.

J'autorise mon enfant à participer aux tests des parcours Algorea en lien avec cette recherche dans les conditions précisées par ce document.

J'accepte que les données enregistrées à l'occasion de cette recherche puissent faire l'objet d'un traitement informatisé. J'ai bien noté que le droit d'accès prévu par la loi "informatique et libertés" s'exerce à tout moment auprès des responsables de l'étude. Je pourrai exercer mon droit de rectification auprès de ceux-ci.

Nom du participant :

Date, nom et signature d'un représentant légal :

## **Annexe 4 : analyse des données à l'échelle nationale**

# analyse\_donnees\_nationales

February 27, 2024

```
[1]: import sys
      sys.path.append('../')

      from src import *
      import pandas as pd #
      import seaborn as sns
      import numpy as np
      import matplotlib.pyplot as plt
      import scipy.stats as st
      from scipy.stats import chi2_contingency

      %load_ext autoreload
      %autoreload 2
```

pygame 2.1.2 (SDL 2.0.16, Python 3.9.15)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

2024-02-26 06:57:33.553451: I tensorflow/core/platform/cpu\_feature\_guard.cc:193]

This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2024-02-26 06:57:33.799168: E tensorflow/stream\_executor/cuda/cuda\_blas.cc:2981]

Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

2024-02-26 06:57:34.395220: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot open shared object file: No such file or directory

2024-02-26 06:57:34.395280: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer\_plugin.so.7'; dlerror: libnvinfer\_plugin.so.7: cannot open shared object file: No such file or directory

2024-02-26 06:57:34.395286: W

tensorflow/compiler/tf2tensorrt/utils/py\_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed

properly.

Pour nos analyses à l'échelle nationale, nous gardons les puzzles pour lesquels l'utilisation d'une boucle est contrainte, l'imbrication de boucles n'est pas contrainte, et il n'y a pas de tests multiples.

### 0.0.1 Collecte de données : nombre de participants aux différents tours du concours Algorea

```
[2]: dt = pd.read_csv('../data/data_these/resultats_concours.csv',
    ↳dtype={"questionId sur chticode" : 'str'})
effectif_tours = dt[['annee', 'tour', 'Effectif_all']].groupby(by=['annee',
    ↳'tour']).max()
eff_tours = effectif_tours.reset_index()
eff_tours['annee-tour'] = eff_tours.apply(lambda row: str(row['annee']) + '-' +
    ↳str(row['tour']), axis=1)
eff_tours[['annee-tour', 'Effectif_all']]
```

```
[2]:
```

	annee-tour	Effectif_all
0	2018-1	71583
1	2018-2	17361
2	2018-3	6497
3	2019-1	89805
4	2019-2	28847
5	2019-3	9094
6	2020-1	25979
7	2020-2	12853
8	2020-3	2737
9	2021-1	76988
10	2021-2	25875
11	2021-3	11519
12	2022-1	92410
13	2022-2	42936
14	2022-3	18749

```
[3]: xlabel = list(eff_tours['annee-tour'].unique())
```

```
[4]: fig, ax = plt.subplots()

fig.set_size_inches(10, 6)

ax.bar(eff_tours['annee-tour'], eff_tours['Effectif_all'], zorder=3)

ax.set_xlabel('Tours du concours Algorea')
ax.xaxis.set_ticklabels(xlabel, rotation = 30, fontsize = 8)
ax.grid(axis='y', zorder=0)

ax.set_ylabel('Nombre de participants')
ax.yaxis.set_ticks(range(0,100000, 10000))
```

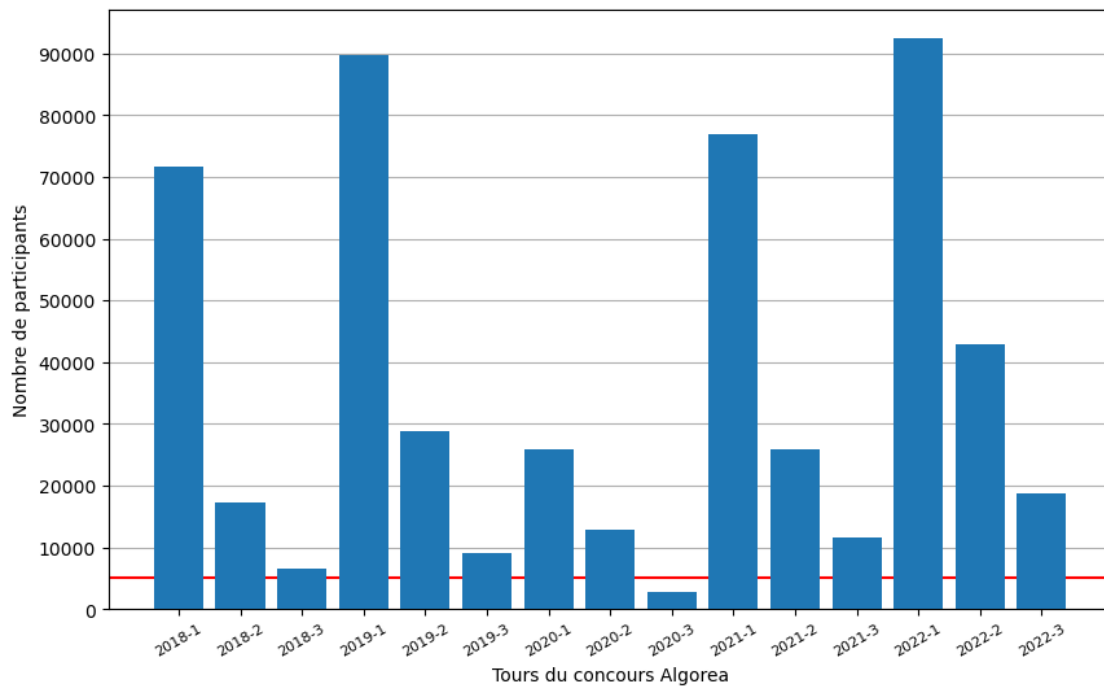


```
ax.axhline(y=5000, color='r', label='seuil')
```

```
#plt.savefig('participants_algorea.pdf')  
plt.show()
```

/tmp/ipykernel\_13094/1911791003.py:8: UserWarning: FixedFormatter should only be used together with FixedLocator

```
ax.xaxis.set_ticklabels(xlabels, rotation = 30, fontsize = 8)
```



## 0.0.2 Étude de la robustesse des données

Simulations avec le test du chi2 pour déterminer des seuils pour lesquels des différences de fréquence de réussite sont significatives.

```
[5]: # Test du chi2
```

```
def calculer_p_value(ligne):  
    chi2, p_value, _, _ = chi2_contingency([ligne["répartition pour la_"  
↪situation 1"], ligne["répartition pour la situation 2"]])  
    return p_value
```

```
[6]:
```

```

data = {
    "effectifs pour les situations 1 et 2" : ["100-100", "100-100", "100-500",
↪ "200-200", "250-250", "300-300", "500-500",
                                                "500-500", "800-1000",
↪ "1000-1000", "3000-3000"
                                                ],
    "différence de fréquence de réussite" : [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
0.05, 0.05, 0.05, 0.05
                                                ],
    "répartition pour la situation 1" : [[45, 55], [10, 90], [45, 55], [90,
↪ 110], [112, 138], [135, 165], [225, 275],
                                                [225, 275], [360, 440], [450, 550], [1350, 1650]
                                                ],
    "répartition pour la situation 2" : [[55, 45], [20, 80], [275, 225], [110,
↪ 90], [138, 112], [165, 135], [275, 225],
                                                [250, 250], [500, 500], [500, 500], [1500, 1500]
                                                ]
}

```

```

[7]: df_chi2 = pd.DataFrame(data)
df_chi2["p_value pour le test du chi2"] = df_chi2.apply(calculer_p_value,
↪ axis=1)

```

```

[8]: df_chi2

```

```

[8]:   effectifs pour les situations 1 et 2  différence de fréquence de réussite \
0          100-100          0.10
1          100-100          0.10
2          100-500          0.10
3          200-200          0.10
4          250-250          0.10
5          300-300          0.10
6          500-500          0.10
7          500-500          0.05
8          800-1000         0.05
9         1000-1000         0.05
10         3000-3000         0.05

```

```

répartition pour la situation 1 répartition pour la situation 2 \
0          [45, 55]          [55, 45]
1          [10, 90]          [20, 80]
2          [45, 55]          [275, 225]
3          [90, 110]         [110, 90]
4          [112, 138]        [138, 112]
5          [135, 165]        [165, 135]
6          [225, 275]        [275, 225]
7          [225, 275]        [250, 250]

```

8	[360, 440]	[500, 500]
9	[450, 550]	[500, 500]
10	[1350, 1650]	[1500, 1500]

	p_value pour le test du chi2
0	0.203092
1	0.074706
2	0.085428
3	0.057433
4	0.025347
5	0.017892
6	0.001942
7	0.128563
8	0.039133
9	0.028229
10	0.000117

Nous déduisons de la simulation précédente qu'une différence de fréquence de réussite entre deux puzzles de 0.05 est très significative ( $p\text{-value} < 0.01$ ) lorsque l'on considère les effectifs globaux (tous niveaux scolaires confondus). Pour les niveaux scolaires du collège, les différences de fréquence de réussite entre deux puzzles sont toutes significatives ( $p\text{-value} < 0.05$ ), sauf pour le tour 2020-3. Les données pour les niveaux scolaires de l'école élémentaires sont moins robustes. Nous mettons sur fond orangé les effectifs qui ne permettent pas qu'une différence de fréquence de réussite de 0.1 soit significative.

Au terme de cette analyse, nous écartons les résultats du tour 2020-3 de notre analyse des puzzles, à cause du trop petit nombre de participants par niveau scolaire (reprise de l'école après le premier confinement). Ainsi, le nombre de participants de chaque tour est supérieur à 5000, ce qui confère une bonne robustesse globale aux données. Dans cette configuration, nous gardons 139 puzzles des éditions 2018 à 2022 pour l'analyse des résultats à l'échelle nationale.

### 0.0.3 Extraction des résultats pris en compte dans la thèse dans le chapitre d'analyse au niveau national

```
[9]: dt_all = pd.read_csv('../data/data_these/tags_situations.csv',
↳ dtype={'questionId sur chticode' : 'str'})
```

```
[10]: dt_all.columns
```

```
[10]: Index(['id_situation', 'questionId sur chticode', 'Version',
↳ 'Référence interne', 'Année', 'Tour', 'Catégorie', 'Numero problème',
↳ 'Effectif_all', 'all', 'CM1', 'CM2', '6ème', '5ème', '4ème', '3ème',
↳ 'Contexte', 'Notion algo', 'Nb cases du motif visuel',
↳ 'composition_motif', 'motif_case', 'orientation', 'disposition',
↳ 'elt_id', 'adj_diff', 'rotation_mentale', 'decor', 'classe_situation',
↳ 'nb_loops', 'length_motif', 'length_program', 'before_loop',
↳ 'after_loop', 'last_instruction_loop', 'answer_ref'],
↳ dtype='object')
```

```
[11]: dt_loop = dt_all[dt_all['classe_situation'].notna()]
dt_loop = dt_loop.astype({'classe_situation': 'int',
                          'nb_loops': 'int',
                          'length_motif': 'int',
                          'length_program': 'int'})
```

### Répartition des puzzles retenus sur les différentes éditions

```
[12]: dt_puzzles_tour = dt_loop[['Année', 'all']].groupby(['Année']).count().
      ↪reset_index()
dt_puzzles_tour
```

```
[12]:
```

	Année	all
0	2018	24
1	2019	26
2	2020	20
3	2021	31
4	2022	38

```
[13]: fig, ax = plt.subplots()

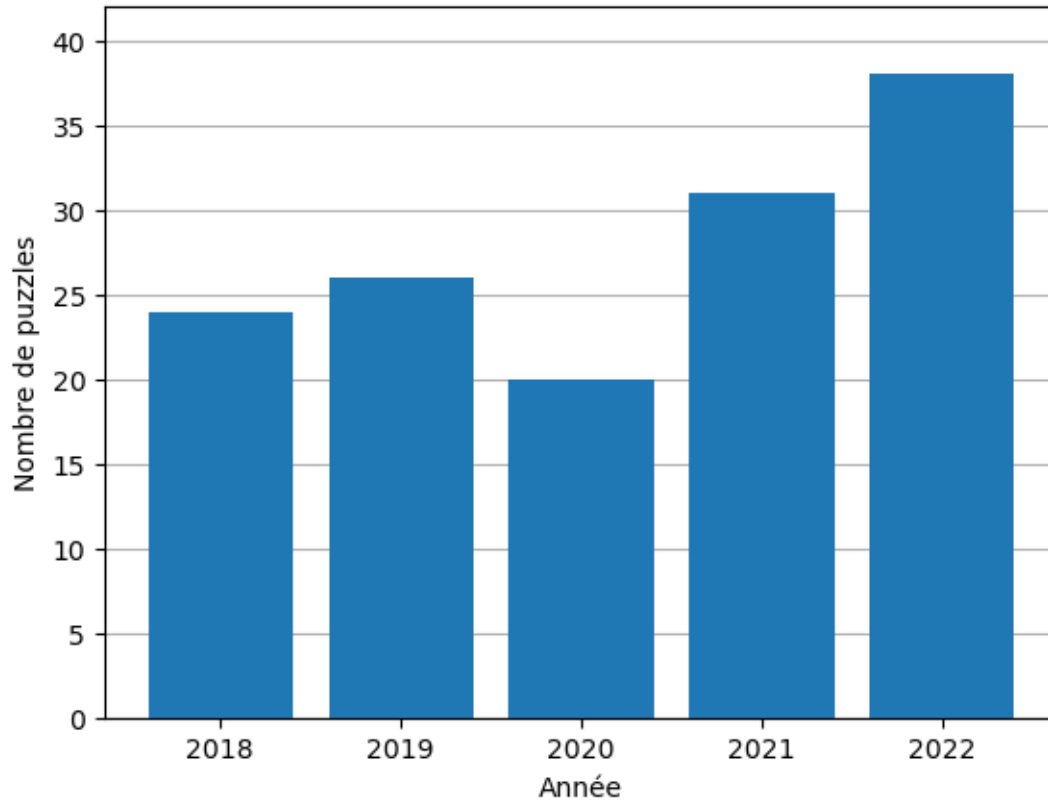
ax.bar(dt_puzzles_tour['Année'], dt_puzzles_tour['all'], zorder=3)

ax.set_xlabel('Année')
ax.set_ylabel('Nombre de puzzles')

ax.grid(True, axis='y', zorder=0)

ax.set_ylim(0, 42)

#plt.savefig('repartition_puzzles_boucles_annees.pdf')
plt.show()
```



##### Corrélation entre le nombre d'instructions de la solution de référence et la fréquence de réussite

```
[14]: dt_taux = dt_loop[['id_situation', 'Version', 'length_program', 'all']]
versions= [('basic', '1 étoile'), ('easy', '2 étoiles'), ('medium', '3_
→étoiles'), ('hard', '4 étoiles')]

for version in versions :
    dt_taux.loc[dt_taux['Version'] == version[0], 'Version'] = version[1]

dt_taux.head(5)
```

```
[14]:   id_situation   Version  length_program  all
0           1  2 étoiles           3  0.982
1           2  2 étoiles           3  0.974
2           3  2 étoiles           4  0.983
3           4  3 étoiles           4  0.936
4           5  3 étoiles           4  0.908
```

```
[15]: # coefficient de corrélation de Pearson entre longueur du programme et_
→fréquence de réussite
```

```

coefficient_correlation, valeur_p = st.pearsonr(dt_taux['length_program'],
↳dt_taux['all'])

print("Coefficient de corrélation de Pearson :", coefficient_correlation)
print("Valeur p_value :", valeur_p)

```

Coefficient de corrélation de Pearson : -0.7697765963525653  
Valeur p\_value : 1.713768308321162e-28

```

[16]: plt.figure(figsize=(12, 9))
fig, ax = plt.subplots()

couleurs = {'1 étoile': '#ffe500', '2 étoiles': '#ff8000', '3 étoiles':
↳'#e03500', '4 étoiles': '#7c1d00'}

sns.scatterplot(data=dt_taux, x="length_program", y="all", hue="Version",
↳size=15, palette= couleurs, zorder=3)

ax.set_xlabel('Nombre de blocs de la solution de référence')
ax.set_ylabel('Fréquence de réussite')

handles, labels = plt.gca().get_legend_handles_labels()
new_order = [2, 0, 1, 3] # Nouvel ordre des étiquettes (indices)
plt.legend([handles[i] for i in new_order], [labels[i] for i in new_order],
↳title='versions')

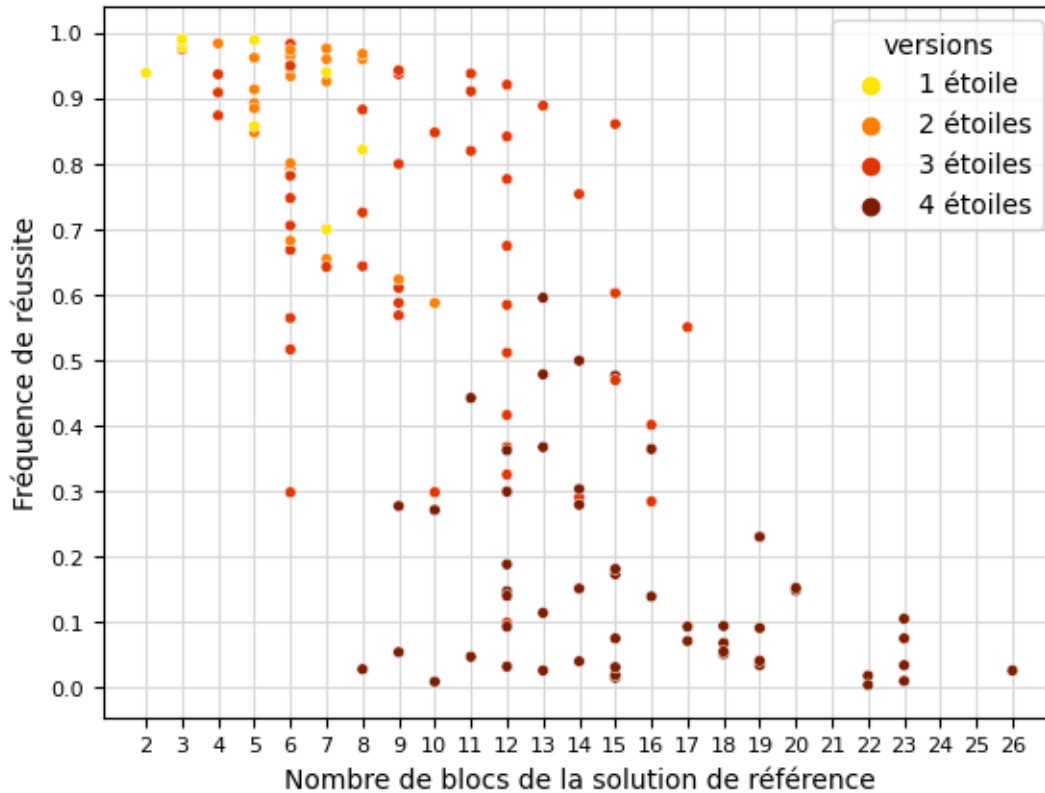
plt.xticks(range(2,27))
plt.yticks(np.arange(0, 1.05, 0.1))
plt.tick_params(labelsize=8)

ax.grid(True , color='#d9d9d9', zorder=0)

#plt.savefig('nb_instructions_sol_reference.pdf')
plt.show()

```

<Figure size 1200x900 with 0 Axes>



#### 0.0.4 Analyses concernant le motif visuel

Corrélation entre le nombre de cases du motif visuel et la fréquence de réussite

```
[17]: plt.figure(figsize=(12, 9))
fig, ax = plt.subplots()

sns.scatterplot(data=dt_loop, x="Nb cases du motif visuel", y="all", size=15,
               zorder=3, legend=False)

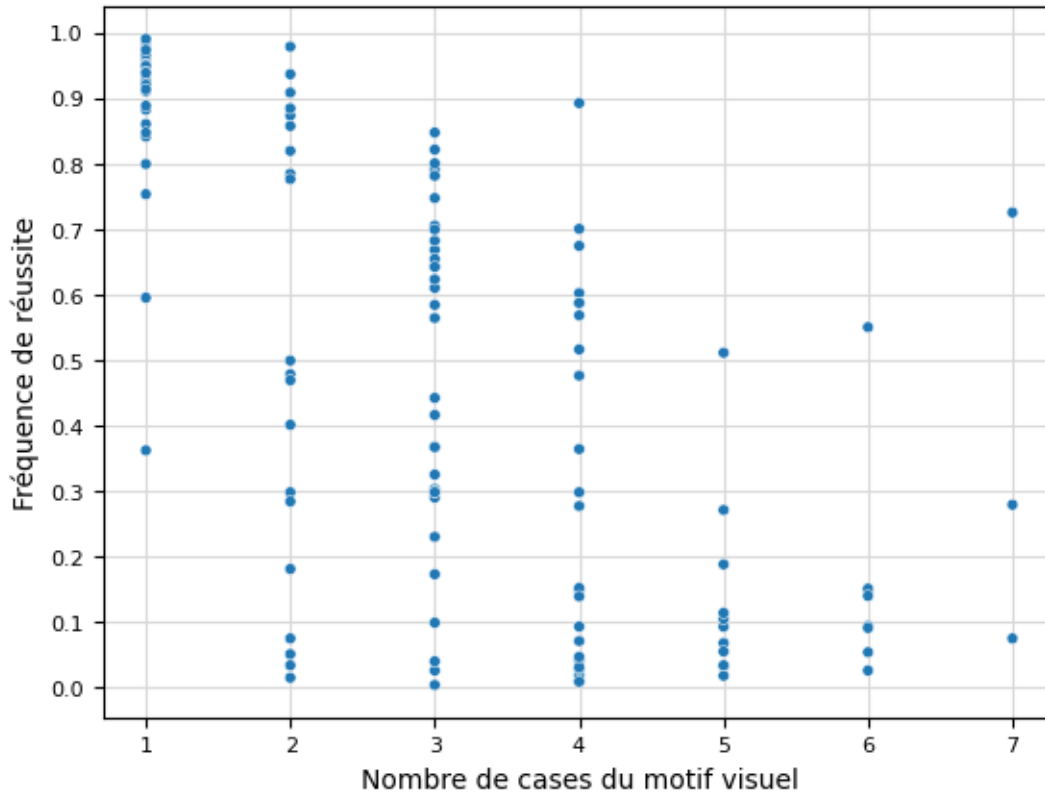
ax.set_xlabel('Nombre de cases du motif visuel')
ax.set_ylabel('Fréquence de réussite')

plt.xticks(range(1,8))
plt.yticks(np.arange(0, 1.05, 0.1))
plt.tick_params(labels=8)

ax.grid(True, color='#d9d9d9', zorder=0)

#plt.savefig('taille_motif.pdf')
plt.show()
```

<Figure size 1200x900 with 0 Axes>



```
[18]: # Coefficient de corrélation de Pearson entre nombre de cases du motif visuel
      ↪ et fréquence de réussite
      coefficient_correlation, valeur_p = st.pearsonr(dt_loop["Nb cases du motif
      ↪ visuel"], dt_loop['all'])

      print("Coefficient de corrélation de Pearson :", coefficient_correlation)
      print("Valeur p_value :", valeur_p)
```

Coefficient de corrélation de Pearson : -0.7027743924152191  
Valeur p\_value : 5.29046772512229e-22

### Motif visuel sur une case vs plusieurs cases

```
[19]: dt_loop['motif_case'].value_counts()
```

```
[19]: non    98
      oui    41
      Name: motif_case, dtype: int64
```

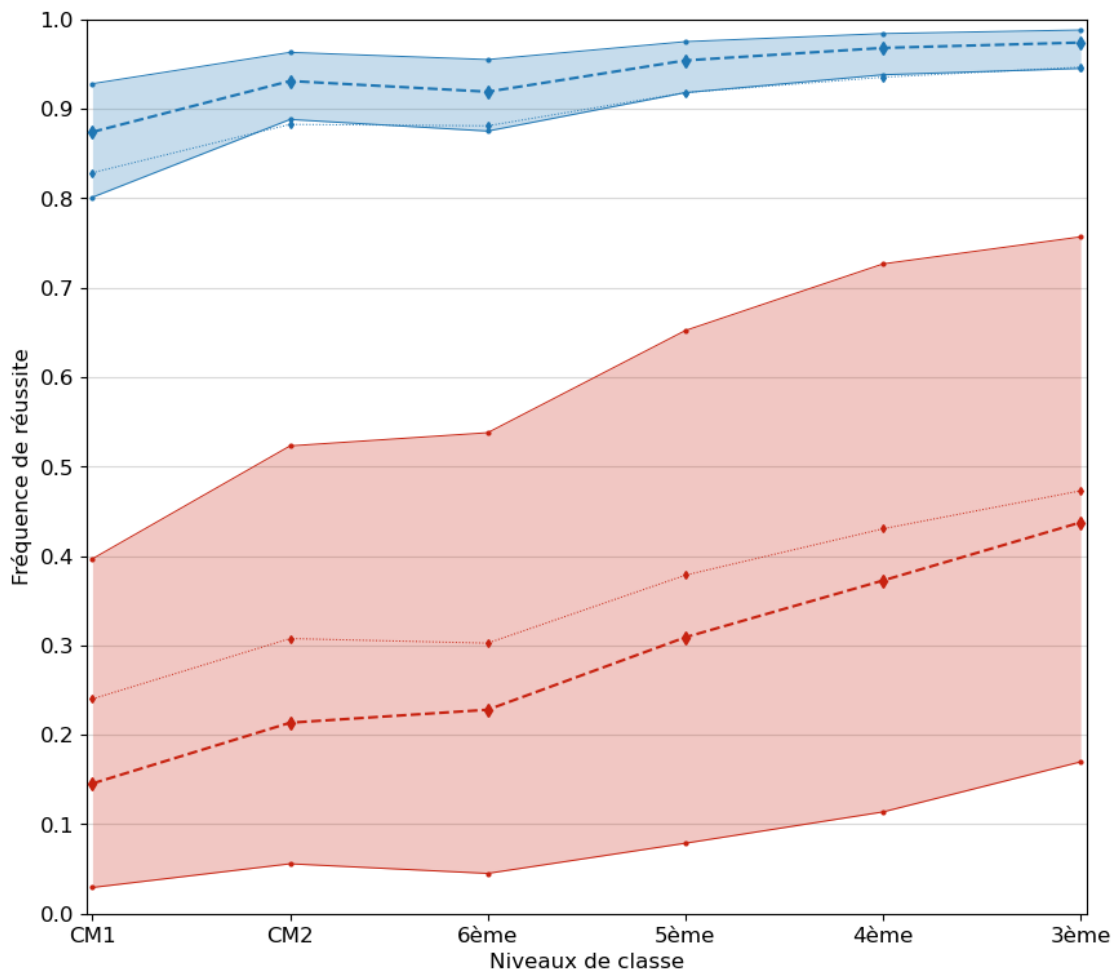
```
[20]: dt_motif_case = dt_loop.groupby('motif_case')
```



```
[21]: dt_loop.groupby('motif_case')['length_program'].mean()
```

```
[21]: motif_case  
non    12.265306  
oui     7.512195  
Name: length_program, dtype: float64
```

```
[22]: diagramme_reussite_categorie_puzzle(dt_loop,  
                                          'motif_case',  
                                          [('oui', '#1f77b4'), ('non', '#c82210')],  
                                          [],  
                                          #nom_fichier = "situations_motif_case.png",  
                                          moyenne=True  
                                          )
```



```
[23]: # Test de Shapiro-Wilk appliqué à tous les groupes

from scipy.stats import shapiro

resultats_shapiro = dt_motif_case['all'].apply(lambda x: shapiro(x))

print(resultats_shapiro)
```

```
motif_case
non      (0.9086430668830872, 4.4952030293643475e-06)
oui      (0.5906064510345459, 1.7374478522569348e-09)
Name: all, dtype: object
```

Comme les distributions ne sont pas normales, nous ne pouvons pas appliqué un test t de Student. Dans ce cas, il faut le remplacer par un test de Mann-Whitney U.

Ce test est utilisé pour comparer les distributions de deux échantillons indépendants lorsque les données ne sont pas distribuées normalement.

Il fonctionne en classant toutes les données combinées des deux groupes, puis en comparant les sommes de rangs des deux groupes. Si les deux groupes viennent de la même distribution, les sommes de rangs devraient être similaires. Si les deux groupes diffèrent significativement, cela suggère que les distributions des deux groupes sont différentes.

```
[24]: # Test de Mann-Whitney U

from scipy.stats import mannwhitneyu

groupes = [groupe[1]['all'].values for groupe in dt_motif_case]

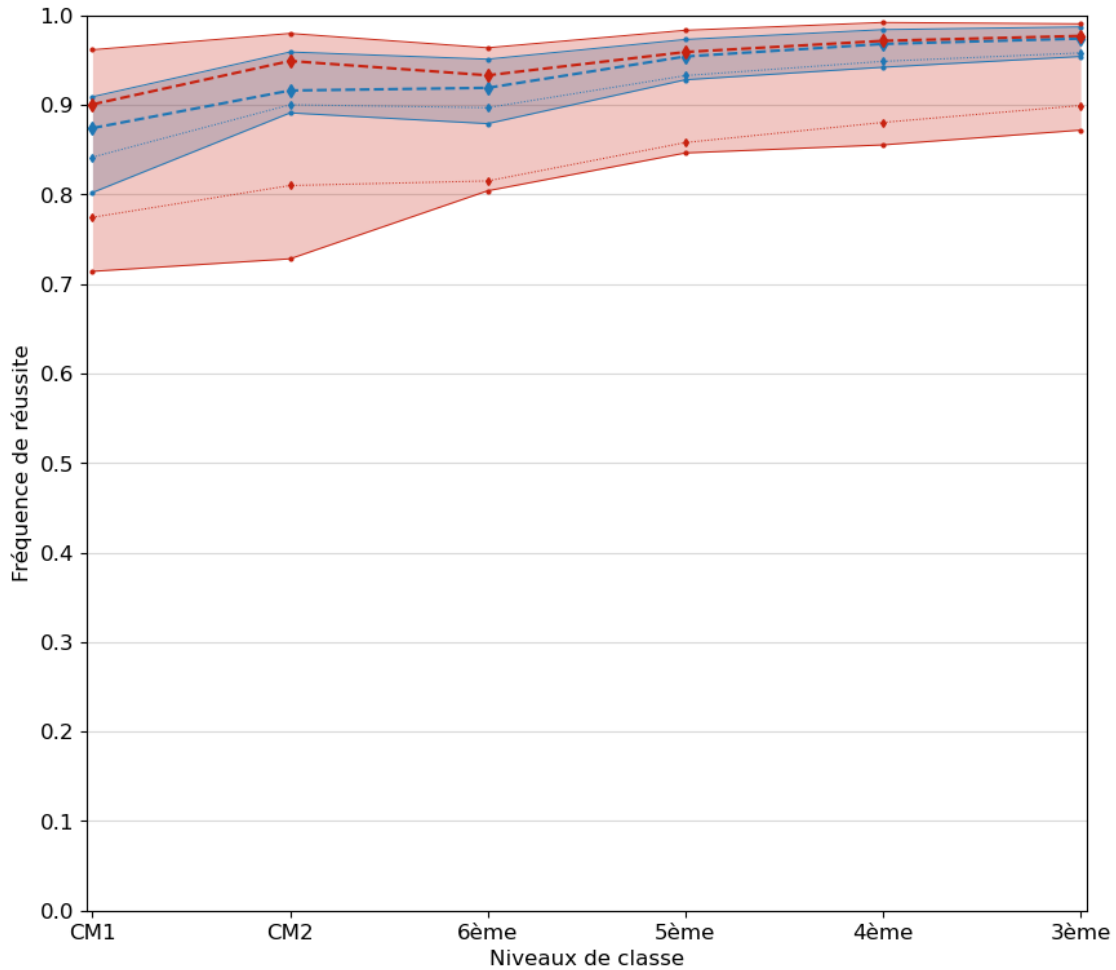
resultats_mannwhitneyu = mannwhitneyu(*groupes)

print(resultats_mannwhitneyu)
```

```
MannwhitneyuResult(statistic=163.0, pvalue=1.5412268396564835e-17)
```

Pour les motifs visuels sur une seule case, comparaison entre les situations avec une et deux instructions dans la boucle

```
[25]: diagramme_reussite_categorie_puzzle(dt_loop[dt_loop['classe_situation']==1],
                                         'length_motif',
                                         [(1, '#1f77b4'),
                                          (2, '#c82210')
                                         ],
                                         [],
                                         '',
                                         moyenne=True
                                         )
```



Le passage de une à deux instructions dans le corps de la boucle ne provoque pas de différence de fréquence de réussite significative, contrairement au passage de une à plusieurs cases pour le motif visuel.

### 0.0.5 Composition des motifs

#### Motifs de taille 2

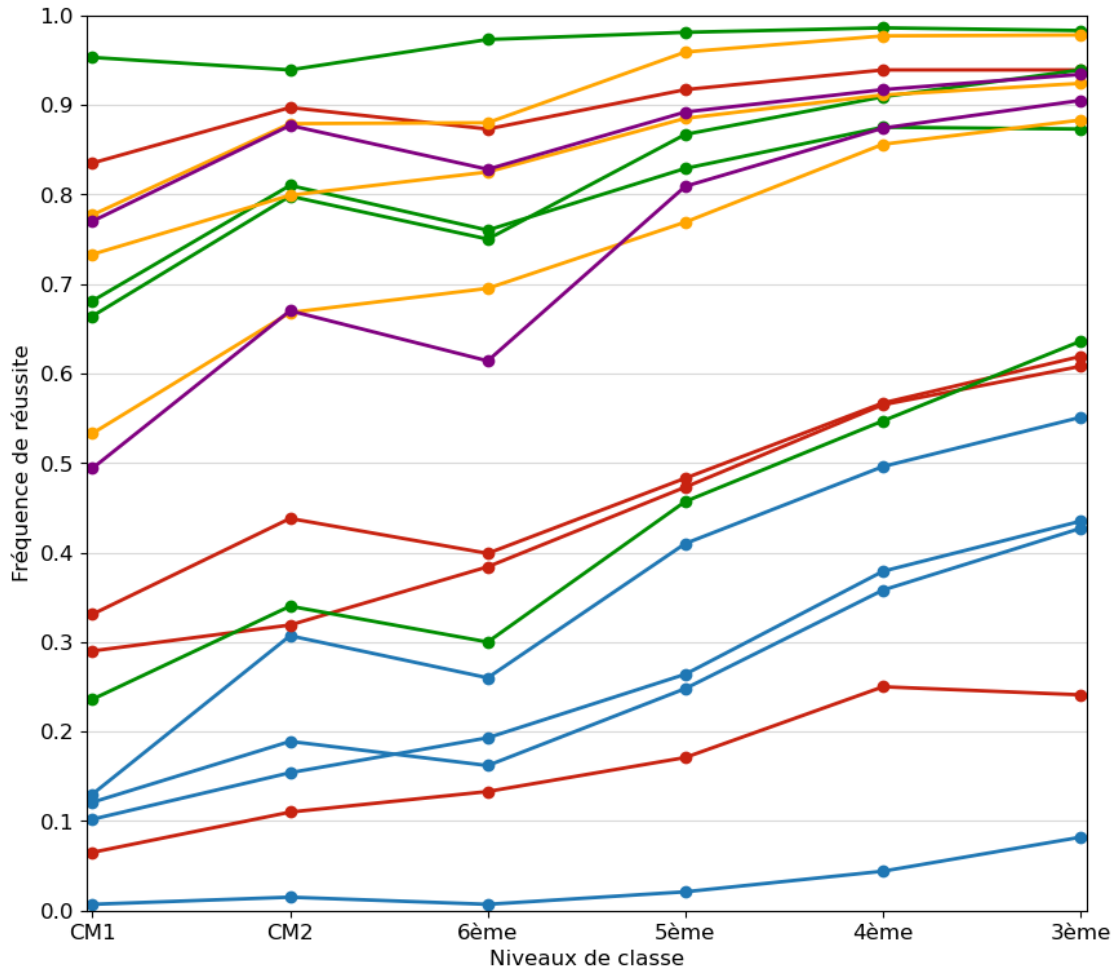
```
[26]: dt_loop[dt_loop['Nb cases du motif visuel']==2][['id_situation', 'Référence_
↳interne', 'Année', 'Tour', 'Version', 'composition_motif']]
```

```
[26]:
```

	id_situation	Référence interne	Année	Tour	Version	composition_motif
3	4	white-1-flowers	2020	1	medium	XA
4	5	white-04-space	2021	3	medium	AX
9	10	white-05-course	2021	3	basic	XX
11	12	white-05-flowers	2021	2	basic	XA
15	16	white-05-flowers	2021	2	medium	AA
26	27	white-04-flowers	2021	1	hard	AA

27	28	white-05-gems	2021	1	medium	AA
28	29	white-05-gems	2021	1	hard	diff
29	30	white-01-paint	2021	2	hard	AX
37	38	white-04-space	2021	3	hard	AX
38	39	white-05-course	2021	3	medium	XX
41	42	white-3-marbles	2020	1	medium	AB
64	65	white-3-dominoes	2019	2	hard	diff
72	73	white-3-gems	2019	3	medium	AA
73	74	white-3-gems	2019	3	hard	diff
86	87	01_repeat_paint	2018	2	hard	AX
87	88	02_repeat_flowers	2018	2	medium	XA
108	109	white-05-gem-2	2022	1	basic	XX
110	111	white-05-gem-2	2022	1	medium	XX
191	192	white-04-castle	2022	2	easy	AB

```
[27]: diagramme_reussite_categorie_puzzle(dt_loop,
                                         'composition_motif',
                                         #[( 'AA', '#1f77b4'), ('AX',
→ '#c82210'), ('XX', '#008e00'), ('XA', 'orange')],
                                         puzzles = [(16, '#1f77b4'), #AA
                                                       (27, '#1f77b4'),
                                                       (28, '#1f77b4'),
                                                       (73, '#1f77b4'),
                                                       (5, '#c82210'), #AX
                                                       (30, '#c82210'),
                                                       (38, '#c82210'),
                                                       (87, '#c82210'),
                                                       (10, '#008e00'), #XX
                                                       (39, '#008e00'),
                                                       (109, '#008e00'),
                                                       (111, '#008e00'),
                                                       (4, 'orange'), #XA
                                                       (12, 'orange'),
                                                       (88, 'orange'),
                                                       (42, 'purple'), #AB
                                                       (192, 'purple'),
                                         ],
                                         nom_fichier = "situations_motif_taille_2.
→png",
                                         moyenne=False
                                         )
```



Étant donné le faible nombre d'occurrences pour chaque type de motifs et la dispersion des fréquences de réussite, nous ne voyons pas comment tirer de conclusions significatives quant à l'impact de la composition du motif.

### Motifs de taille 3

```
[28]: dt_loop[dt_loop['Nb cases du motif visuel']==3][['id_situation', 'Référence_
↳interne', 'Année', 'Tour', 'Version',_
↳'composition_motif']]['composition_motif'].value_counts()
```

```
[28]: AXX      5
      XXA      4
      AAX      3
      diff     3
      XXX      3
      AXB      2
      AAA      2
      ABX      2
```

```

AAB      2
XAB      2
ABA      1
XAA      1
AXA      1
XAX      1
Name: composition_motif, dtype: int64

```

### 0.0.6 Fonction des éléments saillants

Étude de la fonction des éléments saillants pour les situations où le motif visuel s'étend sur plusieurs cases

```
[29]: dt_plusieurs_cases = dt_loop[dt_loop['motif_case']=='non']
```

```
[30]: dt_plusieurs_cases['decor'].value_counts()
```

```
[30]: non          33
      présent     31
      parcours    18
      visuel      16
      Name: decor, dtype: int64
```

```
[31]: dt_plusieurs_cases.groupby('decor')['length_program'].mean()
```

```
[31]: decor
      non          9.575758
      parcours    10.888889
      présent     13.000000
      visuel      17.937500
      Name: length_program, dtype: float64
```

```
[32]: dt_plusieurs_cases_describe = dt_plusieurs_cases.groupby('decor')['all'].
      ↪describe()
      dt_plusieurs_cases_describe['EI'] =
      ↪dt_plusieurs_cases_describe['75%']-dt_plusieurs_cases_describe['25%']
      dt_plusieurs_cases_describe
```

```
[32]:
```

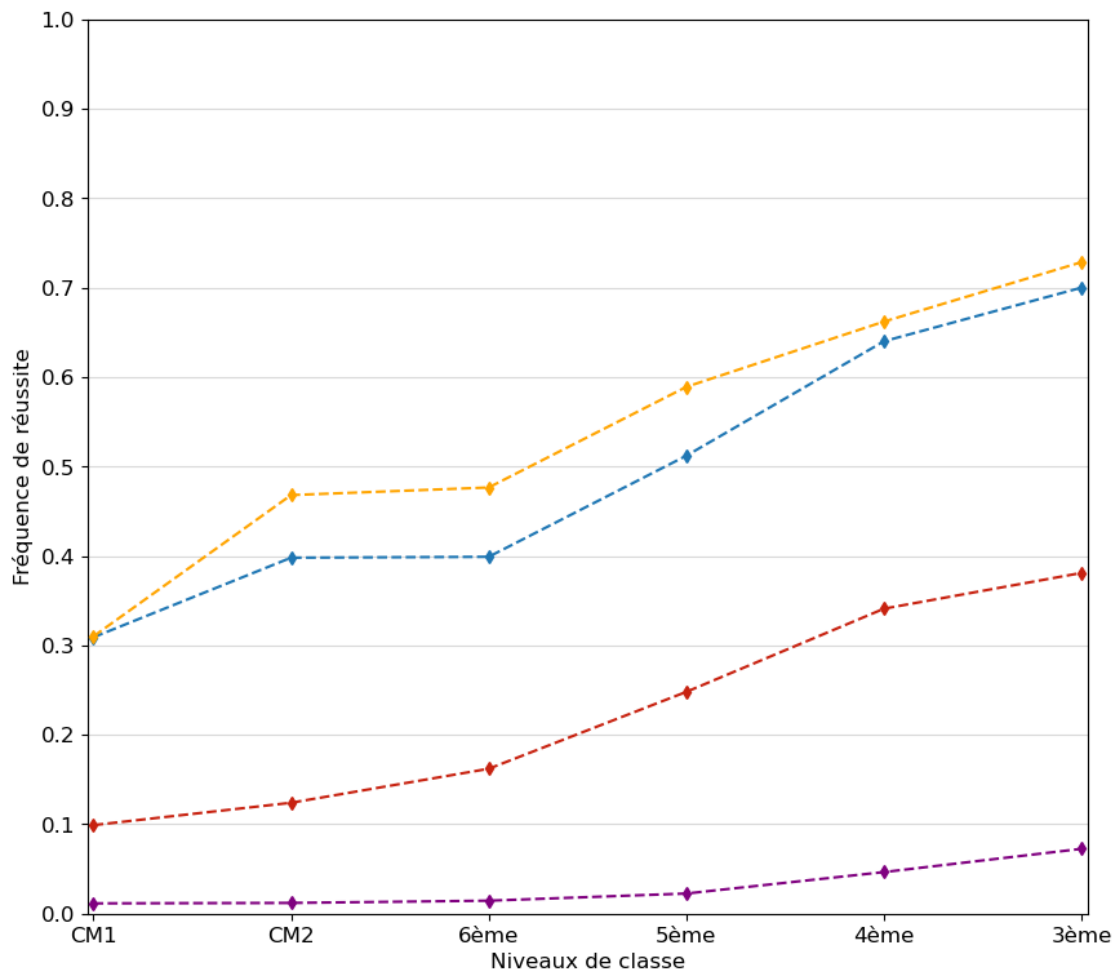
	count	mean	std	min	25%	50%	75%	max \
decor								
non	33.0	0.499697	0.290522	0.028	0.29000	0.516	0.7760	0.936
parcours	18.0	0.575556	0.248859	0.031	0.44875	0.583	0.7200	0.978
présent	31.0	0.308871	0.250363	0.009	0.09650	0.277	0.4925	0.821
visuel	16.0	0.054063	0.049705	0.004	0.01875	0.033	0.0795	0.188

EI

decor

non 0.48600  
parcours 0.27125  
présent 0.39600  
visuel 0.06075

```
[33]: diagramme_reussite_categorie_puzzle(dt_plusieurs_cases,  
      'decor',  
      [('non', '#1f77b4'),  
       ('présent', '#c82210'),  
       ('parcours', 'orange'),  
       ('visuel', 'purple')  
      ],  
      puzzles=[],  
      #moyenne=True,  
      ecart_interquartile=False,  
      #nom_fichier='decor_perturbation_visuelle.  
      ↪png'  
      )#
```



```
[34]: groupe1 = dt_plusieurs_cases[dt_plusieurs_cases['decor']=='non']['all']
groupe2 = dt_plusieurs_cases[dt_plusieurs_cases['decor']=='parcours']['all']
groupe3 = dt_plusieurs_cases[dt_plusieurs_cases['decor']=='présent']['all']
groupe4 = dt_plusieurs_cases[dt_plusieurs_cases['decor']=='visuel']['all']
```

```
[35]: # Test de Shapiro-Wilk pour la normalité

from scipy.stats import shapiro

num =1

for groupe in [groupe1, groupe2, groupe3, groupe4] :
    print("Groupe " + str(num) + ":")

    statistique_shapiro, p_value_shapiro = shapiro(groupe)

    # Afficher les résultats
    print("Statistique de Shapiro-Wilk :", statistique_shapiro)
    print("p-value :", p_value_shapiro)

    # Interprétation
    if p_value_shapiro > 0.05:
        print("Les données semblent suivre une distribution normale.")
    else:
        print("Les données ne suivent pas une distribution normale.")

    num+=1
```

```
Groupe 1:
Statistique de Shapiro-Wilk : 0.9357710480690002
p-value : 0.05123157426714897
Les données semblent suivre une distribution normale.
Groupe 2:
Statistique de Shapiro-Wilk : 0.9800954461097717
p-value : 0.9509924054145813
Les données semblent suivre une distribution normale.
Groupe 3:
Statistique de Shapiro-Wilk : 0.875853955745697
p-value : 0.0018966669449582696
Les données ne suivent pas une distribution normale.
Groupe 4:
Statistique de Shapiro-Wilk : 0.8455430269241333
p-value : 0.011722786352038383
Les données ne suivent pas une distribution normale.
```



Comme deux des 4 groupes ne suivent pas une distribution normale, nous appliquons un test de Kruskal-Wallis, non paramétrique

```
[36]: # Test de Kruskal-Wallis

from scipy.stats import kruskal

statistique_kw, p_value_kw = kruskal(groupe1, groupe2, groupe3, groupe4)

print("Statistique de Kruskal-Wallis :", statistique_kw)
print("p-value :", p_value_kw)
```

```
Statistique de Kruskal-Wallis : 36.52434738159624
p-value : 5.80105198091641e-08
```

Deux au moins de ces groupes présentent une différence de fréquence de réussite significative

```
[37]: # Calcul de la variance pour chaque groupe

num = 1

for groupe in [groupe1, groupe2, groupe3, groupe4] :
    variance = np.var(groupe, ddof=1) # ddof=1 pour calculer la variance non-
    ↪biaisée
    print("Variance du groupe " + str(num) + ":", variance)
    num+=1
```

```
Variance du groupe 1: 0.08440296780303032
Variance du groupe 2: 0.06193061437908495
Variance du groupe 3: 0.06268184946236559
Variance du groupe 4: 0.0024705958333333332
```

La variance peut être considérée comme équivalente pour les groupes 1, 2 et 3, mais pas pour le groupe 4. Nous appliquons donc un test de Conovan-Iman, suivi d'une correction de Bonferroni, pour comparer les groupes deux à deux.

```
[38]: # Test de Conover-Iman

from scikit_posthocs import posthoc_conover

donnees = [groupe1, groupe2, groupe3, groupe4]

resultats = posthoc_conover(donnees)

print(resultats)
```

	1	2	3	4
1	1.000000e+00	3.107440e-01	0.007802	2.785439e-09
2	3.107440e-01	1.000000e+00	0.001356	1.569100e-09

```

3  7.801744e-03  1.355560e-03  1.000000  4.307891e-05
4  2.785439e-09  1.569100e-09  0.000043  1.000000e+00

```

```

[39]: # Correction de Bonferroni pour ajuster les p-valeurs en fonction du nombre de
      ↪comparaisons effectuées

      # Nombre total de comparaisons (hors diagonale)
      nb_comparaisons = np.size(resultats) - len(resultats)

      # Correction de Bonferroni
      p_values_adjusted = resultats * nb_comparaisons

      print(round(p_values_adjusted, 4))

```

```

          1          2          3          4
1  12.0000   3.7289   0.0936   0.0000
2   3.7289  12.0000   0.0163   0.0000
3   0.0936   0.0163  12.0000   0.0005
4   0.0000   0.0000   0.0005  12.0000

```

Le test de Tukey permet de comparer toutes les paires de moyennes de groupes après avoir réalisé le test de Kruskal-Wallis, lorsque les variances des différents groupes sont comparables. Il affiche les différences significatives entre les moyennes de chaque paire de groupes, ainsi que des intervalles de confiance pour chaque comparaison.

Le test de Tukey n'est normalement pas applicable pour les 4 groupes, car la variance du groupe 4 diffère trop des autres, mais il est applicable si on ne prend que les groupes 1, 2 et 3.

```

[40]: # Test de Tukey 4 groupes

      from statsmodels.stats.multicomp import pairwise_tukeyhsd

      donnees = np.concatenate([groupe1, groupe2, groupe3, groupe4])
      etiquettes_groupes = ['groupe1'] * len(groupe1) + ['groupe2'] * len(groupe2) +
      ↪['groupe3'] * len(groupe3) + ['groupe4'] * len(groupe4)

      tukey = pairwise_tukeyhsd(endog=donnees, groups=etiquettes_groupes, alpha=0.05)

      print(tukey)

```

```

Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj  lower  upper  reject
-----
groupe1 groupe2  0.0759 0.7181 -0.1124  0.2641  False
groupe1 groupe3 -0.1908 0.0132 -0.3515 -0.0301  True
groupe1 groupe4 -0.4456  0.0  -0.6414 -0.2499  True
groupe2 groupe3 -0.2667 0.0023 -0.4571 -0.0763  True
groupe2 groupe4 -0.5215  0.0  -0.7422 -0.3007  True

```

```
groupe3 groupe4 -0.2548 0.0059 -0.4526 -0.057 True
-----
```

```
[41]: # Test de Tukey 3 groupes

from statsmodels.stats.multicomp import pairwise_tukeyhsd

donnees = np.concatenate([groupe1, groupe2, groupe3])
etiquettes_groupes = ['groupe1'] * len(groupe1) + ['groupe2'] * len(groupe2) +
↳ ['groupe3'] * len(groupe3)

tukey = pairwise_tukeyhsd(endog=donnees, groups=etiquettes_groupes, alpha=0.05)

print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
-----
groupe1 groupe2 0.0759 0.5983 -0.1111 0.2628 False
groupe1 groupe3 -0.1908 0.0149 -0.3504 -0.0313 True
groupe2 groupe3 -0.2667 0.0033 -0.4557 -0.0777 True
-----
```

### 0.0.7 Disposition

Les situations dont la visibilité du motif visuel est perturbée sont écartés

```
[42]: dt_disposition = dt_loop[(dt_loop['motif_case']=='non') & (dt_loop['decor']!
↳ =='visuel')]
```

```
[43]: dt_disposition['disposition'].value_counts()
```

```
[43]: linéaire    60
1D              13
cyclique        9
Name: disposition, dtype: int64
```

```
[44]: dt_disposition_describe = dt_disposition.groupby('disposition')['all'].
↳ describe()
dt_disposition_describe['EI'] =
↳ dt_disposition_describe['75%']-dt_disposition_describe['25%']
dt_disposition_describe
```

```
[44]:
```

	count	mean	std	min	25%	50%	75%	max \
disposition								
1D	13.0	0.677077	0.266961	0.028	0.610	0.791	0.873	0.936
cyclique	9.0	0.244778	0.251113	0.034	0.068	0.140	0.416	0.705

```

linéaire      60.0  0.423667  0.265732  0.009  0.179  0.384  0.645  0.978

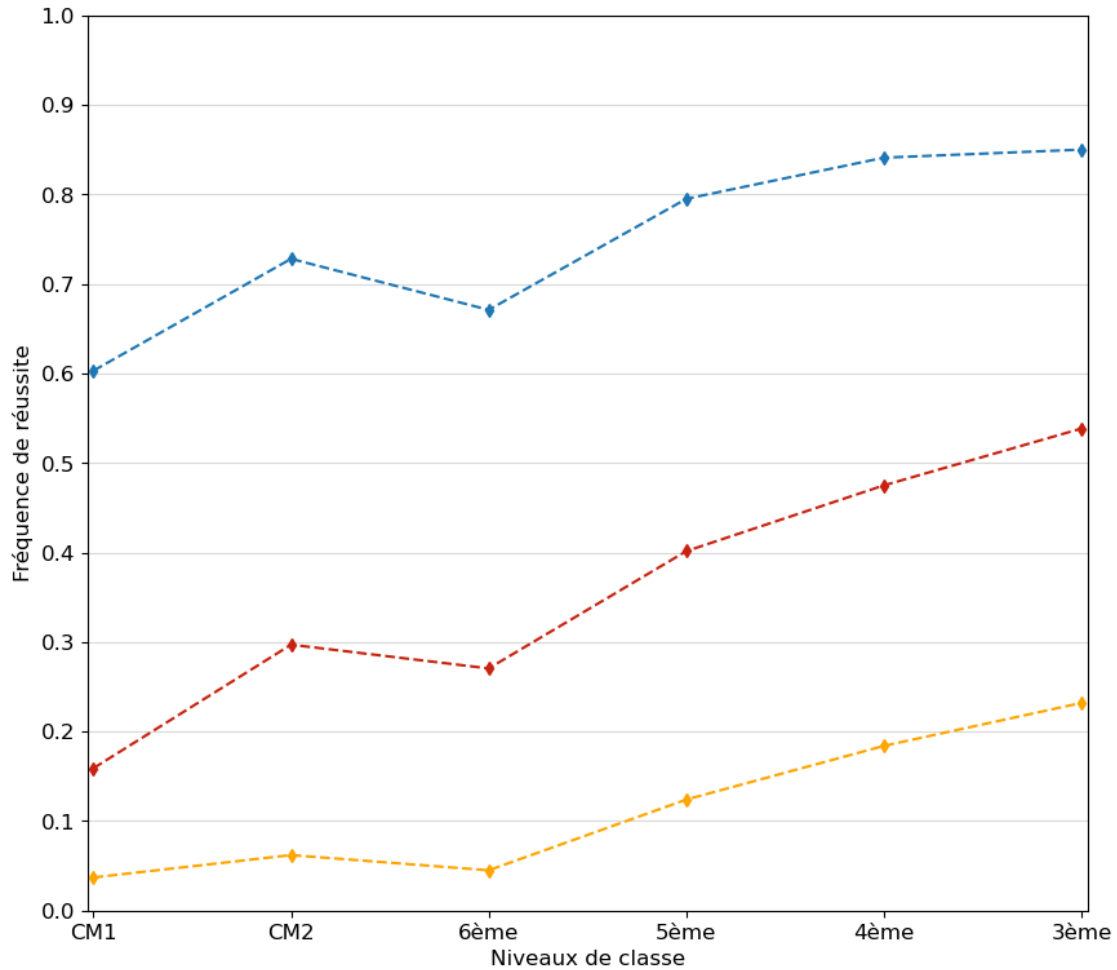
                EI
disposition
1D             0.263
cyclique      0.348
linéaire      0.466

```

```
[45]: dt_disposition.groupby('disposition')['length_program'].mean()
```

```
[45]: disposition
1D             5.923077
cyclique      13.000000
linéaire      12.016667
Name: length_program, dtype: float64
```

```
[46]: diagramme_reussite_categorie_puzzle(dt_disposition,
                                           'disposition',
                                           [('1D', '#1f77b4'),
                                            ('linéaire', '#c82210'),
                                            ('cyclique', 'orange')
                                           ],
                                           puzzles=[],
                                           #moyenne=True,
                                           ecart_interquartile=False,
                                           #nom_fichier = 'disposition_tous.png'
                                           )
```



```
[47]: groupe1 = dt_disposition[dt_disposition['disposition']=='1D']['all']
      groupe2 = dt_disposition[dt_disposition['disposition']=='linéaire']['all']
      groupe3 = dt_disposition[dt_disposition['disposition']=='cyclique']['all']
```

```
[48]: from scipy.stats import shapiro

num =1

for groupe in [groupe1, groupe2, groupe3] :
    print("Groupe " + str(num) + ":")
    # Test de Shapiro-Wilk pour la normalité
    statistique_shapiro, p_value_shapiro = shapiro(groupe)

    print("Statistique de Shapiro-Wilk :", statistique_shapiro)
    print("p-value :", p_value_shapiro)

    # Interprétation
```

```

if p_value_shapiro > 0.05:
    print("Les données semblent suivre une distribution normale.")
else:
    print("Les données ne suivent pas une distribution normale.")

num+=1

```

Groupe 1:  
 Statistique de Shapiro-Wilk : 0.8499878644943237  
 p-value : 0.02849447727203369  
 Les données ne suivent pas une distribution normale.

Groupe 2:  
 Statistique de Shapiro-Wilk : 0.9542176127433777  
 p-value : 0.024696094915270805  
 Les données ne suivent pas une distribution normale.

Groupe 3:  
 Statistique de Shapiro-Wilk : 0.8023068904876709  
 p-value : 0.02169734612107277  
 Les données ne suivent pas une distribution normale.

```

[49]: # Test de Kruskal-Wallis

from scipy.stats import kruskal

statistique_kw, p_value_kw = kruskal(groupe1, groupe2, groupe3)

print("Statistique de Kruskal-Wallis :", statistique_kw)
print("p-value :", p_value_kw)

```

Statistique de Kruskal-Wallis : 13.130091051805344  
 p-value : 0.001408759814383541

```

[50]: # Calcul de la variance pour chaque groupe

num = 1

for groupe in [groupe1, groupe2, groupe3] :
    variance = np.var(groupe, ddof=1)
    print("Variance du groupe " + str(num) + ":", variance)
    num+=1

```

Variance du groupe 1: 0.07126841025641027  
 Variance du groupe 2: 0.07061361581920902  
 Variance du groupe 3: 0.06305769444444444

Les variances sont comparables (rapports proches de 1) donc un test de Tukey est applicable

```
[51]: # Test de Tukey

from statsmodels.stats.multicomp import pairwise_tukeyhsd

donnees = np.concatenate([groupe1, groupe2, groupe3])
etiquettes_groupes = ['groupe1'] * len(groupe1) + ['groupe2'] * len(groupe2) +
↳ ['groupe3'] * len(groupe3)

tukey = pairwise_tukeyhsd(endog=donnees, groups=etiquettes_groupes, alpha=0.05)

print(tukey)
```

```
Multiple Comparison of Means - Tukey HSD, FWER=0.05
=====
group1 group2 meandiff p-adj lower upper reject
-----
groupe1 groupe2 -0.2534 0.0068 -0.4467 -0.0601 True
groupe1 groupe3 -0.4323 0.0009 -0.7062 -0.1584 True
groupe2 groupe3 -0.1789 0.1476 -0.4047 0.0469 False
-----
```

### Cas particulier des cases adjacentes ne faisant pas partie du même motif

```
[52]: dt_adj_diff = dt_loop[(dt_loop['motif_case']=='non') &
↳ (dt_loop['classe_situation']!=5)]
len(dt_adj_diff)
```

[52]: 82

```
[53]: dt_adj_diff.groupby('adj_diff')['length_program'].mean()
```

```
[53]: adj_diff
non          10.166667
oui - elt    12.785714
oui - fond   11.562500
Name: length_program, dtype: float64
```

```
[54]: dt_adj_diff_describe = dt_adj_diff.groupby('adj_diff')['all'].describe()
dt_adj_diff_describe['EI'] =
↳ dt_adj_diff_describe['75%']-dt_adj_diff_describe['25%']
dt_adj_diff_describe
```

```
[54]:
```

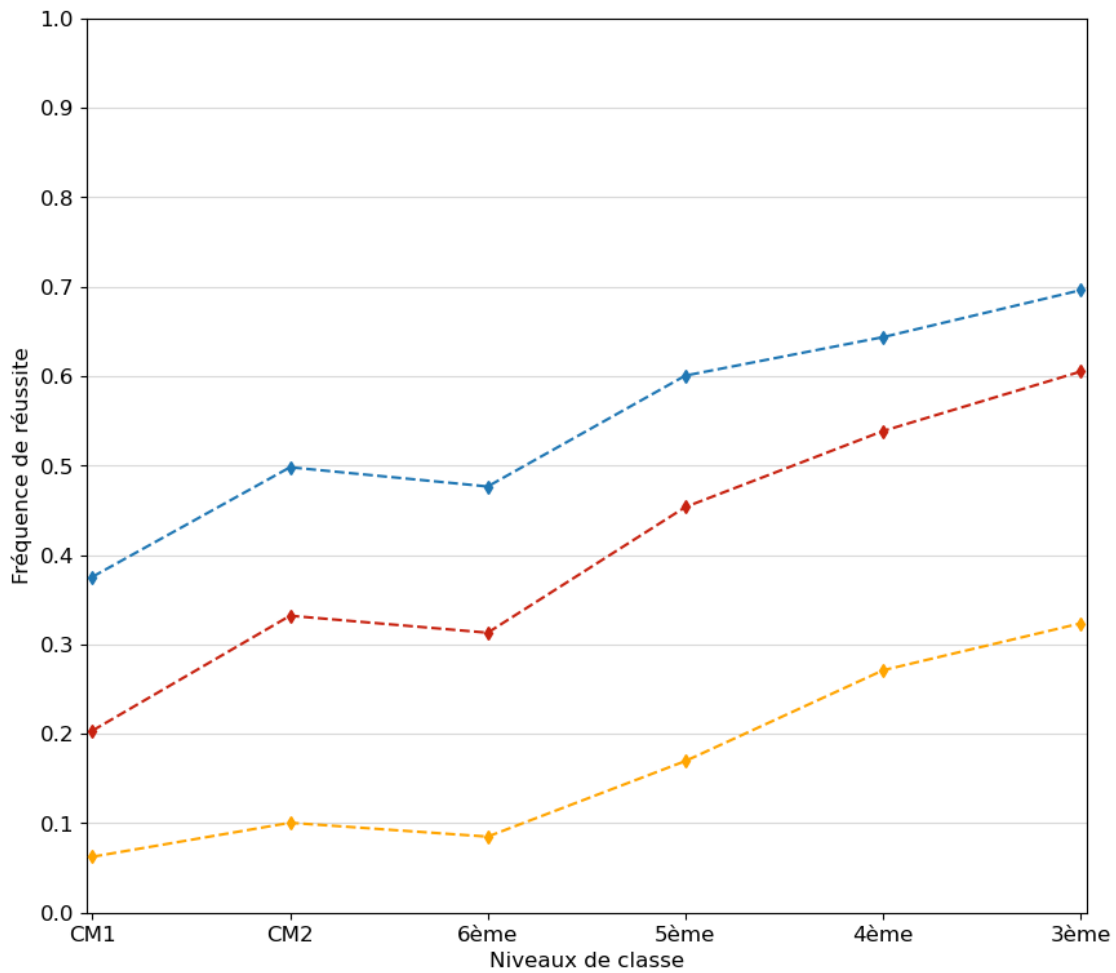
	count	mean	std	min	25%	50%	75%	max \
adj_diff								
non	36.0	0.528556	0.278196	0.009	0.29975	0.5855	0.77725	0.936
oui - elt	14.0	0.251000	0.237149	0.028	0.06875	0.1885	0.29800	0.800
oui - fond	32.0	0.433844	0.277784	0.031	0.16750	0.4555	0.63575	0.978

```

EI
adj_diff
non      0.47750
oui - elt 0.22925
oui - fond 0.46825

```

```
[55]: diagramme_reussite_categorie_puzzle(dt_loop[dt_loop['classe_situation']!=5],
      'adj_diff',
      [('non', '#1f77b4'),
       ('oui - fond', '#c82210'),
       ('oui - elt', 'orange')
      ],
      puzzles=[],
      #moyenne=True,
      ecart_interquartile=False,
      #nom_fichier='adj_diff_oui_elt.png'
    )
```





```
[56]: groupe1 = dt_adj_diff[dt_adj_diff['adj_diff']=='non']['all']
groupe2 = dt_adj_diff[dt_adj_diff['adj_diff']=='oui - fond']['all']
groupe3 = dt_adj_diff[dt_adj_diff['adj_diff']=='oui - elt']['all']
```

```
[57]: # Test de Shapiro-Wilk pour la normalité

from scipy.stats import shapiro

num =1

for groupe in [groupe1, groupe2, groupe3] :
    print("Groupe " + str(num) + ":")

    statistique_shapiro, p_value_shapiro = shapiro(groupe)

    # Afficher les résultats
    print("Statistique de Shapiro-Wilk :", statistique_shapiro)
    print("p-value :", p_value_shapiro)

    # Interprétation
    if p_value_shapiro > 0.05:
        print("Les données semblent suivre une distribution normale.")
    else:
        print("Les données ne suivent pas une distribution normale.")

    num+=1
```

Groupe 1:  
Statistique de Shapiro-Wilk : 0.9395002722740173  
p-value : 0.04901392385363579  
Les données ne suivent pas une distribution normale.

Groupe 2:  
Statistique de Shapiro-Wilk : 0.9477495551109314  
p-value : 0.12437868118286133  
Les données semblent suivre une distribution normale.

Groupe 3:  
Statistique de Shapiro-Wilk : 0.8415133953094482  
p-value : 0.017068540677428246  
Les données ne suivent pas une distribution normale.

La distribution du groupe 2 est normale, celle du groupe 1 se rapproche d'une distribution normale, mais pas celle du groupe 3.

```
[58]: # Test de Kruskal-Wallis

from scipy.stats import kruskal
```

```

statistique_kw, p_value_kw = kruskal(groupe1, groupe2, groupe3)

print("Statistique de Kruskal-Wallis :", statistique_kw)
print("p-value :", p_value_kw)

```

Statistique de Kruskal-Wallis : 9.744362973760918  
p-value : 0.007656644187252965

```

[59]: # Calcul de la variance pour chaque groupe

num = 1

for groupe in [groupe1, groupe2, groupe3] :
    variance = np.var(groupe, ddof=1)
    print("Variance du groupe " + str(num) + ":", variance)
    num+=1

```

Variance du groupe 1: 0.07739299682539683  
Variance du groupe 2: 0.07716381350806452  
Variance du groupe 3: 0.05623953846153847

Les variances sont comparables donc un test de Tukey est applicable.

```

[60]: # Test de Tukey

from statsmodels.stats.multicomp import pairwise_tukeyhsd

donnees = np.concatenate([groupe1, groupe2, groupe3])
etiquettes_groupes = ['groupe1'] * len(groupe1) + ['groupe2'] * len(groupe2) +
    → ['groupe3'] * len(groupe3)

tukey = pairwise_tukeyhsd(endog=donnees, groups=etiquettes_groupes, alpha=0.05)

print(tukey)

```

Multiple Comparison of Means - Tukey HSD, FWER=0.05  
=====

group1	group2	meandiff	p-adj	lower	upper	reject
-----	-----	-----	-----	-----	-----	-----
groupe1	groupe2	-0.0947	0.3281	-0.2524	0.063	False
groupe1	groupe3	-0.2776	0.0049	-0.482	-0.0731	True
groupe2	groupe3	-0.1828	0.0963	-0.3908	0.0251	False
-----	-----	-----	-----	-----	-----	-----

### 0.0.8 Motif algorithmique

#### Fréquence de réussite en fonction de la taille du motif algorithmique

```
[61]: plt.figure(figsize=(12, 9))
fig, ax = plt.subplots()

sns.scatterplot(data=dt_loop, x="length_motif", y="all", size=15, zorder=3,
↳ legend=False)

ax.set_xlabel('Nombre de blocs du motif algorithmique')
ax.set_ylabel('Fréquence de réussite')

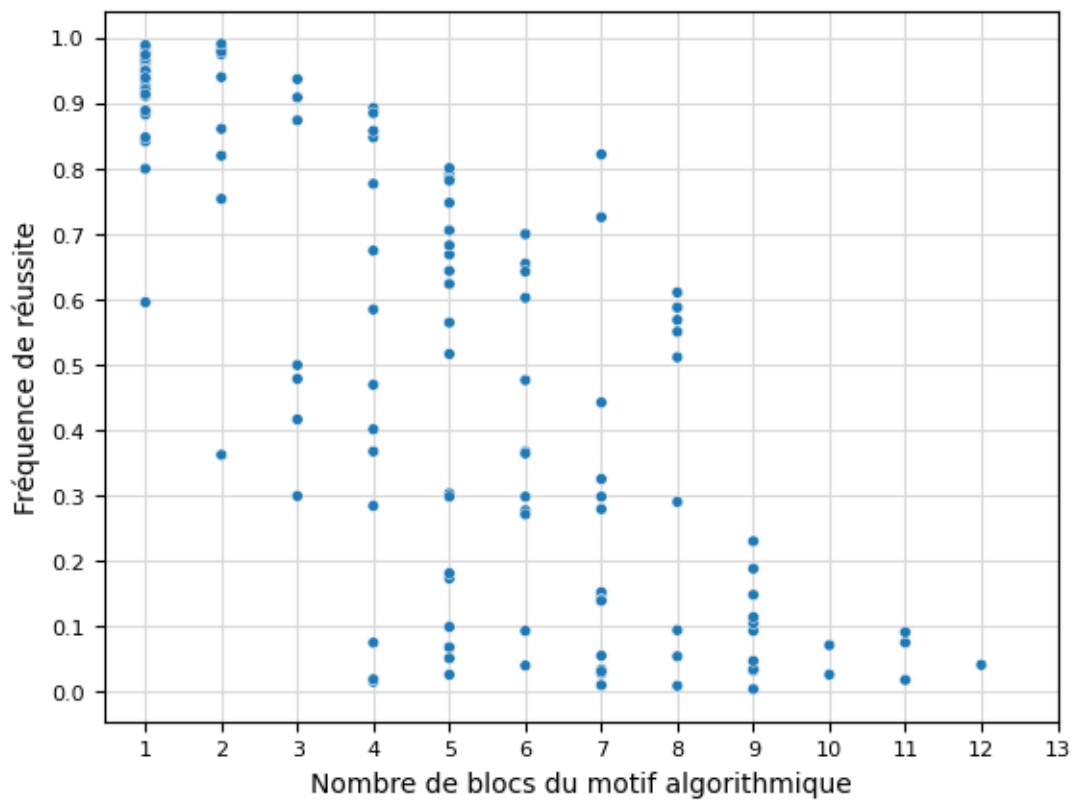
plt.xticks(range(1,14))
plt.yticks(np.arange(0, 1.05, 0.1))
plt.tick_params(labelsize=8)

ax.grid(True , color='#d9d9d9', zorder=0)

#plt.savefig('taille_motif_algorithmique.pdf')

plt.show()
```

<Figure size 1200x900 with 0 Axes>



Corrélation entre taille du motif algorithmique et fréquence de réussite

```
[62]: # Coefficient de corrélation de Pearson
coefficent_correlation, valeur_p = st.pearsonr(dt_loop["length_motif"],
→dt_loop['all'])

print("Coefficient de corrélation de Pearson :", coefficent_correlation)
print("Valeur p_value :", valeur_p)
```

```
Coefficient de corrélation de Pearson : -0.7825170050224153
Valeur p_value : 5.581843946506549e-30
```

**Système d'orientation du robot virtuel** Pour les grilles où le robot ne peut se déplacer que dans une seule direction, nous les plaçons dans la même catégorie que celles en orientation absolue. En effet, ce qui différencie principalement les déplacements du robot dans un système d'orientation relative par rapport à un système d'orientation absolue sont les actions de pivotement. Or, même si le robot est orienté, ces actions de pivotement sont absentes lorsque le robot ne se déplace que dans une seule direction. Nous distinguons donc deux classes de situations selon la nécessité ou non de programmer des actions de pivotement du robot virtuel. La modalité 'absolu' correspond à l'absence de pivotement, alors que la modalité 'relatif' correspond à la nécessité d'instructions de pivotement.

```
[63]: dt_loop['orientation_ajustee'] = dt_loop['orientation']
dt_loop.loc[dt_loop['orientation']=='unidirectionnel', 'orientation_ajustee'] =
→'absolu'
dt_loop.loc[dt_loop['orientation']=='relatif-latéral', 'orientation_ajustee'] =
→'absolu'
dt_loop['orientation_ajustee'].value_counts()
```

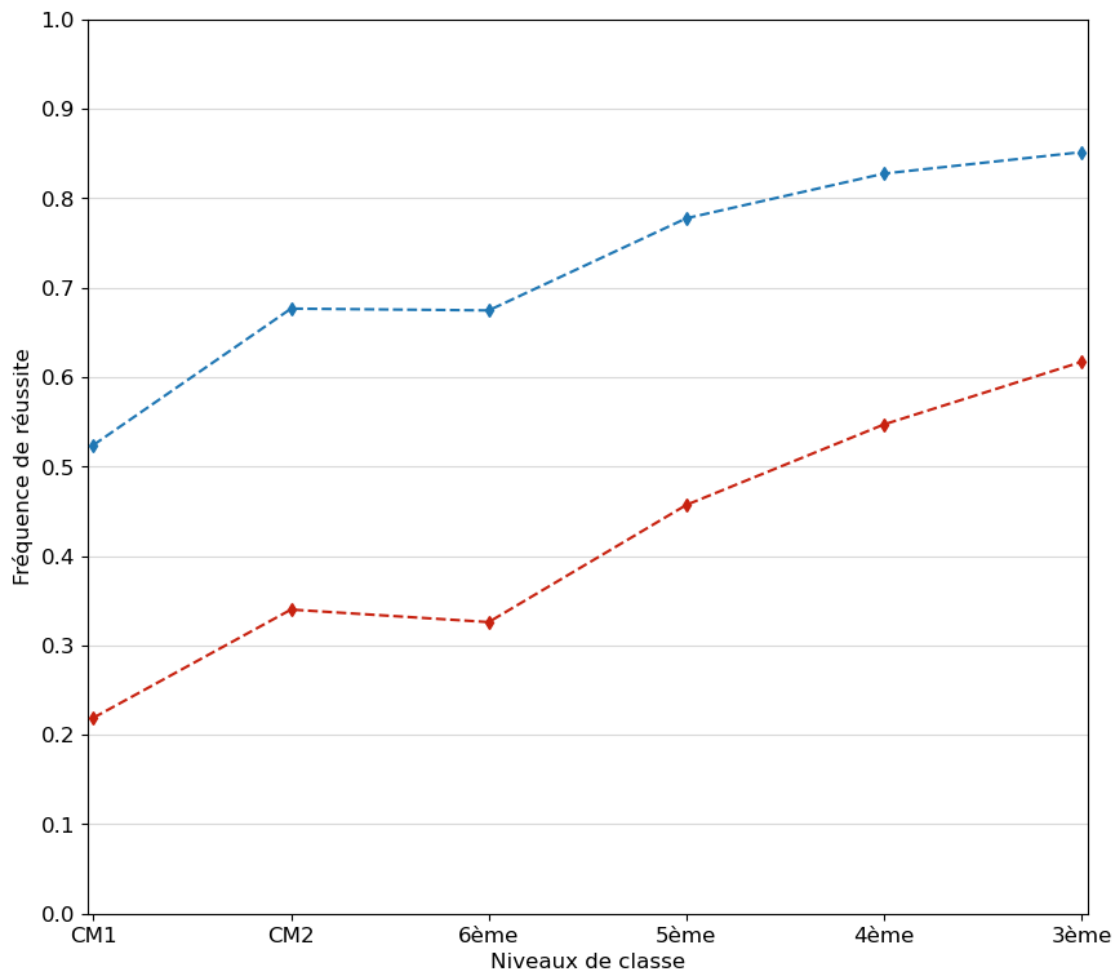
```
[63]: relatif      83
absolu       56
Name: orientation_ajustee, dtype: int64
```

```
[64]: dt_loop_describe = dt_loop.groupby('orientation_ajustee')['all'].describe()
dt_loop_describe['EI'] = dt_loop_describe['75%']-dt_loop_describe['25%']
dt_loop_describe
```

```
[64]:
```

	count	mean	std	min	25%	50%	\
orientation_ajustee							
absolu	56.0	0.626482	0.352398	0.009	0.29325	0.7645	
relatif	83.0	0.478084	0.347240	0.004	0.12700	0.4690	
		75%	max	EI			
orientation_ajustee							
absolu	0.9515	0.990	0.65825				
relatif	0.8310	0.986	0.70400				

```
[65]: diagramme_reussite_categorie_puzzle(dt_loop,
                                         'orientation_ajustee',
                                         [('absolu', '#1f77b4'),
                                          ('relatif', '#c82210')],
                                         ],
                                         puzzles=[],
                                         #moyenne=True,
                                         ecart_interquartile=False,
                                         #nom_fichier='adj_diff_oui_elt.png'
                                         )
```



```
[66]: dt_orientation = dt_loop.groupby('orientation_ajustee')
```

```
[67]: # Test de Shapiro-Wilk
from scipy.stats import shapiro
```

```
resultats_shapiro = dt_orientation['all'].apply(lambda x: shapiro(x))

print(resultats_shapiro)
```

```
orientation_ajustee
absolu      (0.8441047072410583, 3.918847141903825e-06)
relatif     (0.8924948573112488, 4.20018386648735e-06)
Name: all, dtype: object
```

```
[68]: # Test de Mann-Whitney U

from scipy.stats import mannwhitneyu

groupes = [groupe[1]['all'].values for groupe in dt_orientation]

resultats_mannwhitneyu = mannwhitneyu(*groupes)

print(resultats_mannwhitneyu)
```

```
MannwhitneyuResult(statistic=2947.5, pvalue=0.007463860415384455)
```

### Instructions en-dehors de la boucle

```
[69]: dt_instructions_hors_boucle = dt_loop[(dt_loop['motif_case']=='non') &
      ((dt_loop['after_loop']=='Oui') |
       (dt_loop['before_loop']=='Oui'))
      ]
dt_non_hors_boucle = dt_loop[(dt_loop['motif_case']=='non') &
      (dt_loop['after_loop']=='Non') &
      (dt_loop['before_loop']=='Non')
      ]
```

```
[70]: len(dt_instructions_hors_boucle)
```

```
[70]: 68
```

```
[71]: len(dt_non_hors_boucle)
```

```
[71]: 30
```

```
[72]: dt_instructions_hors_boucle['length_program'].mean()
```

```
[72]: 14.897058823529411
```

```
[73]: dt_non_hors_boucle['length_program'].mean()
```

```
[73]: 6.3
```

```
[74]: dt_oui_avant_oui_après = dt_loop[(dt_loop['motif_case']=='non') &
                                         (dt_loop['after_loop']=='Oui') &
                                         (dt_loop['before_loop']=='Oui')
                                         ]
```

```
[75]: dt_oui_avant_oui_après['all'].describe()
```

```
[75]: count      52.000000
      mean       0.238865
      std        0.218306
      min        0.004000
      25%        0.054000
      50%        0.162500
      75%        0.375500
      max        0.819000
      Name: all, dtype: float64
```

```
[76]: dt_oui_avant_non_après = dt_loop[(dt_loop['motif_case']=='non') &
                                         (dt_loop['after_loop']=='Non') &
                                         (dt_loop['before_loop']=='Oui')
                                         ]
```

```
[77]: dt_oui_avant_non_après['all'].describe()
```

```
[77]: count      9.000000
      mean       0.172778
      std        0.165833
      min        0.009000
      25%        0.047000
      50%        0.140000
      75%        0.188000
      max        0.511000
      Name: all, dtype: float64
```

```
[78]: dt_non_avant_oui_après = dt_loop[(dt_loop['motif_case']=='non') &
                                         (dt_loop['after_loop']=='Oui') &
                                         (dt_loop['before_loop']=='Non')
                                         ]
```

```
[79]: dt_non_avant_oui_après['all'].describe()
```

```
[79]: count      7.000000
      mean       0.387857
      std        0.223559
      min        0.139000
      25%        0.212500
      50%        0.298000
```

```
75%      0.605000
max       0.643000
Name: all, dtype: float64
```

```
[80]: dt_non_avant_non_après = dt_loop[(dt_loop['motif_case']=='non') &
                                         (dt_loop['after_loop']=='Non') &
                                         (dt_loop['before_loop']=='Non')]
                                         ]
```

```
[81]: dt_non_avant_non_après['all'].describe()
```

```
[81]: count      30.000000
      mean       0.686633
      std        0.227063
      min        0.028000
      25%        0.618000
      50%        0.715000
      75%        0.840500
      max        0.978000
      Name: all, dtype: float64
```

```
[82]: dt_loop[(dt_loop['classe_situation']!=1) & (dt_loop['after_loop']=='Non')].
      ↪groupby('before_loop')['length_program'].mean()
```

```
[82]: before_loop
      Non      6.300000
      Oui     11.888889
      Name: length_program, dtype: float64
```

```
[83]: dt_loop[(dt_loop['classe_situation']!=1) & (dt_loop['before_loop']=='Oui')].
      ↪groupby('after_loop')['length_program'].mean()
```

```
[83]: after_loop
      Non     11.888889
      Oui     15.846154
      Name: length_program, dtype: float64
```

```
[84]: dt_loop[(dt_loop['classe_situation']!=1) &
      ↪(dt_loop['before_loop']=='Oui')]['after_loop'].value_counts()
```

```
[84]: Oui      52
      Non      9
      Name: after_loop, dtype: int64
```

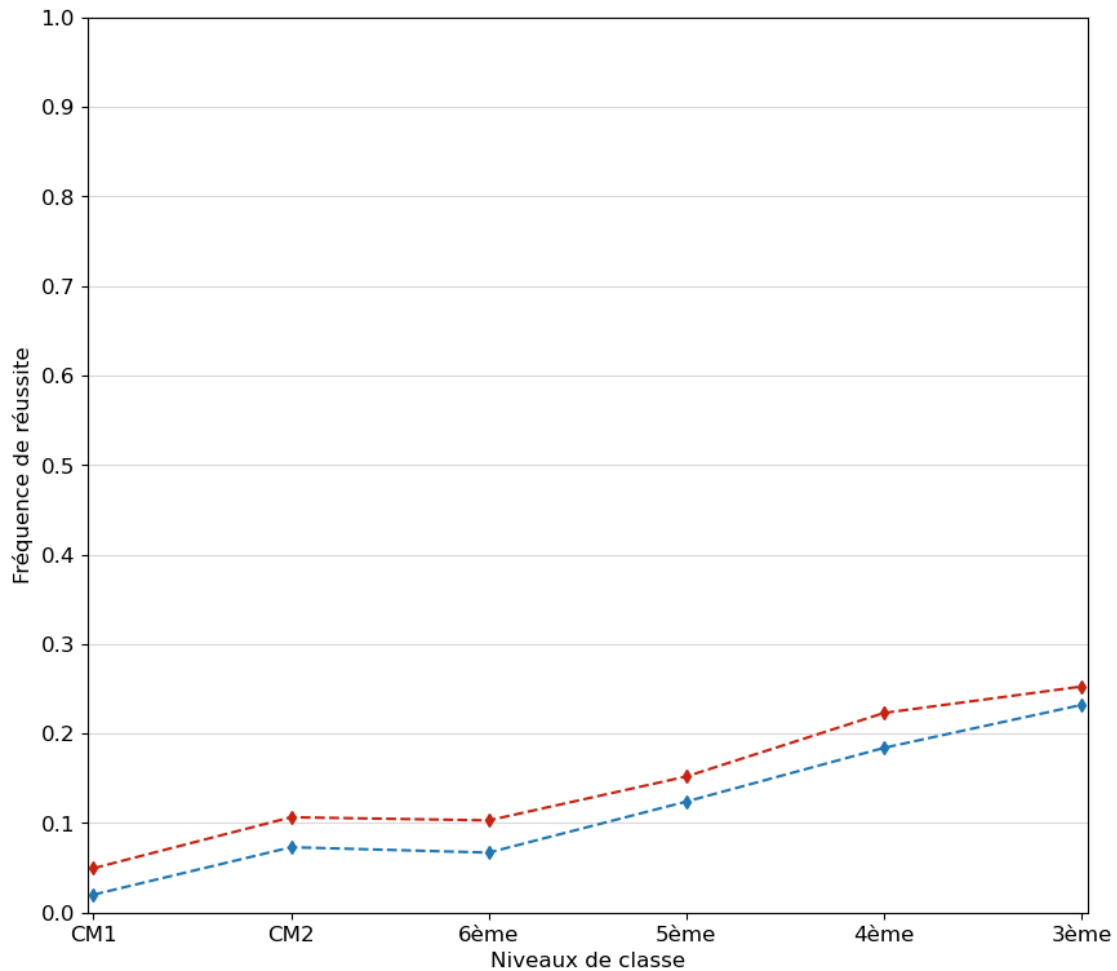
```
[85]: diagramme_reussite_categorie_puzzle(dt_loop[(dt_loop['classe_situation']!=1) &
      ↪(dt_loop['before_loop']=='Oui')],
      'after_loop',
```



```

[('Non', '#1f77b4'),
 ('Oui', '#c82210'),
 ],
 puzzles=[],
 #moyenne=True,
 ecart_interquartile=False,
 #nom_fichier = 'instr_oui_avant_toutes.png'
 )

```



La présence d'instructions après la boucle n'ajoute pas de difficulté supplémentaire.

### 0.0.9 Régression linéaire

```

[86]: dt_loop['decor_ajuste'] = dt_loop['decor']
      dt_loop.loc[dt_loop['adj_diff']=='oui - elt', 'decor_ajuste'] = 'visuel'

```

```
[87]: dt_loop['hors_boucle'] = 'Non'
dt_loop.loc[(dt_loop['before_loop']=='oui') | (dt_loop['after_loop']=='Oui'),
↳'hors_boucle'] = 'Oui'
```

```
[88]: dt_loop['hors_boucle'].value_counts()
```

```
[88]: Oui    95
Non     44
Name: hors_boucle, dtype: int64
```

```
[89]: target_name = "all"
target = dt_loop[target_name]
data = dt_loop[['disposition',
                'decor_ajuste',
                'orientation_ajustee',
                'hors_boucle',
                'length_program',
                'length_motif'
                ]]
```

```
[90]: categorical_columns = ['disposition',
                            'hors_boucle',
                            'decor_ajuste',
                            'orientation_ajustee']
numerical_columns = ['length_program',
                    'length_motif'
                    ]
data_cat = data[categorical_columns]
data_num = data[numerical_columns]
```

```
[91]: from sklearn.preprocessing import OneHotEncoder, StandardScaler

cat_preprocessor = OneHotEncoder(handle_unknown="ignore")
num_preprocessor = StandardScaler()
```

```
[92]: from sklearn.compose import ColumnTransformer

preprocessor = ColumnTransformer([
    ('one-hot-encoder', cat_preprocessor, categorical_columns),
    ('standard_scaler', num_preprocessor, numerical_columns)])
```

```
[93]: from sklearn import set_config

set_config(display="diagram")
```

```
[94]: import numpy as np
from sklearn.pipeline import make_pipeline
```

```

from sklearn.linear_model import RidgeCV
from sklearn.linear_model import LinearRegression

alphas = np.linspace(0.1,20,20)

model = make_pipeline(
    preprocessor,
    #RidgeCV(alphas=alphas, store_cv_values=True)
    LinearRegression()
)
model

```

```

[94]: Pipeline(steps=[('columntransformer',
                       ColumnTransformer(transformers=[('one-hot-encoder',
                                                         OneHotEncoder(handle_unknown='ignore'),
                                                         ['disposition', 'hors_boucle',
                                                         'decor_ajuste',
                                                         'orientation_ajustee']),
                                                         ('standard_scaler',
                                                         StandardScaler(),
                                                         ['length_program',
                                                         'length_motif'])])),
                      ('linearregression', LinearRegression())])

```

```

[95]: from sklearn.metrics import make_scorer, r2_score
scorer = make_scorer(r2_score)

```

```

[103]: from sklearn.model_selection import ShuffleSplit, cross_validate

cv = ShuffleSplit(n_splits=5, random_state=1)
cv_results = cross_validate(model, data, target,
                            cv=cv,
                            #scoring="neg_mean_squared_error",
                            scoring = make_scorer(r2_score),
                            return_train_score=True,
                            return_estimator=True, n_jobs=2)

```

```

[104]: train_error = -cv_results["train_score"]
print(f"Erreur quadratique moyenne pour les données d'entraînement :\n"
      f"{train_error.mean():.2f} ± {train_error.std():.3f}")

```

Erreur quadratique moyenne pour les données d'entraînement :  
0.02 ± 0.001

```

[105]: test_error = -cv_results["test_score"]
print(f"Erreur quadratique moyenne pour les données de test :\n"
      f"{test_error.mean():.2f} ± {test_error.std():.3f}")

```

Erreur quadratique moyenne pour les données de test :  
0.03 ± 0.010

```
[101]: train_r2 = cv_results["train_score"]
print("Coefficient de détermination moyen (R²) pour les données d'entraînement :
↪", train_r2.mean())
```

Coefficient de détermination moyen (R²) pour les données d'entraînement :  
0.8232024824165378

```
[102]: test_r2 = cv_results["test_score"]
print("Coefficient de détermination moyen (R²) pour les données de test :",
↪test_r2.mean())
```

Coefficient de détermination moyen (R²) pour les données de test :  
0.5846973590992501

```
[106]: model_first_fold = cv_results["estimator"][0]
model_first_fold
```

```
[106]: Pipeline(steps=[('columntransformer',
                        ColumnTransformer(transformers=[('one-hot-encoder',
                                                         OneHotEncoder(handle_unknown='ignore'),
                                                         ['disposition', 'hors_boucle',
                                                         'decor_ajuste',
                                                         'orientation_ajustee'])),
                        ('standard_scaler',
                        StandardScaler(),
                        ['length_program',
                        'length_motif'])))],
                ('linearregression', LinearRegression()))
```

```
[107]: feature_names = model_first_fold[0].get_feature_names_out(
input_features=data.columns)
feature_names
```

```
[107]: array(['one-hot-encoder__disposition_1 case',
            'one-hot-encoder__disposition_1D',
            'one-hot-encoder__disposition_cyclique',
            'one-hot-encoder__disposition_linéaire',
            'one-hot-encoder__hors_boucle_Non',
            'one-hot-encoder__hors_boucle_Oui',
            'one-hot-encoder__decor_ajuste_non',
            'one-hot-encoder__decor_ajuste_parcours',
            'one-hot-encoder__decor_ajuste_présent',
            'one-hot-encoder__decor_ajuste_visuel',
            'one-hot-encoder__orientation_ajustee_absolu',
            'one-hot-encoder__orientation_ajustee_relatif',
```

```
'standard_scaler__length_program', 'standard_scaler__length_motif'],  
dtype=object)
```

```
[108]: new_names = [  
    'motif sur une seule case',  
    'motif sur plusieurs cases avec disposition 1D',  
    'motif sur plusieurs cases avec disposition cyclique',  
    'motif sur plusieurs cases avec disposition linéaire en 2D',  
    "pas d'instruction hors de la boucle",  
    "présence d'instructions hors de la boucle",  
    "pas d'éléments de décor",  
    "éléments de décor qui contraignent le parcours du robot",  
    "éléments de décor présents sans contraindre le parcours du robot",  
    "perturbation visuelle pour l'identification du motif",  
    "pas de pivotement nécessaire",  
    "nécessité de pivotement(s) du robot virtuel",  
    "nombre d'instructions de la solution de référence",  
    "nombre d'instructions du motif algorithmique"  
]
```

```
[109]: coefs = [est[-1].coef_ for est in cv_results["estimator"]]  
weights_linear_regression = pd.DataFrame(coefs, columns=new_names)  
weights_linear_regression
```

```
[109]: motif sur une seule case motif sur plusieurs cases avec disposition 1D \  
0          0.111538          0.041014  
1          0.156309          0.022305  
2          0.129764          0.040438  
3          0.110411          0.048592  
4          0.116135          0.018434  
  
motif sur plusieurs cases avec disposition cyclique \  
0          -0.142846  
1          -0.151293  
2          -0.148950  
3          -0.138947  
4          -0.127358  
  
motif sur plusieurs cases avec disposition linéaire en 2D \  
0          -0.009706  
1          -0.027321  
2          -0.021252  
3          -0.020057  
4          -0.007210  
  
pas d'instruction hors de la boucle \  
0          0.035996
```

1	0.045259
2	0.030765
3	0.013672
4	0.031504

	présence d'instructions hors de la boucle	pas d'éléments de décor	\
0	-0.035996	0.045569	
1	-0.045259	0.016774	
2	-0.030765	0.046079	
3	-0.013672	0.059021	
4	-0.031504	0.037074	

	éléments de décor qui contraignent le parcours du robot	\
0	0.068028	
1	0.099174	
2	0.081887	
3	0.088779	
4	0.083887	

	éléments de décor présents sans contraindre le parcours du robot	\
0	0.013749	
1	0.026104	
2	0.004426	
3	-0.007978	
4	0.034570	

	perturbation visuelle pour l'identification du motif	\
0	-0.127346	
1	-0.142052	
2	-0.132393	
3	-0.139823	
4	-0.155532	

	pas de pivotement nécessaire	nécessité de pivotement(s) du robot virtuel	\
0	0.017572	-0.017572	
1	0.032455	-0.032455	
2	0.019851	-0.019851	
3	0.024355	-0.024355	
4	0.022985	-0.022985	

	nombre d'instructions de la solution de référence	\
0	-0.100646	
1	-0.089078	
2	-0.105070	
3	-0.114410	
4	-0.095628	

```

    nombre d'instructions du motif algorithmique
0          -0.148910
1          -0.138916
2          -0.133408
3          -0.123441
4          -0.143552

```

```
[110]: weights_linear_regression.mean().sort_values()
```

```

[110]: motif sur plusieurs cases avec disposition cyclique          -0.141879
perturbation visuelle pour l'identification du motif          -0.139429
nombre d'instructions du motif algorithmique                  -0.137645
nombre d'instructions de la solution de référence            -0.100967
présence d'instructions hors de la boucle                   -0.031439
nécessité de pivotement(s) du robot virtuel                 -0.023444
motif sur plusieurs cases avec disposition linéaire en 2D    -0.017109
éléments de décor présents sans contraindre le parcours du robot 0.014174
pas de pivotement nécessaire                                 0.023444
pas d'instruction hors de la boucle                         0.031439
motif sur plusieurs cases avec disposition 1D                0.034157
pas d'éléments de décor                                     0.040904
éléments de décor qui contraignent le parcours du robot    0.084351
motif sur une seule case                                    0.124831
dtype: float64

```

```
[111]: list_sort = list(weights_linear_regression.mean().sort_values(ascending=False).
↳index)
```

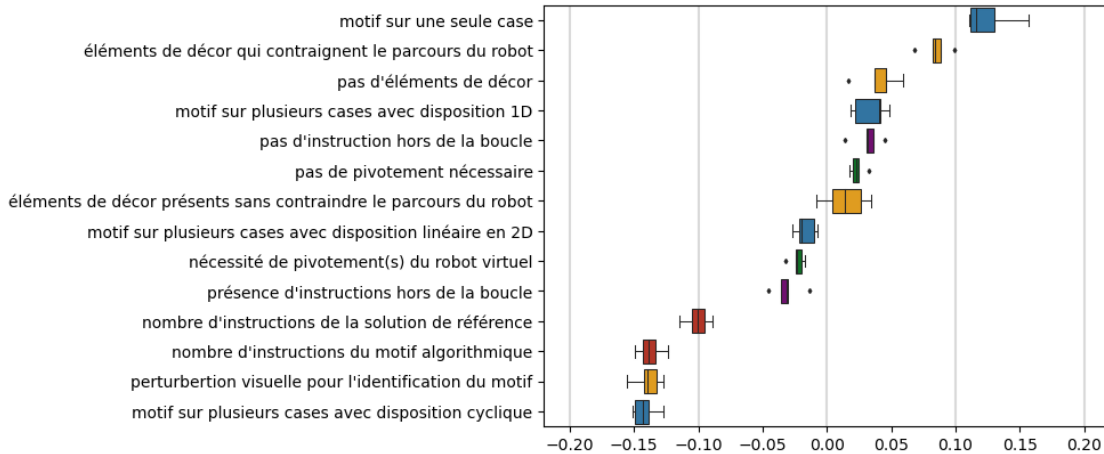
```
[112]: weights_linear_regression = weights_linear_regression[list_sort]
```

```

[113]: sns.boxplot(data=weights_linear_regression,
    orient='h',
    linewidth=0.75,
    fliersize=2,
    palette=["#1f77b4", "#ffa500", "#ffa500", "#1f77b4", "#800080",
↳"#00801f", "#ffa500", "#1f77b4", "#00801f",
    "#800080", "#c82210", "#c82210", "#ffa500", "#1f77b4"],
    zorder=3
    )
#c6dcec
positions = [-0.2, -0.1, 0.0, 0.1, 0.2]
for pos in positions:
    plt.axvline(x=pos, color='#d9d9d9', zorder=0)

plt.savefig('coef_regression_lineaire.png', bbox_inches='tight')
plt.show()

```



Coefficient de détermination qui indique la part de variance expliquée par le modèle

### 0.0.10 Classes de situations suite à l'analyse des coefficients de la régression linéaire

```
[139]: dt_loop['classes_ajustees'] = dt_loop['classe_situation']
dt_loop.loc[dt_loop['adj_diff']=='oui - elt', 'classes_ajustees'] = 4
dt_loop.loc[dt_loop['classe_situation']==5, 'classes_ajustees'] = 4
dt_situations = dt_loop.groupby('classes_ajustees')
```

```
[140]: dt_situations['classes_ajustees'].value_counts()
```

```
[140]: classes_ajustees  classes_ajustees
1                    1            41
2                    2            31
3                    3            30
4                    4            37
Name: classes_ajustees, dtype: int64
```

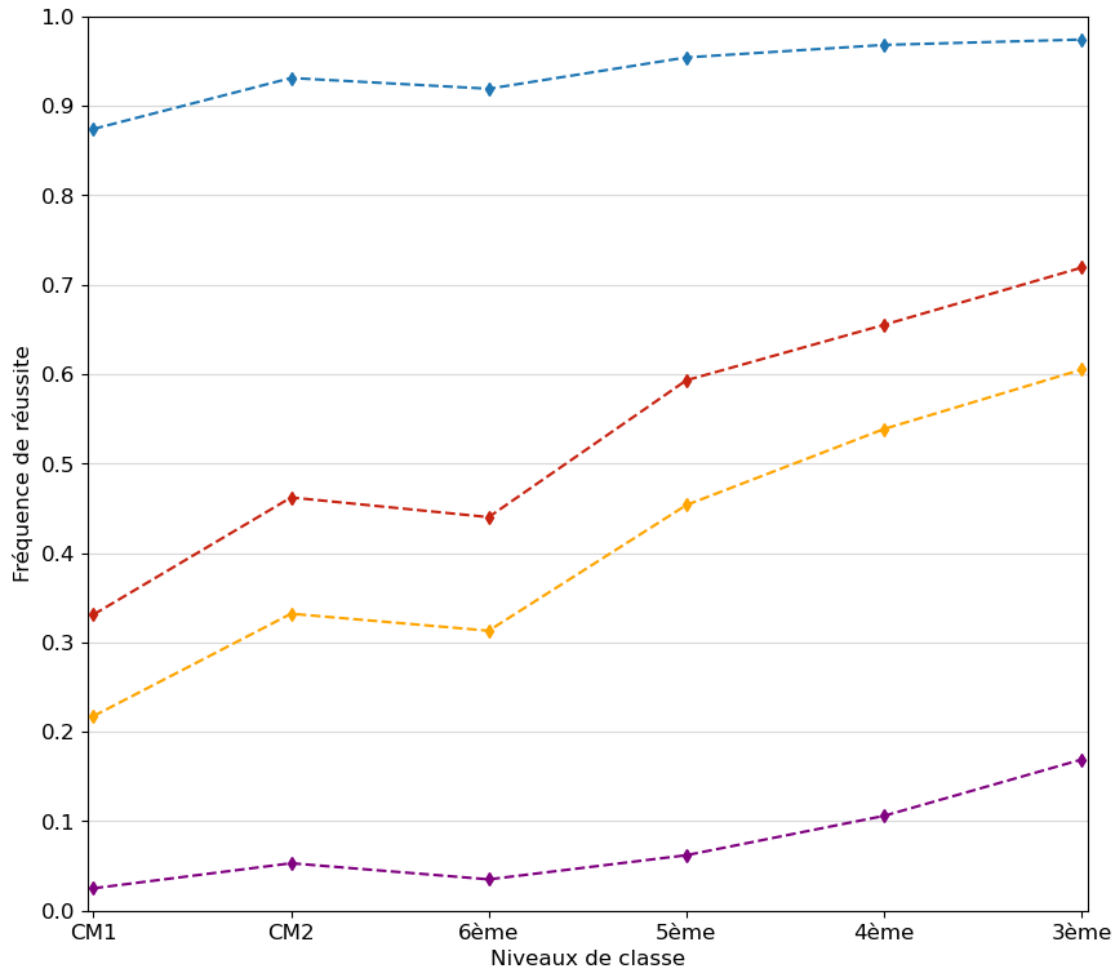
```
[156]: diagramme_reussite_categorie_puzzle(dt_loop,
                                           'classes_ajustees',
                                           [(1, '#1f77b4'),
                                            (2, '#c82210'),
                                            (3, 'orange'),
                                            (4, 'purple')
                                           ],
                                           puzzles=[#(24, '#1f77b4'),
                                                    #(7, '#c82210'),
                                                    #(16, 'orange'),
                                                    #(21, 'purple')
                                           ],
                                           #moyenne=True,
```



```

ecart_interquartile=False,
#nom_fichier=''
)

```



```
[151]: dt_loop[dt_loop['Numero problème']==5]
```

```
[151]:
```

	id_situation	questionId	sur	chtcodes	Version	Référence interne	\
9	10	868967553868256517	basic	white-05-course			
10	11	868967553868256517	easy	white-05-course			
11	12	41449531262818974	basic	white-05-flowers			
12	13	41449531262818974	easy	white-05-flowers			
13	14	374286687759083351	easy	white-05-gems			
15	16	41449531262818974	medium	white-05-flowers			
27	28	374286687759083351	medium	white-05-gems			
28	29	374286687759083351	hard	white-05-gems			
32	33	41449531262818974	hard	white-05-flowers			
38	39	868967553868256517	medium	white-05-course			

108	109	169929086869040532	basic	white-05-gem-2
109	110	169929086869040532	easy	white-05-gem-2
110	111	169929086869040532	medium	white-05-gem-2
111	112	169929086869040532	hard	white-05-gem-2
194	195	74670111826351495	basic	white-05-veterinary-2
195	196	74670111826351495	easy	white-05-veterinary-2
196	197	74670111826351495	medium	white-05-veterinary-2
197	198	74670111826351495	hard	white-05-veterinary-2
218	219	748042633662361316	basic	white-05-space
219	220	748042633662361316	easy	white-05-space
220	221	748042633662361316	medium	white-05-space
221	222	748042633662361316	hard	white-05-space

	Année	Tour	Catégorie	Numero problème	Effectif_all	all	...	\
9	2021	3	blanche	5	11158	0.978	...	
10	2021	3	blanche	5	11158	0.892	...	
11	2021	2	blanche	5	24511	0.784	...	
12	2021	2	blanche	5	24511	0.654	...	
13	2021	1	blanche	5	72835	0.682	...	
15	2021	2	blanche	5	24511	0.298	...	
27	2021	1	blanche	5	72835	0.284	...	
28	2021	1	blanche	5	72835	0.015	...	
32	2021	2	blanche	5	24511	0.010	...	
38	2021	3	blanche	5	11158	0.819	...	
108	2022	1	blanche	5	84206	0.857	...	
109	2022	1	blanche	5	84206	0.700	...	
110	2022	1	blanche	5	84206	0.469	...	
111	2022	1	blanche	5	84206	0.031	...	
194	2022	2	blanche	5	36924	0.821	...	
195	2022	2	blanche	5	36924	0.623	...	
196	2022	2	blanche	5	36924	0.099	...	
197	2022	2	blanche	5	36924	0.004	...	
218	2022	3	blanche	5	16967	0.699	...	
219	2022	3	blanche	5	16967	0.587	...	
220	2022	3	blanche	5	16967	0.298	...	
221	2022	3	blanche	5	16967	0.047	...	

	length_motif	length_program	before_loop	after_loop	\
9	2	3	Non	Non	
10	4	5	Non	Non	
11	5	6	Non	Non	
12	6	7	Non	Non	
13	5	6	Non	Non	
15	6	10	Oui	Oui	
27	4	16	Oui	Oui	
28	4	15	Oui	Oui	
32	7	23	Oui	Oui	

38	2	11	Oui	Oui
108	4	5	Non	Non
109	6	7	Non	Non
110	4	15	Oui	Oui
111	7	15	Oui	Oui
194	7	8	Non	Non
195	5	9	Non	Oui
196	5	12	Oui	Oui
197	9	22	Oui	Oui
218	6	7	Non	Non
219	8	10	Non	Oui
220	7	10	Non	Oui
221	9	11	Oui	Non

	last_instruction_loop		answer_ref	\
9	deplacement	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
10	deplacement	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
11	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
12	deplacement	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
13	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
15	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
27	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
28	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
32	deplacement	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
38	deplacement	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
108	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
109	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
110	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
111	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
194	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
195	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
196	deplacement	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
197	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
218	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
219	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
220	deplacement	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		
221	rotation	<xml xmlns="http://www.w3.org/1999/xhtml"><blo...		

	orientation_ajustee	decor_ajuste	hors_boucle	classes_ajustees
9	absolu	parcours	Non	2
10	absolu	parcours	Non	2
11	relatif	non	Non	3
12	relatif	non	Non	3
13	relatif	non	Non	3
15	relatif	visuel	Oui	4
27	relatif	visuel	Oui	4
28	relatif	visuel	Oui	4

32	relatif	visuel	Oui	4
38	absolu	parcours	Oui	2
108	relatif	parcours	Non	3
109	relatif	parcours	Non	3
110	relatif	parcours	Oui	3
111	relatif	parcours	Oui	3
194	relatif	présent	Non	3
195	relatif	présent	Oui	3
196	relatif	présent	Oui	3
197	relatif	visuel	Oui	4
218	relatif	présent	Non	3
219	relatif	présent	Oui	3
220	relatif	présent	Oui	3
221	relatif	présent	Non	3

[22 rows x 39 columns]

```
[143]: # Test de Shapiro-Wilk à tous les groupes

from scipy.stats import shapiro

resultats_shapiro = dt_situations['all'].apply(lambda x: shapiro(x))

print(resultats_shapiro)
```

```
classes_ajustees
1 (0.5906064510345459, 1.7374478522569348e-09)
2 (0.9524739980697632, 0.1826171576976776)
3 (0.9250361919403076, 0.03630509227514267)
4 (0.7539098858833313, 1.7442550870327977e-06)
Name: all, dtype: object
```

Seul le groupe 2 a une distribution normale

```
[144]: # Test de Kruskal-Wallis

from scipy.stats import kruskal

groupes = [groupe[1]['all'].values for groupe in dt_situations]

# Appliquer le test de Kruskal-Wallis
resultats_kruskal = kruskal(*groupes)

print(resultats_kruskal)
```

```
KruskalResult(statistic=91.46834362926711, pvalue=1.0596184873209828e-19)
```

```
[145]: # Calcul de la variance
dt_situations['all'].var(ddof=1)
```

```
[145]: classes_ajustees
1      0.013441
2      0.075493
3      0.071192
4      0.045229
Name: all, dtype: float64
```

Test de tukey

```
[146]: # Test de Tukey

from statsmodels.stats.multicomp import pairwise_tukeyhsd

resultats_tukey = pairwise_tukeyhsd(dt_loop['all'], dt_loop['classes_ajustees'])

print(resultats_tukey)
```

Multiple Comparison of Means - Tukey HSD, FWER=0.05

```
=====
group1 group2 meandiff p-adj  lower  upper  reject
-----
1      2  -0.3476    0.0 -0.4834 -0.2118  True
1      3  -0.4731    0.0 -0.6102 -0.336   True
1      4  -0.7382    0.0 -0.8676 -0.6088  True
2      3  -0.1255  0.1196 -0.2716  0.0207  False
2      4  -0.3906    0.0 -0.5295 -0.2517  True
3      4  -0.2651    0.0 -0.4053 -0.125   True
-----
```

## **Annexe 5 : étude de la robustesse des données**

## Étude de la robustesse des données à l'échelle nationale

Dans le tableau 1, nous relevons le nombre de participants individuels en catégorie blanche, en langage scratch, par tour, pour chaque niveau de classe.

Nous étudions la robustesse de ces données, c'est à dire que nous déterminons à partir de quels effectifs d'échantillon une différence de fréquence de réussite de 0,1 puis de 0,05 est significative statistiquement. À cette fin, nous effectuons une simulation avec des échantillons de tailles fictives (Figure 1).

effectifs pour les situations 1 et 2	différence de fréquence de réussite	répartition pour la situation 1	répartition pour la situation 2	p_value pour le test du chi2
100-100	0.10	[45, 55]	[55, 45]	0.203092
100-100	0.10	[10, 90]	[20, 80]	0.074706
100-500	0.10	[45, 55]	[275, 225]	0.085428
200-200	0.10	[90, 110]	[110, 90]	0.057433
250-250	0.10	[112, 138]	[138, 112]	0.025347
300-300	0.10	[135, 165]	[165, 135]	0.017892
500-500	0.10	[225, 275]	[275, 225]	0.001942
500-500	0.05	[225, 275]	[250, 250]	0.128563
1000-1000	0.05	[450, 550]	[500, 500]	0.028229
3000-3000	0.05	[1350, 1650]	[1500, 1500]	0.000117

FIGURE 1 – Simulation pour l'étude de la robustesse des données

Quelques essais nous ont amenés à prendre des fréquences de réussite moyennes (0.45 et 0.55 pour une différence de 0.1, et 0.45 et 0.5 pour une différence de 0.05), car ce sont celles qui génèrent les p-values les plus élevées au test du chi2.

Nous déduisons de nos simulations, qu'une différence de 0.1 de fréquence de réussite est significative statistiquement ( $p\text{-value} < 0.5$ , seuil couramment admis) à partir d'un effectif de 250 sujets environ pour les deux échantillons. Une différence de 0.05 de fréquence de réussite est quant à elle significative à partir d'échantillons d'effectif supérieur à 1000.

Si nous reprenons les effectifs de participants au différents tours du concours Algoréa (Tableau 1), nous déduisons de la simulation précédente qu'une différence de fréquence de réussite entre deux puzzles de 0.05 est très significative ( $p\text{-value} < 0.01$ ) lorsque l'on considère les effectifs globaux (tous niveaux scolaires confondus). Pour les niveaux scolaires du collège, les différences de fréquence de réussite entre deux puzzles sont toutes significatives ( $p\text{-value} < 0.05$ ), sauf pour le tour 2020-3. Les données pour les niveaux scolaires de l'école élémentaires sont moins robustes. Nous mettons sur fond orangé les effectifs qui ne permettent pas qu'une différence de fréquence de réussite de 0.1 soit significative.

Au terme de cette analyse, nous écartons les résultats du tour 2020-3 de notre

	2018-1	2018-2	2018-3	2019-1	2019-2	2019-3	2020-1	2020-2	2020-3	2021-1	2021-2	2021-3	2022-1	2022-2	2022-3
CMI	505	146	97	543	190	94	920	313	89	722	408	261	733	368	203
CM2	967	264	146	813	182	78	1419	109	11	1160	502	145	1108	536	132
6ème	21374	5976	2442	24779	7622	2830	6578	4101	977	22169	7606	3741	27142	13471	6446
5ème	19500	4735	1756	23523	5525	2621	6202	3405	651	19872	6567	3317	23621	11344	5252
4ème	16077	3566	1175	22429	4892	2051	6459	2716	524	18535	6219	2381	22785	10739	4371
3ème	13160	2674	881	17718	2863	1420	4401	2209	485	14530	4573	1530	17021	6478	2345
Total	71583	17361	6497	89805	21274	9094	25979	12853	2737	76988	25875	11375	92410	43705	18939

TABLEAU 1 – Nombre de participants aux différents tours du concours Algoréa selon les niveaux de classe



---

analyse des puzzles, à cause du trop petit nombre de participants par niveau scolaire (reprise de l'école après le premier confinement). Ainsi, le nombre de participants de chaque tour est supérieur à 5000, ce qui confère une bonne robustesse globale aux données. Dans cette configuration, nous gardons 139 puzzles des éditions 2018 à 2022 pour l'analyse des résultats à l'échelle nationale.

## **Annexe 6 : préparation des données à l'échelle des classes**

# Preparation\_donnees\_v3\_annexe\_these

June 24, 2024

## 1 Annexe : Préparation des données issues de la base be-bras.chticode

```
[1]: import sys
      sys.path.append('../')

      from src import *
      import pandas as pd

      %load_ext autoreload
      %autoreload 2
```

pygame 2.1.2 (SDL 2.0.16, Python 3.9.15)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

2024-06-24 17:08:02.349740: I tensorflow/core/platform/cpu\_feature\_guard.cc:193]

This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2024-06-24 17:08:02.607906: E tensorflow/stream\_executor/cuda/cuda\_blas.cc:2981]

Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

2024-06-24 17:08:03.279039: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dLError: libnvinfer.so.7: cannot open shared object file: No such file or directory

2024-06-24 17:08:03.279145: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer\_plugin.so.7'; dLError: libnvinfer\_plugin.so.7: cannot open shared object file: No such file or directory

2024-06-24 17:08:03.279150: W

tensorflow/compiler/tf2tensorrt/utils/py\_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed properly.

```
[ ]: pd.set_option('display.max_rows',200)
```

### 1.0.1 Préparation du fichier des données nettoyées à partir de l'export brut

Requête SQL dans datafroi/data/data\_these/exports\_bruts

Le fichier activity.csv contient les données brutes : - des sujets suivis en visio en catégorie blanche et jaune lors des éditions 2021 et 2022 (sauf 2021-3 jaune : log non activés) - de l'expérimentation complémentaire de l'été 2022 - des groupes classes de 2021-1, des trois tours de 2022

```
[2]: dt_brut = pd.read_csv('../data/data_these/exports_bruts/activity.csv')
```

```
[3]: len(dt_brut)
```

```
[3]: 141901
```

```
[5]: dt_brut.sort_values(by=["ID"]).head(10)
```

```
[5]:
```

	ID	password	groupID	questionID	\
63420	250365	qpqcet78	546641506633946078	603617216933141191	
63421	250366	qpqcet78	546641506633946078	603617216933141191	
63422	250367	qpqcet78	546641506633946078	603617216933141191	
63423	250368	qpqcet78	546641506633946078	603617216933141191	
63424	250369	qpqcet78	546641506633946078	603617216933141191	
63425	250370	qpqcet78	546641506633946078	603617216933141191	
63426	250371	qpqcet78	546641506633946078	603617216933141191	
63427	250372	qpqcet78	546641506633946078	603617216933141191	
63428	250373	qpqcet78	546641506633946078	603617216933141191	
63429	250374	qpqcet78	546641506633946078	603617216933141191	

		name	type	\
63420	1 - Planter des fleurs	[white 2021]	load	
63421	1 - Planter des fleurs	[white 2021]	extra	
63422	1 - Planter des fleurs	[white 2021]	attempt	
63423	1 - Planter des fleurs	[white 2021]	submission	
63424	1 - Planter des fleurs	[white 2021]	extra	
63425	1 - Planter des fleurs	[white 2021]	extra	
63426	1 - Planter des fleurs	[white 2021]	attempt	
63427	1 - Planter des fleurs	[white 2021]	attempt	
63428	1 - Planter des fleurs	[white 2021]	attempt	
63429	1 - Planter des fleurs	[white 2021]	attempt	

		answer	score	\
63420		NaN	NaN	
63421		"loadLevel;basic"	0.0	
63422	"{"basic": [{"blockly": "<xml xmlns=\\\\"		NaN	
63423	"{"basic": [{"blockly": "<xml xmlns=\\\\"		10.0	
63424		"loadLevel;hard"	0.0	

```

63425                                     "loadLevel;easy"      0.0
63426  "{\"basic\": [{\"blockly\": \"<xml xmlns=\\\\"htt...  NaN
63427  "{\"basic\": [{\"blockly\": \"<xml xmlns=\\\\"htt...  NaN
63428  "{\"basic\": [{\"blockly\": \"<xml xmlns=\\\\"htt...  NaN
63429  "{\"basic\": [{\"blockly\": \"<xml xmlns=\\\\"htt...  NaN

```

```

                                     date
63420  2020-12-13 16:07:08
63421  2020-12-13 16:07:10
63422  2020-12-13 16:08:40
63423  2020-12-13 16:08:40
63424  2020-12-13 16:08:41
63425  2020-12-13 16:08:45
63426  2020-12-13 16:09:36
63427  2020-12-13 16:11:05
63428  2020-12-13 16:11:31
63429  2020-12-13 16:12:01

```

Nettoyage des données pour ne garder que les enregistrements utiles, et mettre en forme les champs answer et score

```
[7]: dt_clean = clean('../data/data_these/exports_bruts/activity.csv', save='../data/
      ↪data_these/clean_nov23.csv', keep_all_tmp=False)
```

```
0%|          0/12
```

```
[8]: len(dt_clean)
```

```
[8]: 92779
```

Suppression des enregistrements des sessions qui ne font pas partie de l'expérimentation (tests, enseignant, sujets pas dans la tranche d'âge)

```
[9]: dt_users = pd.read_csv('../data/data_these/users.csv')
      list_users = list(dt_users['password'].unique())
```

```
[10]: dt_clean = dt_clean[dt_clean['password'].isin(list_users)]
```

```
[11]: len(dt_clean)
```

```
[11]: 91847
```

```
[12]: dt_clean.to_csv('../data/data_these/clean_nov23.csv', index=False)
```

## 1.0.2 Ajout d'un champ context dans la table des enregistrements nettoyés

```
[13]: dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv', parse_dates=[DATE])
dt_clean.fillna(EMPTY)
```

```
[13]:
```

	ID	password	grade	id_problem	problem	level	type
0	741412	33wxj8ri	603931811100778560	808561348827292307	1 - Dans l'espace [2022-3 white]	menu	load
1	741426	33wxj8ri	603931811100778560	808561348827292307	1 - Dans l'espace [2022-3 white]	basic	load
2	741481	33wxj8ri	603931811100778560	808561348827292307	1 - Dans l'espace [2022-3 white]	basic	submission
3	741496	33wxj8ri	603931811100778560	808561348827292307	1 - Dans l'espace [2022-3 white]	easy	load
4	741578	33wxj8ri	603931811100778560	808561348827292307	1 - Dans l'espace [2022-3 white]	easy	submission
...	...	...	...	...	...	...	...
91842	258429	yymtwyqt	886668386140095886	231773283829863694	4 - Planter des fleurs [white 2021]	hard	load
91843	258455	yymtwyqt	886668386140095886	231773283829863694	4 - Planter des fleurs [white 2021]	hard	attempt
91844	258466	yymtwyqt	886668386140095886	231773283829863694	4 - Planter des fleurs [white 2021]	hard	attempt
91845	258467	yymtwyqt	886668386140095886	374286687759083351	5 - Collecter les pierres précieuses [white 2021]	menu	load
91846	258470	yymtwyqt	886668386140095886	374286687759083351	5 - Collecter les pierres précieuses [white 2021]	basic	load
...	...	...	...	...	...	...	...
0						answer	score
0							0
1							0
2						<xml><block type="robot_start" x="0" y="0"><ne...	10
3							0
4						<xml><block type="robot_start" x="0" y="0"><ne...	20
...	...	...	...	...	...	...	...
91842							0
91843						<xml><block type="robot_start" x="0" y="0"><ne...	0
91844						<xml><block type="robot_start" x="0" y="0"><ne...	0
91845							0
91846							0
...	...	...	...	...	...	...	...
			date	time_session	time_attempt	time_load	
0			2022-05-12 08:48:37	0	0	3	
1			2022-05-12 08:48:40	3	3	16	

2	2022-05-12 08:48:56	19	16	4
3	2022-05-12 08:49:00	23	4	23
4	2022-05-12 08:49:23	46	23	5
...	...	...	...	...
91842	2021-01-28 10:54:48	2600	2	54
91843	2021-01-28 10:55:42	2654	54	11
91844	2021-01-28 10:55:53	2665	11	0
91845	2021-01-28 10:55:53	2665	0	5
91846	2021-01-28 10:55:58	2670	5	0

[91847 rows x 13 columns]

```
[14]: def contexte(str_probleme) :
    list_corres = [('Chauffer le château', 'castle'),
                  ('Atteindre le drapeau', 'course'),
                  ('Atteindre un drapeau', 'course'),
                  ('Planter des fleurs', 'flowers'),
                  ('Collecter les pierres', 'gems'),
                  ('Collecter les diamants', 'gems'),
                  ('Collecter les pierres précieuses', 'gems'),
                  ('Ranger les billes', 'marbles'),
                  ('Peindre le dessin', 'paint'),
                  ('Peindre le motif', 'paint'),
                  ('Pousser les caisses', 'sokoban'),
                  ("Dans l'espace", 'space'),
                  ('Rejoindre la fusée', 'space'),
                  ('Distribuer le bois', 'veterinary'),
                  ('Préparer les colis', 'packages'),
                  ('Jouer aux dominos', 'dominoes'),
                  ('Apporter du poisson', 'fishing')]
    for iCorres in list_corres :
        if iCorres[0] in str_probleme :
            return iCorres[1]
    return 'Ne correspond pas à un contexte défini'
```

```
[15]: dt_clean['context'] = dt_clean['problem'].apply(contexte)
```

### 1.0.3 Ajout d'un id avec le contexte dans la solution xml

```
[16]: def ajoute_id_contexte(str_answer, context) :
    try :
        str_answer = re.sub('block type="robot_start"', 'block_
→type="robot_start" id="' + context + "'", str_answer)
    except :
        pass
```

```
return str_answer
```

```
[17]: dt_clean['answer'] = dt_clean.apply(lambda x: ajoute_id_contexte(x['answer'],  
↳x['context']), axis=1)
```

```
[26]: dt_clean.to_csv('../data/data_these/clean_nov23.csv', index=False)
```

#### 1.0.4 Ajout du niveau scolaire

```
[19]: repertory = '../data/data_these'  
file_in = 'clean_nov23.csv'  
file_out = 'clean_nov23.csv'  
file_users = 'users.csv'
```

```
[20]: add_users_grade (repertory, file_in, file_out, file_users)
```

```
[24]: dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv', parse_dates=[DATE])  
dt_clean.fillna(EMPTY)[['ID', 'password', 'problem', 'level', 'type', 'answer',  
↳'score', 'date', 'context', 'grade']].head(10)
```

```
[24]:
```

	ID	password	problem	level	\
0	741412	33wxj8ri	1 - Dans l'espace [2022-3 white]	menu	
1	741426	33wxj8ri	1 - Dans l'espace [2022-3 white]	basic	
2	741481	33wxj8ri	1 - Dans l'espace [2022-3 white]	basic	
3	741496	33wxj8ri	1 - Dans l'espace [2022-3 white]	easy	
4	741578	33wxj8ri	1 - Dans l'espace [2022-3 white]	easy	
5	741598	33wxj8ri	1 - Dans l'espace [2022-3 white]	medium	
6	741752	33wxj8ri	1 - Dans l'espace [2022-3 white]	medium	
7	741768	33wxj8ri	2 - Atteindre le drapeau [2022-3 white]	menu	
8	741787	33wxj8ri	2 - Atteindre le drapeau [2022-3 white]	basic	
9	741890	33wxj8ri	2 - Atteindre le drapeau [2022-3 white]	basic	

	type	answer	score	\
0	load		0	
1	load		0	
2	submission	<xml><block type="robot_start" id="space" x="0...	10	
3	load		0	
4	submission	<xml><block type="robot_start" id="space" x="0...	20	
5	load		0	
6	submission	<xml><block type="robot_start" id="space" x="0...	30	
7	load		0	
8	load		0	
9	submission	<xml><block type="robot_start" id="course" x="...	10	

	date	context	grade
0	2022-05-12 08:48:37	space	5em
1	2022-05-12 08:48:40	space	5em



```

2 2022-05-12 08:48:56 space 5em
3 2022-05-12 08:49:00 space 5em
4 2022-05-12 08:49:23 space 5em
5 2022-05-12 08:49:28 space 5em
6 2022-05-12 08:50:17 space 5em
7 2022-05-12 08:50:23 course 5em
8 2022-05-12 08:50:26 course 5em
9 2022-05-12 08:51:02 course 5em

```

### 1.0.5 Création du fichiers des profils

[28]: *#Si une erreur est renvoyée, vérifier que toutes les solutions de référence #sont bien dans datafrio/data/sub/sol.ref.csv*

```

matrice_profils = build_profile_array(dt_clean,
                                     0.5,
                                     save=False,
                                     individual_times=True,
                                     multi_index_col=False,
                                     add_info=True)

dt_profils = matrice_profils.data
dt_profils.reset_index(inplace=True)
dt_profils.drop(['time_expert', 'time_adjust'], axis = 1, inplace = True)
dt_profils.rename(columns = {'groping_index': 'ind_speed'}, inplace = True)
dt_profils.head(10)

```

```

[28]: password          id_problem          problem \
0 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
1 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
2 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
3 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
4 33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]
5 33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]
6 33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]
7 33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]
8 33wxj8ri  748042633662361316      5 - Dans l'espace [2022-3 white]
9 33wxj8ri  808561348827292307      1 - Dans l'espace [2022-3 white]

level          profile  success  attempts_number  time_spent_on_question \
0  basic  accommodation  True          6              77
1  easy   expert          True          1              21
2  medium accommodation  True         17             319
3  hard   échec          False         1              56
4  basic  expert          True          1              36
5  easy   expert          True          1              26
6  medium expert          True          1              53
7  hard   échec          False         7             141

```

8	basic	accommodation	True	20	224
9	basic	expert	True	1	16

	mean_time_per_block	block_count_ref	ind_speed
0	5.1	4.0	2.774510
1	5.1	7.0	-0.411765
2	5.1	13.0	3.811463
3	5.1	11.0	-0.001783
4	5.1	6.0	0.176471
5	5.1	6.0	-0.150327
6	5.1	11.0	-0.055258
7	5.1	7.0	2.949580
8	5.1	9.0	3.880174
9	5.1	3.0	0.045752

```
[29]: len(dt_profils)
```

```
[29]: 14891
```

Ajout du niveau scolaire

```
[30]: dt_profils = dt_profils.merge(dt_clean[['password', 'problem', 'level',
      ↳ 'grade']],
      how='left',
      on=['password', 'problem', 'level']
      )
```

```
[31]: dt_profils.drop_duplicates(inplace=True)
```

```
[32]: len(dt_profils)
```

```
[32]: 14891
```

```
[33]: dt_profils.to_csv('../data/data_these/profils_nov23.csv', index=False)
```

### 1.0.6 Renseignement du taux de réussite au concours national dans le fichier des profils

```
[34]: dt_profils = pd.read_csv('../data/data_these/profils_nov23.csv',
      ↳ dtype={'id_problem': str})
      dt_profils.head(5)
```

```
[34]:   password      id_problem      problem \
0  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
1  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
2  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
3  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
```

4 33wxj8ri 613408804708837047 2 - Atteindre le drapeau [2022-3 white]

	level	profile	success	attempts_number	time_spent_on_question	\
0	basic	accommodation	True	6	77	
1	easy	expert	True	1	21	
2	medium	accommodation	True	17	319	
3	hard	échec	False	1	56	
4	basic	expert	True	1	36	

	mean_time_per_block	block_count_ref	ind_speed	grade
0	5.1	4.0	2.774510	5em
1	5.1	7.0	-0.411765	5em
2	5.1	13.0	3.811463	5em
3	5.1	11.0	-0.001783	5em
4	5.1	6.0	0.176471	5em

```
[35]: dt_concours = pd.read_csv('../data/data_these/resultats_concours.csv',  
    ↳ dtype={'questionId sur chticode': str})  
dt_concours.rename(columns={'questionId sur chticode': 'id_problem', 'version':  
    ↳ 'level'}, inplace=True)  
dt_concours.head(5)
```

```
[35]: id_situation      id_problem  level      ref_svn  annee  tour  \  
0          1  231773283829863694  easy  white-04-flowers  2021    1  
1          2   78619974426136376  easy  white-04-dominoes  2021    2  
2          3  401046766737353289  easy  white-04-space    2021    3  
3          4  988549597388007380  medium white-1-flowers  2020    1  
4          5  401046766737353289  medium white-04-space    2021    3
```

	categorie	Effectif_all	all	CM1	CM2	6ème	5ème	4ème	3ème	\
0	blanche	74932	0.982	0.936	0.982	0.960	0.984	0.995	0.997	
1	blanche	25095	0.974	0.959	0.963	0.958	0.973	0.985	0.988	
2	blanche	11248	0.983	0.969	0.979	0.975	0.983	0.991	0.989	
3	blanche	25979	0.936	0.777	0.879	0.880	0.959	0.977	0.978	
4	blanche	11248	0.908	0.835	0.897	0.873	0.917	0.939	0.939	

	classe_situation
0	1.0
1	1.0
2	1.0
3	2.0
4	2.0

```
[36]: dt_profils = dt_profils.merge(dt_concours[['id_problem', 'level', 'all']],  
    ↳ how='left',  
    ↳ on=['id_problem', 'level'])
```

```
[37]: dt_profils.rename(columns={'all':'tx_reussite_concours'}, inplace=True)
```

```
[38]: dt_profils.head(5)
```

```
[38]: password          id_problem          problem \
0  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
1  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
2  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
3  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
4  33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]

   level      profile  success  attempts_number  time_spent_on_question \
0  basic  accommodation    True                6                77
1  easy      expert      True                1                21
2  medium  accommodation    True               17               319
3  hard      échec      False                1                56
4  basic      expert      True                1                36

   mean_time_per_block  block_count_ref  ind_speed  grade  tx_reussite_concours
0                5.1                4.0  2.774510  5em            0.938
1                5.1                7.0 -0.411765  5em            0.913
2                5.1               13.0  3.811463  5em            0.799
3                5.1               11.0 -0.001783  5em            0.054
4                5.1                6.0  0.176471  5em            0.994
```

```
[39]: dt_profils.to_csv('../data/data_these/profils_nov23.csv', index=False)
```

### 1.0.7 Champ qui renseigne un champ id\_sol, si l'image de solution est présente dans la banque d'image

```
[40]: # Chemin du répertoire contenant les fichiers de réponses
reponses_path = '../data/banque_these/sol/'

# Liste des fichiers dans le répertoire de réponses
fichiers_reponses = os.listdir(reponses_path)
```

```
[41]: # Création d'un dictionnaire pour stocker le contenu de chaque fichier
contenu_fichiers = {}
for fichier in fichiers_reponses:
    with open(os.path.join(reponses_path, fichier), 'r') as file:
        contenu_fichiers[fichier] = file.read()
```

```
[42]: def renseigne_id_sol(chaine):
    for nom_fichier, contenu in contenu_fichiers.items():
        if chaine == contenu:
            id_sol = nom_fichier.replace('sol-', '')
            id_sol = id_sol.replace('.txt', '')
```

```
        return int(id_sol)
    return 0
```

```
[43]: dt_clean['id_sol'] = dt_clean['answer'].apply(renseigne_id_sol)
```

```
[44]: dt_clean.head(5)
```

```
[44]:
```

	ID	password	id_group	id_problem	\
0	741412	33wxj8ri	603931811100778560	808561348827292307	
1	741426	33wxj8ri	603931811100778560	808561348827292307	
2	741481	33wxj8ri	603931811100778560	808561348827292307	
3	741496	33wxj8ri	603931811100778560	808561348827292307	
4	741578	33wxj8ri	603931811100778560	808561348827292307	

		problem	level	type	\
0	1	- Dans l'espace [2022-3 white]	menu	load	
1	1	- Dans l'espace [2022-3 white]	basic	load	
2	1	- Dans l'espace [2022-3 white]	basic	submission	
3	1	- Dans l'espace [2022-3 white]	easy	load	
4	1	- Dans l'espace [2022-3 white]	easy	submission	

		answer	score	\
0		NaN	0	
1		NaN	0	
2	<xml><block type="robot_start" id="space" x="0...		10	
3		NaN	0	
4	<xml><block type="robot_start" id="space" x="0...		20	

	date	time_session	time_attempt	time_load	context	grade	\
0	2022-05-12 08:48:37	0	0	3	space	5em	
1	2022-05-12 08:48:40	3	3	16	space	5em	
2	2022-05-12 08:48:56	19	16	4	space	5em	
3	2022-05-12 08:49:00	23	4	23	space	5em	
4	2022-05-12 08:49:23	46	23	5	space	5em	

	id_sol
0	0
1	0
2	1
3	0
4	2

```
[45]: dt_clean.to_csv('../data/data_these/clean_nov23.csv', index=False)
```

## 1.0.8 Génération des images pour les solutions non encore présentes dans la banques d'images

```
[46]: dt_no_img = dt_clean[(dt_clean['type'].isin(['attempt', 'submission'])) &
↳(dt_clean['id_sol']==0)]
```

```
[47]: dt_no_img
```

```
[47]: Empty DataFrame
```

```
Columns: [ID, password, id_group, id_problem, problem, level, type, answer,
score, date, time_session, time_attempt, time_load, context, grade, id_sol]
Index: []
```

```
[ ]: def sol_xml_to_files_context(data, champ_xml):
    activities_grouped = data.groupby('context')
    activities_formatted = pd.DataFrame()
    repertoire_in = '../data/provisoire'

    for context, context_activities in activities_grouped:
        try:
            os.mkdir(repertoire_in + '/' + str(context) + '_sol')
            os.mkdir(repertoire_in + '/' + str(context) + '_img')
        except FileExistsError:
            pass

        context_activities['id_answer'], answers =
↳context_activities[champ_xml].factorize()
        context_activities['id_answer'] += 1

        for idAnswer, answer in enumerate(answers):
            if answer == "0":
                continue
            with open(os.path.join(repertoire_in + '/' + str(context) + '_sol',
↳f"{context}_{idAnswer+1}.txt"), "w") as answer_file:
                answer_file.write(str(answer))

        activities_formatted = pd.concat([activities_formatted,
↳context_activities])
        #la ligne ci-dessous doit être commentée pour générer les images des
↳id_prot manquants
        #activities_formatted = activities_formatted.sort_values(by=[PASSWORD,
↳DATE])
        activities_formatted.to_csv(repertoire_in + '/sol_out.csv', index=False)
```

```
[ ]: sol_xml_to_files_context(dt_no_img, 'answer')
```

```
[ ]: dt_no_img = pd.read_csv('../data/provisoire/sol_out.csv', parse_dates=[DATE])
dt_no_img.fillna(EMPTY).head(5)
```

### 1.0.9 Intégration au dataframe d'origine des id\_sol générés lors de la création des images

```
[ ]: dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv', parse_dates=[DATE])
```

```
[ ]: def ajouter_id_generes(data, champ_id, champ_merge) :
    dt_ajout = pd.read_csv('../data/provisoire/reassort_id.csv')
    data = data.merge(dt_ajout[[champ_merge, champ_id]], on=champ_merge,
↳how='left', suffixes=('_df', '_dfajout'))
    data[champ_id] = data[champ_id + '_dfajout'].combine_first(data[champ_id +
↳'_df'])
    data = data.astype({champ_id : 'int32'})
    data = data.drop([champ_id + '_dfajout', champ_id + '_df'], axis=1)
    return data
```

```
[ ]: ajouter_id_generes(dt_clean, 'id_sol', 'ID')
```

```
[ ]: #Vérifier que ce dataframe est maintenant vide
dt_clean[(dt_clean['type'].isin(['attempt', 'submission'])) &
↳(dt_clean['id_sol']==0)]
```

```
[ ]: dt_clean.to_csv('../data/data_these/clean_nov23.csv', index=False)
```

### 1.0.10 Création de la table des programmes soumis

```
[ ]: dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv', parse_dates=[DATE])
dt_clean.fillna(EMPTY).head(5)
```

```
[ ]: dt_sol = dt_clean[dt_clean['id_sol']>0].groupby(['id_problem', 'problem',
↳'context', 'level', 'answer', 'id_sol']).agg({'ID': 'count'})
dt_sol = dt_sol.reset_index()
dt_sol.rename(columns={'ID': 'nb_occurrences'}, inplace=True)
dt_sol.sort_values('nb_occurrences', ascending=False).head(10)
```

```
[ ]: len(dt_sol)
```

### 1.0.11 Création de la table des solutions prototypiques

```
[ ]: dt_sol['sol_prototypique'] = dt_sol['answer'].apply(sol_canonique)
```

```
[ ]: dt_prot = dt_sol.groupby(['id_problem', 'problem', 'context', 'level',
↳'sol_prototypique']).agg({'nb_occurrences': 'sum', 'id_sol': 'unique'}).
↳reset_index()
```

```
dt_prot.sort_values('nb_occurrences', ascending=False).head(10)
```

```
[ ]: len(dt_prot)
```

### 1.0.12 Ajout d'un id\_prot à la table des solutions prototypiques

```
[ ]: dt_base_sol = pd.read_csv('../data/banque_these/db_solutions.csv')  
dt_base_sol.head(3)
```

```
[ ]: def ajoute_id(dt_donnees, dt_base_sol, champ_xml, champ_id) :  
    f = lambda x :  
    ↪dt_base_sol['id'][(dt_base_sol[dt_base_sol['xml']==(dt_prot[champ_xml][x])].  
    ↪index[0])]  
    for enreg in range(len(dt_prot)) :  
        try :  
            dt_prot.loc[enreg, champ_id] = f(enreg)  
        except :  
            dt_prot.loc[enreg, champ_id] = 0
```

```
[ ]: ajoute_id(dt_prot, dt_base_sol, 'sol_prototypique', 'id_prot')
```

```
[ ]: dt_prot = dt_prot.astype({'id_prot': int})  
dt_prot.head(5)
```

```
[ ]: len(dt_prot[dt_prot['id_prot']==0])
```

### 1.0.13 Génération d'une image de solution et d'un id\_prot lorsque celui-ci n'est pas présent dans la banque de solutions

```
[ ]: dt_no_id_prot = dt_prot[dt_prot['id_prot']==0].sort_values(by='nb_occurrences',  
    ↪ascending=False)  
dt_no_id_prot.head(10)
```

```
[ ]: sol_xml_to_files_context(dt_no_id_prot, 'sol_prototypique')
```

- générer les images dans la console du navigateur
- générer les id\_prot avec le notebook banque\_solutions

```
[ ]: dt_prot = ajouter_id_generees(dt_prot, 'id_prot', 'sol_prototypique')
```

```
[ ]: dt_prot[dt_prot['id_prot']==0]
```

```
[ ]: dt_prot.to_csv("../data/data_these/table_sol_prot.csv", index=False)
```



### 1.0.14 Complétion de la table des solutions prototypiques avec l'ensemble des indicateurs

```
[48]: dt_prot = pd.read_csv('../data/data_these/table_sol_prot.csv')
dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv')
dt_profils = pd.read_csv('../data/data_these/profils_nov23.csv')
dt_users = pd.read_csv('../data/data_these/users.csv')
```

```
[49]: dt_prot.head(5)
```

```
[49]:
```

	id_problem	problem	context	level	\
0	1555753216143902	3 - Pousser les caisses [2021 yellow]	sokoban	basic	
1	1555753216143902	3 - Pousser les caisses [2021 yellow]	sokoban	easy	
2	1555753216143902	3 - Pousser les caisses [2021 yellow]	sokoban	easy	
3	1555753216143902	3 - Pousser les caisses [2021 yellow]	sokoban	easy	
4	1555753216143902	3 - Pousser les caisses [2021 yellow]	sokoban	easy	

	sol_prototypique	nb_occurrences	id_sol	\
0	<xml><block type="robot_start" id="sokoban" x=...	2	[8160]	
1	<xml><block type="robot_start" id="sokoban" x=...	1	[7554]	
2	<xml><block type="robot_start" id="sokoban" x=...	1	[7553]	
3	<xml><block type="robot_start" id="sokoban" x=...	1	[7552]	
4	<xml><block type="robot_start" id="sokoban" x=...	1	[18101]	

	id_prot	valide	freq	freq_users	mean_accomm	std_accomm	mean_essais	\
0	8160	True	1.00	1.000	-0.12	0.07	1.0	
1	7554	False	0.05	0.167	1.31	NaN	5.0	
2	7553	False	0.05	0.167	1.31	NaN	5.0	
3	7552	False	0.05	0.167	1.31	NaN	5.0	
4	18101	False	0.05	0.167	1.79	NaN	6.0	

	std_essais	id_sujets_visio	type_sol
0	0.0	{'aigvrpue': 'expert', 'pekx5imk': 'expert'}	optimal
1	NaN	{'9yiiqsjg': 'accommodation'}	erreur
2	NaN	{'9yiiqsjg': 'accommodation'}	erreur
3	NaN	{'9yiiqsjg': 'accommodation'}	erreur
4	NaN	{'pekx5imk': 'accommodation'}	erreur

```
[ ]: # renvoie dt_prot avec les champs ajoutés
dt_prot = renseigner_infos_table_sol_prot (dt_prot, dt_clean, dt_profils,
↳ dt_users)
```

```
[ ]: dt_prot.sort_values('nb_occurrences', ascending=False).head(5)
```

```
[ ]: len(dt_prot)
```

```
[ ]: dt_prot.to_csv('../data/data_these/table_sol_prot.csv', index=False)
```

### 1.0.15 Ajout d'un champ type\_sol qui sert à déterminer la couleur de fond des images de programme

```
[ ]: file_sol_prot = '../data/data_these/table_sol_prot.csv'  
file_ref = '../data/data_these/sol_ref.csv'
```

```
[ ]: dt_prot = pd.read_csv(file_sol_prot)  
list_problems = dt_prot['problem'].unique()  
list_levels = ['basic', 'easy', 'medium', 'hard']
```

```
[ ]: for problem in list_problems :  
    for level in list_levels :  
        try :  
            ajouter_type_sol(file_sol_prot, file_ref, problem, level)  
        except :  
            print("erreur pour le problème " + problem + " version " + level)
```

```
[ ]: dt_prot = pd.read_csv(file_sol_prot)  
dt_prot.sort_values('nb_occurrences', ascending=False).head(5)
```

```
[50]: dt_prot['type_sol'].value_counts()
```

```
[50]: erreur                22472  
incomplet opti             780  
non optimal                734  
incomplet non opti        703  
optimal                    507  
Name: type_sol, dtype: int64
```

### 1.0.16 Ajout d'un champ id\_prot à dt\_clean

```
[51]: dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv')  
dt_prot = pd.read_csv('../data/data_these/table_sol_prot.csv')
```

```
[ ]: dt_clean.head(3)
```

```
[ ]: dt_prot.head(3)
```

```
[ ]: len(dt_prot)
```

```
[53]: #À lancer une seule fois, après le chargement de dt_prot  
  
def string_to_list(chaine_sol) :  
    list_sol = chaine_sol[1:-1].split()  
    return list(map(int, list_sol))  
  
dt_prot['id_sol'] = dt_prot['id_sol'].apply(string_to_list)
```

```
[54]: dt_clean['id_prot'] = 0
```

La fonction explode en pandas est utilisée pour transformer les éléments de chaque liste d'une colonne en des lignes séparées, tout en dupliquant les autres données de la ligne.

```
[55]: dt_clean = dt_clean.merge(dt_prot[['id_sol', 'id_prot']].explode('id_sol'),
    → on='id_sol', how='left', suffixes=('_clean', '_prot'))
dt_clean['id_prot'] = dt_clean['id_prot_prot'].
    → combine_first(dt_clean['id_prot_clean'])
dt_clean = dt_clean.astype({'id_prot' : 'int32'})
dt_clean = dt_clean.drop(['id_prot_prot', 'id_prot_clean'], axis=1)
```

```
[59]: dt_clean[dt_clean['id_sol'] != dt_clean['id_prot']][['ID', 'password',
    → 'problem', 'level', 'type', 'answer', 'score', 'date', 'grade', 'id_sol',
    → 'id_prot']].head(10)
```

```
[59]:
```

	ID	password	problem	level	\
25	742443	33wxj8ri	3 - Pousser les caisses [2022-3 white]	basic	
347	406825	33yjhik9	3 - Planter des fleurs [2022-1 white]	easy	
348	406855	33yjhik9	3 - Planter des fleurs [2022-1 white]	easy	
349	406876	33yjhik9	3 - Planter des fleurs [2022-1 white]	easy	
414	408177	33yjhik9	5 - Collecter les pierres [2022-1 white]	basic	
415	408198	33yjhik9	5 - Collecter les pierres [2022-1 white]	basic	
416	408219	33yjhik9	5 - Collecter les pierres [2022-1 white]	basic	
437	408434	33yjhik9	5 - Collecter les pierres [2022-1 white]	medium	
438	408440	33yjhik9	5 - Collecter les pierres [2022-1 white]	medium	
439	408448	33yjhik9	5 - Collecter les pierres [2022-1 white]	medium	

	type	answer	score	\
25	attempt	<xml><block type="robot_start" id="sokoban" x=...	0	
347	attempt	<xml><block type="robot_start" id="flowers" x=...	0	
348	attempt	<xml><block type="robot_start" id="flowers" x=...	0	
349	attempt	<xml><block type="robot_start" id="flowers" x=...	0	
414	attempt	<xml><block type="robot_start" id="gems" x="0"...	0	
415	attempt	<xml><block type="robot_start" id="gems" x="0"...	0	
416	submission	<xml><block type="robot_start" id="gems" x="0"...	10	
437	attempt	<xml><block type="robot_start" id="gems" x="0"...	0	
438	attempt	<xml><block type="robot_start" id="gems" x="0"...	0	
439	attempt	<xml><block type="robot_start" id="gems" x="0"...	0	

	date	grade	id_sol	id_prot
25	2022-05-12 08:53:38	5em	11	3677
347	2022-01-26 09:28:40	6em	116	871
348	2022-01-26 09:28:53	6em	117	3725
349	2022-01-26 09:29:07	6em	116	871
414	2022-01-26 09:42:33	6em	147	146
415	2022-01-26 09:42:54	6em	148	1492

416	2022-01-26 09:43:06	6em	149	964
437	2022-01-26 09:45:30	6em	159	7005
438	2022-01-26 09:45:41	6em	160	4214
439	2022-01-26 09:45:55	6em	161	25137

```
[58]: dt_clean.to_csv('../data/data_these/clean_nov23.csv', index=False)
```

## **Annexe 7 : traitements relatifs à la détermination des profils**

# traitements\_determination\_profls\_annexe\_these

June 29, 2024

## 1 Traitements relatifs à la détermination des profils de confrontation sujet / puzzle

Traiments effectués en amont de l'article publié lors de la conférence EIAH 2023

```
[1]: import sys
      sys.path.append('../')

      from src import *
      import pandas as pd

      %load_ext autoreload
      %autoreload 2
```

pygame 2.1.2 (SDL 2.0.16, Python 3.9.15)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

2024-06-29 08:48:00.594251: I tensorflow/core/platform/cpu\_feature\_guard.cc:193]

This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2024-06-29 08:48:00.739302: E tensorflow/stream\_executor/cuda/cuda\_blas.cc:2981]

Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

2024-06-29 08:48:01.302884: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dLError: libnvinfer.so.7: cannot open shared object file: No such file or directory

2024-06-29 08:48:01.303014: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer\_plugin.so.7'; dLError: libnvinfer\_plugin.so.7: cannot open shared object file: No such file or directory

2024-06-29 08:48:01.303020: W

tensorflow/compiler/tf2tensorrt/utils/py\_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed

properly.

### 1.0.1 Sessions de sujets qui sont au collège et ont participé en 2022

```
[7]: dt_users = pd.read_csv('../data/sub/users.csv')
dt_users = dt_users[(dt_users['school_year'].isin(['6em', '5em', '4em', '3em'])) &
                    (dt_users['session'].isin(['2022-1', '2022-2', '2022-3']))]
dt_users.head(5)
```

```
[7]:      id_common  password  school_year  session  category
35           6  yxge6gn8           6em  2022-1      blanc
36           6  i9k4am5j           6em  2022-1      jaune
37           6  i6nmn46d           6em  2022-2      blanc
38           6  eckqnn4d           6em  2022-3      jaune
39           6  7acipdym           6em  2022-3      blanc
```

```
[12]: dt_users.to_csv('../data/algorea_2022/users_2022_college.csv', index=False)
```

```
[8]: list_users = dt_users.password.to_list()
len(list_users)
```

```
[8]: 670
```

### 1.0.2 Chargement du fichier de l'édition 2022 nettoyé

Sessions supprimées : - sessions de tests (ML et enseignants) - sessions vides (connexion seulement)  
- sessions avec très peu d'enregistrements (moins d'une dizaine, qui correspondent à des sessions qui ont été quittées puis recommencées)

```
[16]: df_2022 = pd.read_csv('../data/algorea_2022/clean_2022_sessions_removed.csv',
                          parse_dates=[DATE])
# la liste des colonnes passées à 'parse_dates' sont converties en dates au lieu de strings
add_users_grade('../data/algorea_2022',
                'clean_2022_sessions_removed.csv',
                'clean_2022_sessions_removed.csv',
                'users.csv')
df_2022.fillna(EMPTY).head(5)
```

```
[16]:      ID  password  id_group  id_problem \
0  405601  33yjhik9  18142768599727689  916979020758482351
1  405606  33yjhik9  18142768599727689  916979020758482351
2  405613  33yjhik9  18142768599727689  916979020758482351
3  405637  33yjhik9  18142768599727689  916979020758482351
4  405642  33yjhik9  18142768599727689  916979020758482351
```

	problem	level	type	\
0	1 - Chauffer le château [2022-1 white]	menu	load	
1	1 - Chauffer le château [2022-1 white]	basic	load	
2	1 - Chauffer le château [2022-1 white]	basic	attempt	
3	1 - Chauffer le château [2022-1 white]	basic	submission	
4	1 - Chauffer le château [2022-1 white]	easy	load	

	answer	score	\
0		0	
1		0	
2	<xml><block type="robot_start" deletable="fals...	0	
3	<xml><block type="robot_start" deletable="fals...	10	
4		0	

	date	time_session	time_attempt	time_load	grade
0	2022-01-26 09:17:35	0	0	2	6em
1	2022-01-26 09:17:37	2	2	10	6em
2	2022-01-26 09:17:47	12	10	21	6em
3	2022-01-26 09:18:08	33	21	5	6em
4	2022-01-26 09:18:13	38	5	10	6em

```
[17]: len(df_2022)
```

```
[17]: 76846
```

```
[18]: df_2022["password"].nunique()
```

```
[18]: 692
```

Après appariement des sessions réalisées par les mêmes sujets, nombre de collégiens qui ont participé à l'édition 2022 :

```
[19]: len(dt_users.id_common.value_counts())
```

```
[19]: 250
```

```
[20]: df_2022_college = df_2022[df_2022['password'].isin(list_users)]
```

Différence avec df\_2022 : les sessions des sujets qui sont à l'école élémentaire (suivis en session vidéo) sont écartées

Ajout d'une colonne avec un id\_student qui rend compte de l'appariement des sessions

```
[22]: df_2022_college = group_users_from_single_clean_file('../data/algorea_2022/
→clean_2022_college.csv',
                                                    path_users_csv='../data/
→algorea_2022/users_2022_college.csv')
```



```
[23]: df_2022_college.head(5)
```

```
[23]:   id_student      ID password      id_group      id_problem \
0         178  405601  33yjhik9  18142768599727689  916979020758482351
1         178  405606  33yjhik9  18142768599727689  916979020758482351
2         178  405613  33yjhik9  18142768599727689  916979020758482351
3         178  405637  33yjhik9  18142768599727689  916979020758482351
4         178  405642  33yjhik9  18142768599727689  916979020758482351

      problem  level      type \
0  1 - Chauffer le château [2022-1 white]  menu      load
1  1 - Chauffer le château [2022-1 white]  basic      load
2  1 - Chauffer le château [2022-1 white]  basic  attempt
3  1 - Chauffer le château [2022-1 white]  basic  submission
4  1 - Chauffer le château [2022-1 white]  easy      load

      answer  score \
0          NaN      0
1          NaN      0
2  <xml><block type="robot_start" deletable="fals...      0
3  <xml><block type="robot_start" deletable="fals...     10
4          NaN      0

      date  time_session  time_attempt  time_load  grade
0  2022-01-26  09:17:35          0          0          2  6em
1  2022-01-26  09:17:37          2          2         10  6em
2  2022-01-26  09:17:47         12         10         21  6em
3  2022-01-26  09:18:08         33         21          5  6em
4  2022-01-26  09:18:13         38          5         10  6em
```

```
[24]: df_2022_college.to_csv('../data/algorea_2022/clean_2022_college.csv',
    ↪index=False)
```

### 1.0.3 Indicateur de rapidité normalisée et profils de confrontation sujet/puzzle

```
[ ]: df_2022_college = pd.read_csv('../data/algorea_2022/clean_2022_college.csv')
```

```
[ ]: #Fonction de la lib datafioi
def build_profile_array(data, seuil, save=True, individual_times=False,
    ↪multi_index_col = True, add_info=False, ref_file=PATH_SOL_REF):
    """Construit le tableau des profils. Fonction très variable en fonction des
    ↪paramètres.

    Args:
        data (pandas.DataFrame): enregistrements dont on veut construire la
    ↪table des profils
```

```

        seuil(float): seuil pour l'indicateur de rapidité normalisé à
        ↳ considérer dans le calcul des profils
        save (bool | str, optional): booléen si l'on souhaite sauvegarder le
        ↳ fichier, on peut également fournir une str précisant le path du fichier à
        ↳ sauvegarder. Defaults to True.
        individual_times (bool, optional): si l'on souhaite les temps
        ↳ personnalisés. Defaults to False.
        multi_index_col (bool, optional): si l'on souhaite que les niveaux de
        ↳ problèmes soient en colonnes. Defaults to True.
        add_info (bool, optional): ajoute les informations de détermination des
        ↳ profils. Ne fonctionne que si 'multi_index_col' est False. Defaults to False.
        ref_file (str, optional): path vers le csv contenant les solutions de
        ↳ référence. Defaults to PATH_SOL_REF.
    """

```

```

[25]: matrice_profils = build_profile_array(df_2022_college, 0.5, save=False,
        ↳ individual_times=True, multi_index_col=False, add_info=True)
dt_profils = matrice_profils.data
dt_profils.reset_index(inplace=True)
dt_profils.drop(['time_expert', 'time_adjust'], axis = 1, inplace = True)
dt_profils.rename(columns = {'groping_index': 'ind_speed'}, inplace = True)
dt_profils.head(5)

```

```

[25]:
password          id_problem          problem \
0  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
1  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
2  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
3  33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
4  33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]

level      profile  success  attempts_number  time_spent_on_question \
0  basic  accommodation  True           6              77
1  easy   expert         True           1              21
2  medium accommodation  True          15             287
3  hard   passé          False          0              56
4  basic   expert         True           1              36

mean_time_per_block  block_count_ref  ind_speed
0                    5.2              4  2.732687
1                    5.2              7 -0.418283
2                    5.2             13  3.280844
3                    5.2             11 -0.012843
4                    5.2              6  0.163435

```

```

[26]: len(dt_profils)

```

[26]: 11826

```
[27]: dt_profils['profile'].value_counts()
```

```
[27]: accommodation    3703
expert                2733
échec                2350
ajustement           1822
passé                1035
indéterminé         183
Name: profile, dtype: int64
```

```
[29]: dt_profils.to_csv('../data/algorea_2022/profils_2022_college.csv', index=False)
```

#### 1.0.4 Temps moyen pour placer un bloc : ajout de précision

```
[ ]: #Fonction de la lib datafioi
def add_precise_mean_time_per_block(file_in, file_to_add, file_out) :
    '''Procédure qui ajoute un champ avec le temps moyen par bloc pour chaque
    ↪ session,
        calculé en enlevant les valeurs aberrantes des premières soumissions
    Args :
        file_in : fichier de type clean, qui contient les solutions soumises au
    ↪ format xml
        file_to_add : fichier dans lequel on veut ajouter ce temps moyen pour
    ↪ placer un bloc calculé de manière précise,
        par exemple le fichier des profils (des mêmes sessions)
        file_out : fichier dans lequel exporter le résultat, peut être le même
    ↪ que file_to_add
    '''
```

```
[30]: add_precise_mean_time_per_block('../data/algorea_2022/clean_2022_college.csv',
                                       '../data/algorea_2022/profils_2022_college.csv',
                                       '../data/algorea_2022/
    ↪ profils_2022_college_precise_time_block.csv')
```

```
[31]: dt_profils_precise_time = pd.read_csv('../data/algorea_2022/
    ↪ profils_2022_college_precise_time_block.csv')
```

```
[32]: dt_mean_time = dt_profils_precise_time[['password', 'mean_time_per_block',
    ↪ 'precise_mean_time_per_block']].groupby(['password']).head(1)
dt_grade = dt_users[['password', 'school_year']].groupby(['password']).head(1)
dt_grade.rename(columns = {'school_year': 'grade'}, inplace = True)
dt_mean_time = dt_mean_time.merge(dt_grade, on=PASSWORD, how='left')
dt_mean_time['difference'] = dt_mean_time['mean_time_per_block'] -
    ↪ dt_mean_time['precise_mean_time_per_block']
```

```
[33]: dt_mean_time.describe()
```

```
[33]:      mean_time_per_block  precise_mean_time_per_block  difference
count          658.000000          658.000000          658.000000
mean            6.846049            6.642097            0.203951
std             2.023272            1.943668            0.611769
min             3.200000            3.200000           -0.100000
25%            5.600000            5.400000            0.000000
50%            6.500000            6.300000            0.000000
75%            7.600000            7.400000            0.200000
max            18.200000           18.200000            8.500000
```

Confrontation pour lesquelles le temps moyen pour placer un bloc diffère de plus d'une seconde entre les deux modes de calcul

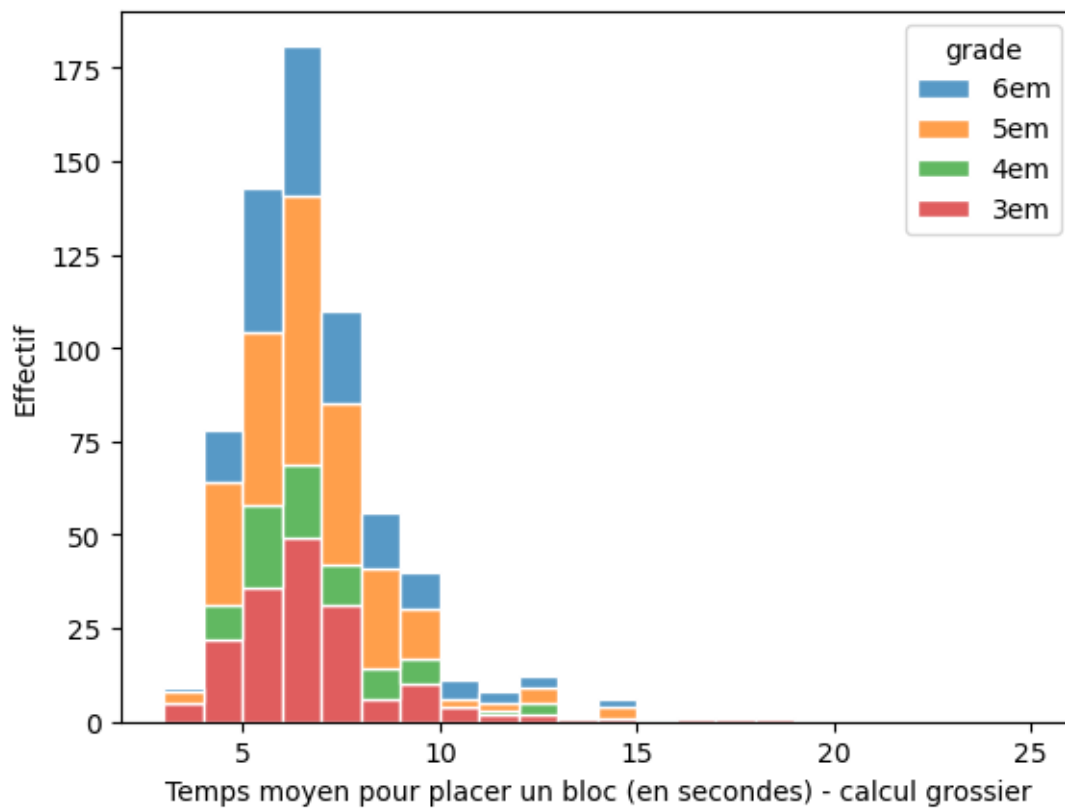
```
[35]: dt_mean_time[dt_mean_time['difference']>1]
```

```
[35]:      password  mean_time_per_block  precise_mean_time_per_block  grade  \
75   635acd5x           8.8                7.5      5em
82   69dth9gh           8.0                6.9      3em
109  7eerr8h4          12.4                8.8      6em
113  7ijwabr5           9.2                7.2      5em
125  7vtt5nud          10.5                8.8      5em
131  88e76ywa           7.7                6.3      6em
158  9hpqe9fr          12.3                8.9      5em
161  9jm3ht4c           6.2                5.0      5em
184  aq58p4us           7.7                6.5      3em
205  bghkjc3           11.8                9.1      6em
217  c4ibvgx7          12.2                5.0      4em
220  c7ineuqa           6.7                5.6      6em
248  dhh3f3k3           7.4                6.2      3em
251  djfhkjsx          14.7                6.2      3em
283  fgcyqbn           6.7                5.5      3em
320  gwje7uht           7.1                6.0      6em
344  i6sjjfyb           8.1                6.5      5em
373  ji86r7tc           9.0                7.0      5em
419  mdw6h9gj           9.1                6.5      4em
427  mrn3bsfq           9.2                7.0      5em
428  mrsj4dic           9.6                8.4      3em
457  nux786me           8.7                6.7      5em
469  pgfbdjdv           9.7                8.5      6em
480  q7ipbhpw           9.3                8.1      5em
508  reyv5cgr           9.1                6.8      3em
522  rwevkpxw           9.5                8.1      5em
541  sy7h7it8           7.2                5.8      5em
563  u98xyfhh          12.0                10.1     5em
590  vi5ptry9           7.3                5.0      4em
596  vr8tqsra          14.6                10.7     5em
```

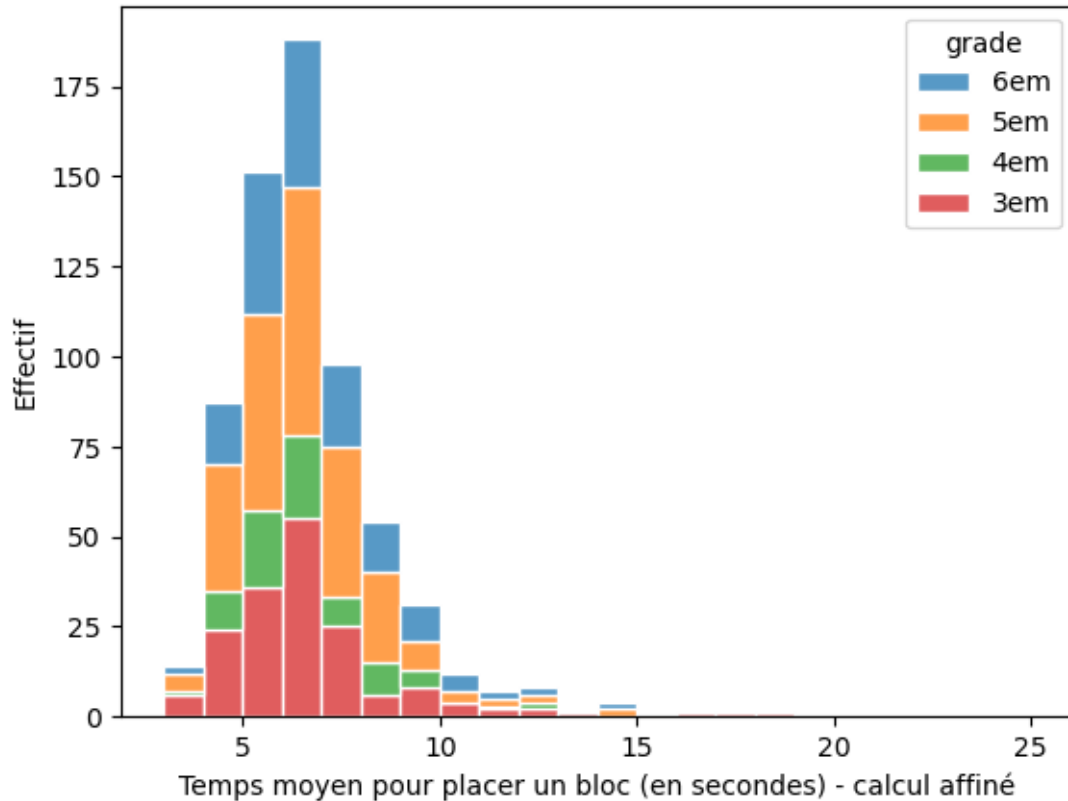
	difference
75	1.3
82	1.1
109	3.6
113	2.0
125	1.7
131	1.4
158	3.4
161	1.2
184	1.2
205	2.7
217	7.2
220	1.1
248	1.2
251	8.5
283	1.2
320	1.1
344	1.6
373	2.0
419	2.6
427	2.2
428	1.2
457	2.0
469	1.2
480	1.2
508	2.3
522	1.4
541	1.4
563	1.9
590	2.3
596	3.9
648	2.0

```
[36]: sns.histplot(data=dt_mean_time,
                  x='mean_time_per_block',
                  hue='grade',
                  binwidth=1,
                  binrange=(3,25),
                  multiple='stack',
                  edgecolor='white',
                  hue_order=['6em', '5em', '4em', '3em']
                  )
plt.gca().set_xlabel("Temps moyen pour placer un bloc (en secondes) - calcul_
↳grossier")
plt.gca().set_ylabel("Effectif")
```

```
plt.savefig('hist_temps_moyen_par_bloc_grossier.pdf')
plt.show()
```



```
[37]: sns.histplot(data=dt_mean_time,
                    x='precise_mean_time_per_block',
                    hue='grade',
                    binwidth=1,
                    binrange=(3,25),
                    multiple='stack',
                    edgecolor='white',
                    hue_order=['6em', '5em', '4em', '3em']
                )
plt.gca().set_xlabel("Temps moyen pour placer un bloc (en secondes) - calcul_
↳affiné")
plt.gca().set_ylabel("Effectif")
plt.savefig('hist_temps_moyen_par_bloc_affine.pdf')
plt.show()
```



Recalcul des profils après prise en compte du correctif pour le calcul du temps passé pour placer un bloc

```
[38]: matrice_profils = build_profile_array(df_2022_college, 0.5, save=False,
      ↪ individual_times=True, multi_index_col=False, add_info=True)
dt_profils_precise = matrice_profils.data
dt_profils_precise.reset_index(inplace=True)
dt_profils_precise.drop(['time_expert', 'time_adjust'], axis = 1, inplace =
      ↪ True)
dt_profils_precise.rename(columns = {'groping_index': 'ind_speed'}, inplace =
      ↪ True)
```

```
[39]: len(dt_profils_precise)
```

```
[39]: 11826
```

```
[40]: dt_profils_precise['profile'].value_counts()
```

```
[40]: accommodation    3762
      expert            2703
      échec             2350
```

```
ajustement      1763
passé          1035
indéterminé    213
Name: profile, dtype: int64
```

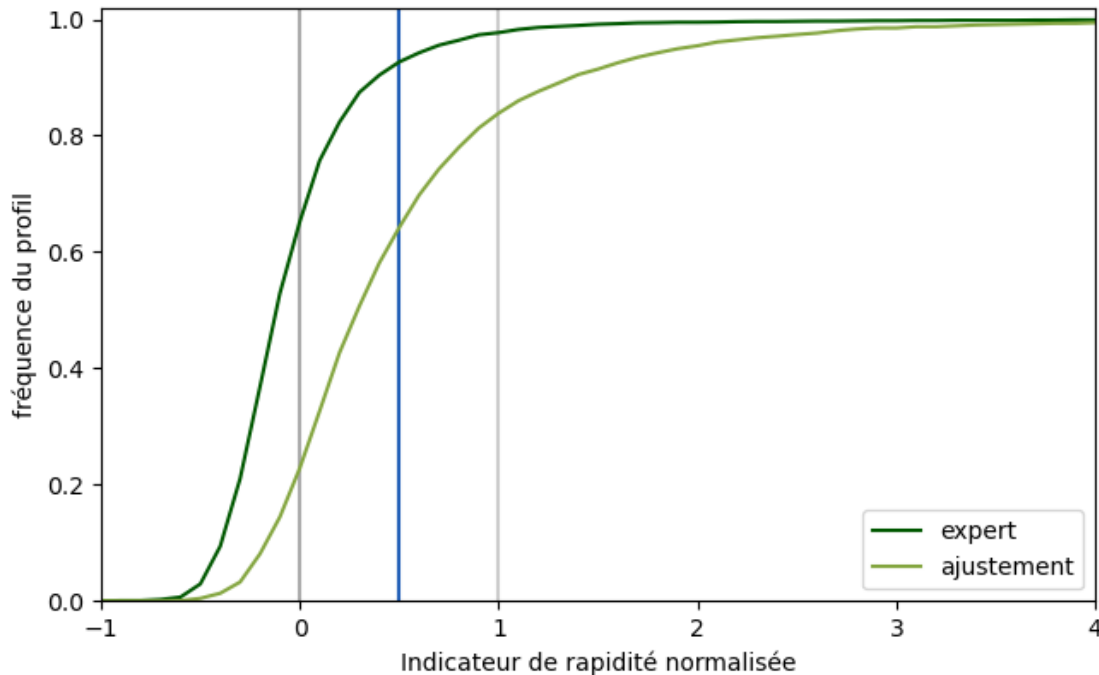
```
[42]: dt_profils_precise.to_csv('../data/algorea_2022/
      ↪ profils_2022_college_precise_time.csv', index=False)
```

### 1.0.5 Recherche du seuil pour l'indicateur de rapidité normalisée

Ce seuil distingue les profils ajustement et accommodation d'une part, expert et indéterminé d'autre part.

Le champ profile n'est pas utilisé, pour la génération du diagramme, les profils sont recalculés à partir du champ ind\_speed.

```
[41]: courbes_pourcent_profils_fonction_du_seuil(dt_profils_precise)
```



Histogrammes des profils en fonction du seuil de rapidité normalisé choisi

```
[ ]: def hist_rapidite(file_profiles, repertory_out=None, problem=None, level=None,
      ↪ seuil=0.5, debut=-1, taille_effectif=None, pas_effectif=None):
      """
      Génère l'histogramme des nombres d'essais avec l'indicateur de rapidité
      ↪ normalisée en abcisse et l'effectif en ordonnée
```



Args:

*file\_profiles* : fichier où a été calculé l'indicateur de rapidité

↳ normalisée pour chaque confrontation sujet/puzzle

ce fichier comprend notamment les champs password, problem, level,

↳ success, attempts\_number et ind\_speed

*repertory\_out* : répertoire dans lequel enregistrer la solution générée

*problem* : intitulé du problème considéré, qui doit être le même que

↳ dans le champ probleme du fichier *file\_profiles*

*level* : version du problème considéré

*seuil* : valeur de l'indicateur de rapidité normalisée qui distingue les

↳ profils expert et indéfini, ajustement et accommodation

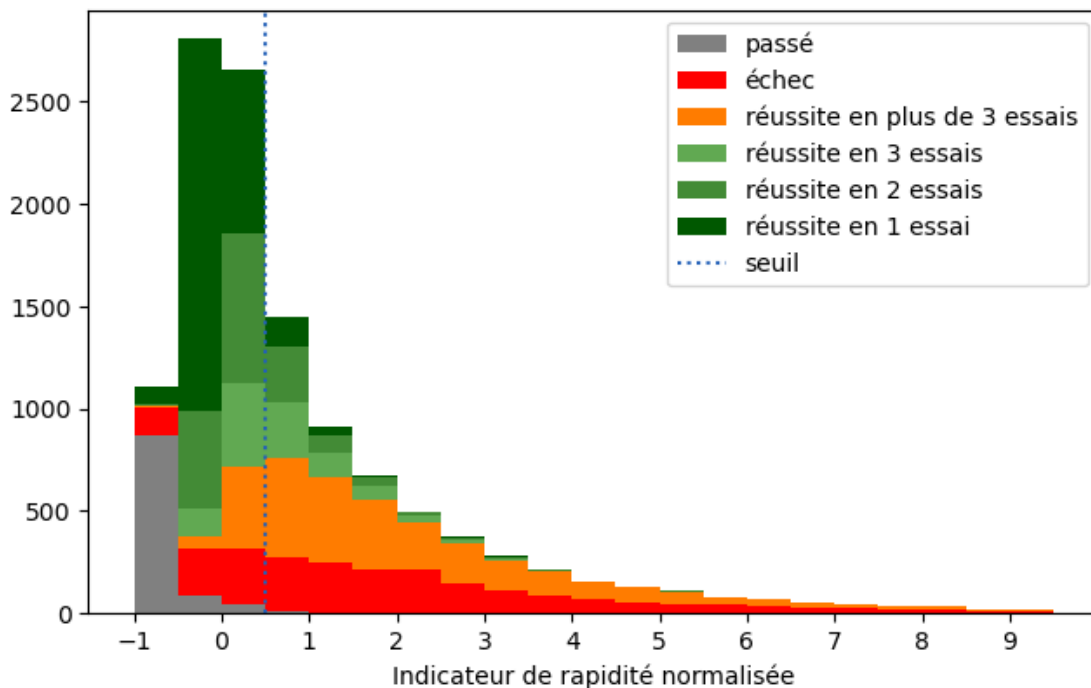
*debut* : valeur à partir de laquelle on commence les tranches, pour que

↳ 0 distingue deux tranches

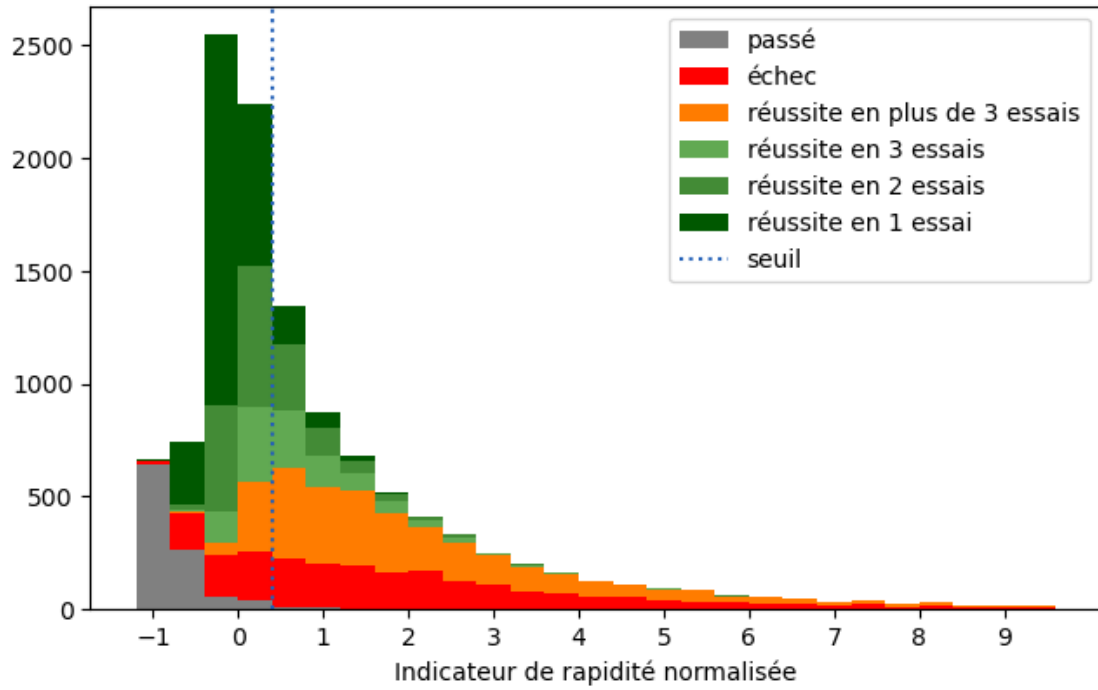
"""

```
[43]: file_profiles = '../data/algorea_2022/profils_2022_college_precise_time.csv'
```

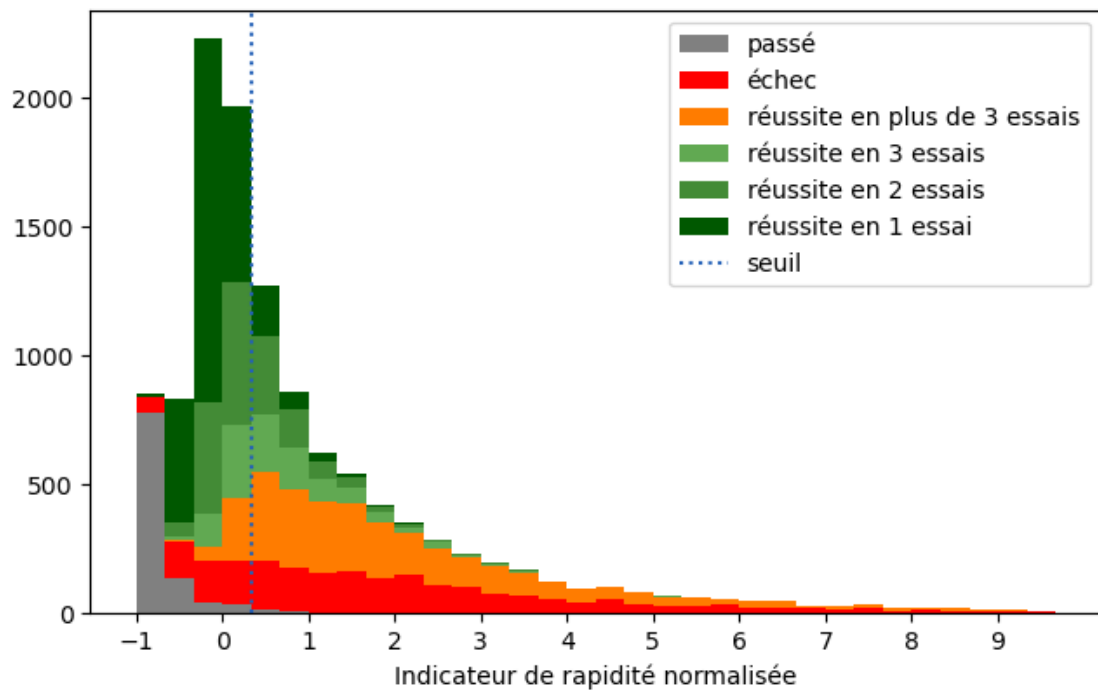
```
[44]: hist_rapidite (file_profiles, seuil = 0.5, debut= -1)
```



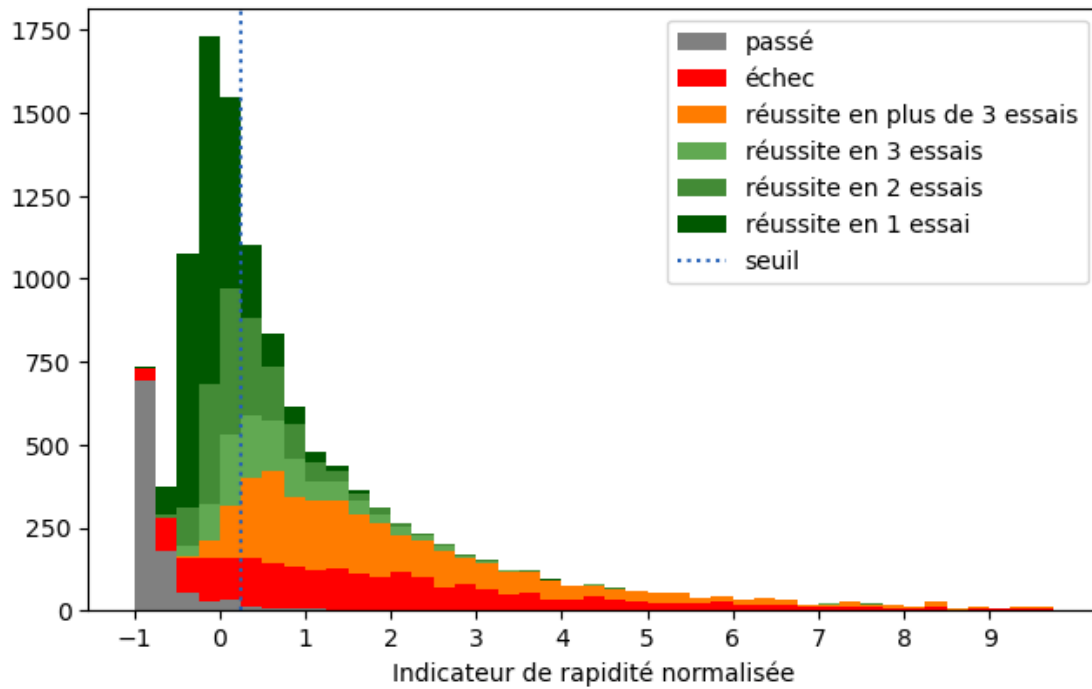
```
[45]: hist_rapidite (file_profiles, seuil = 0.4, debut= -1.2)
```



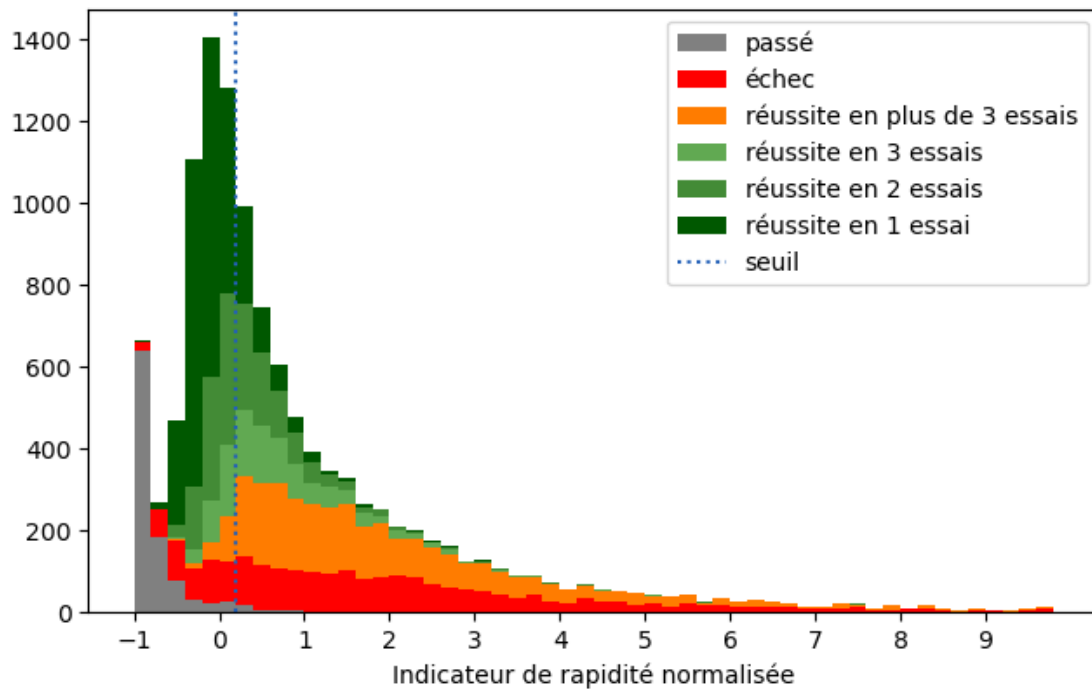
```
[46]: hist_rapidite (file_profiles, seuil = 1/3, debut= -1)
```



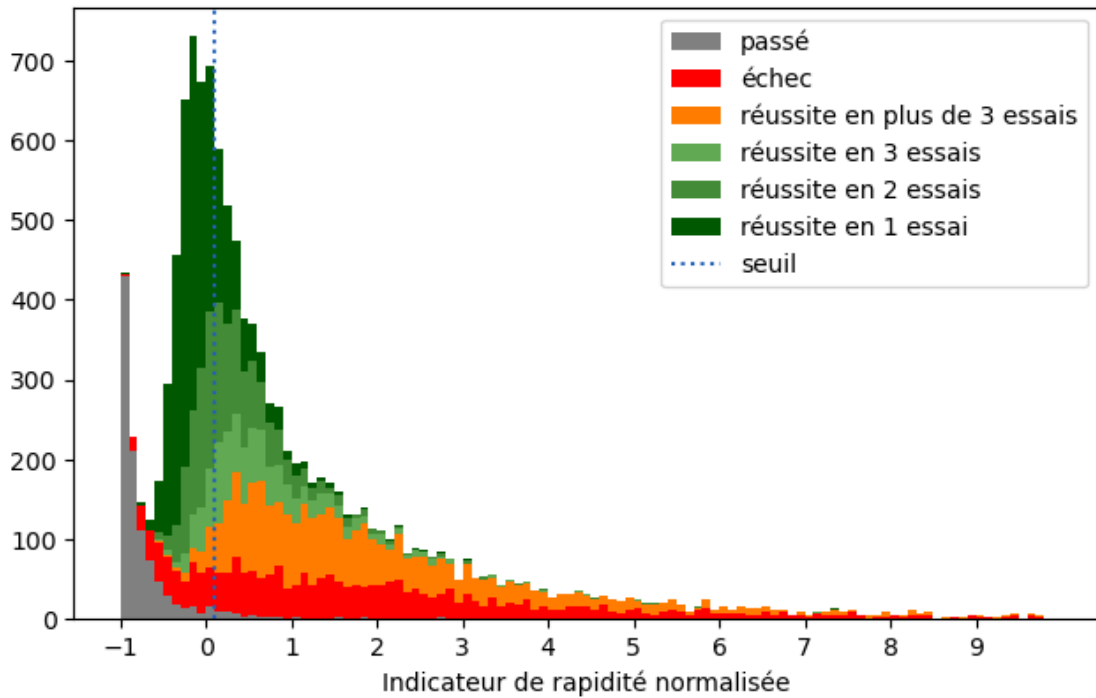
```
[47]: hist_rapidite (file_profiles, seuil = 1/4, debut= -1)
```



```
[48]: hist_rapidite (file_profiles, seuil = 0.2, debut= -1)
```



```
[49]: hist_rapidite (file_profiles, seuil = 0.1, debut= -1)
```



### 1.0.6 Comparaison avec l'analyse des extraits vidéos pour les sujets de l'échelle individuelle (toutes classes)

```
[64]: df = pd.read_csv('../data/algorea_2022/profiles_users_video.csv')
df_compare = df[["password", "profile", "profil_analyse_expert"]]
df_compare.head(5)
```

```
[64]:   password profile profil_analyse_expert
0  363spjfx  expert                    expert
1  363spjfx  expert                    expert
2  363spjfx  expert                    expert
3  363spjfx  passé                      échec
4  363spjfx  expert                    expert
```

```
[65]: df_compare["profil_analyse_expert"].value_counts()
```

```
[65]: expert                    487
accommodation                339
ajustement                   166
```

```

échec                146
passé                 59
non pris en charge   30
ajust / accommodation 20
sous-prog            8
expert / accommodation 7
expert/ajustement    4
sous-prog/ajustement 3
stratégie de décomposition 1
Name: profil_analyse_expert, dtype: int64

```

```
[68]: pd.crosstab(df_compare["profil_analyse_expert"], df_compare["profile"])
```

```
[68]: profile                accommodation  ajustement  expert  indéfini \
profil_analyse_expert
accommodation                286           24      4        24
ajust / accommodation         5           15      0         0
ajustement                   21          136      6         3
expert                        1            5     461        19
expert / accommodation        1            3      3         0
expert/ajustement             1            1      2         0
non pris en charge            0            0     23         7
passé                         0            0      0         0
sous-prog                     2            5      1         0
sous-prog/ajustement          2            1      0         0
stratégie de décomposition    0            1      0         0
échec                         0            0      0         0
```

```

profile                passé  échec
profil_analyse_expert
accommodation                1    0
ajust / accommodation        0    0
ajustement                   0    0
expert                       0    1
expert / accommodation       0    0
expert/ajustement            0    0
non pris en charge           0    0
passé                       57    2
sous-prog                    0    0
sous-prog/ajustement         0    0
stratégie de décomposition   0    0
échec                       10  136

```

```
[69]: df_cross_profiles = pd.crosstab(df_compare["profil_analyse_expert"],
↳df_compare["profile"], normalize='index')
round(df_cross_profiles,3)
```

```
[69]: profile      accommodation  ajustement  expert  indéfini  \
profil_analyse_expert
accommodation      0.844      0.071      0.012      0.071
ajust / accommodation  0.250      0.750      0.000      0.000
ajustement         0.127      0.819      0.036      0.018
expert             0.002      0.010      0.947      0.039
expert / accommodation  0.143      0.429      0.429      0.000
expert/ajustement   0.250      0.250      0.500      0.000
non pris en charge  0.000      0.000      0.767      0.233
passé              0.000      0.000      0.000      0.000
sous-prog         0.250      0.625      0.125      0.000
sous-prog/ajustement  0.667      0.333      0.000      0.000
stratégie de décomposition  0.000      1.000      0.000      0.000
échec              0.000      0.000      0.000      0.000
```

```
profile      passé  échec
profil_analyse_expert
accommodation  0.003  0.000
ajust / accommodation  0.000  0.000
ajustement     0.000  0.000
expert         0.000  0.002
expert / accommodation  0.000  0.000
expert/ajustement  0.000  0.000
non pris en charge  0.000  0.000
passé          0.966  0.034
sous-prog     0.000  0.000
sous-prog/ajustement  0.000  0.000
stratégie de décomposition  0.000  0.000
échec         0.068  0.932
```

```
[70]: df_cross_profiles_2 = pd.crosstab(df_compare["profile"],
↳df_compare["profil_analyse_expert"], normalize='index')
round(df_cross_profiles_2,3)
```

```
[70]: profil_analyse_expert  accommodation  ajust / accommodation  ajustement  \
profile
accommodation      0.897      0.016      0.066
ajustement         0.126      0.079      0.712
expert             0.008      0.000      0.012
indéfini           0.453      0.000      0.057
passé              0.015      0.000      0.000
échec              0.000      0.000      0.000

profil_analyse_expert  expert  expert / accommodation  expert/ajustement  \
profile
accommodation      0.003      0.003      0.003
ajustement         0.026      0.016      0.005
```

expert	0.922	0.006	0.004
indéfini	0.358	0.000	0.000
passé	0.000	0.000	0.000
échec	0.007	0.000	0.000

profil_analyse_expert	non pris en charge	passé	sous-prog	\
profile				
accommodation	0.000	0.000	0.006	
ajustement	0.000	0.000	0.026	
expert	0.046	0.000	0.002	
indéfini	0.132	0.000	0.000	
passé	0.000	0.838	0.000	
échec	0.000	0.014	0.000	

profil_analyse_expert	sous-prog/ajustement	stratégie de décomposition	échec
profile			
accommodation	0.006		0.000 0.000
ajustement	0.005		0.005 0.000
expert	0.000		0.000 0.000
indéfini	0.000		0.000 0.000
passé	0.000		0.000 0.147
échec	0.000		0.000 0.978

### 1.0.7 Reprise des traitements précédents pour l'ensemble des données de la thèse

```
[50]: dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv', parse_dates=[DATE])
dt_clean.fillna(EMPTY).head(5)
```

```
[50]:      ID password      id_group      id_problem \
0  741412  33wxj8ri  603931811100778560  808561348827292307
1  741426  33wxj8ri  603931811100778560  808561348827292307
2  741481  33wxj8ri  603931811100778560  808561348827292307
3  741481  33wxj8ri  603931811100778560  808561348827292307
4  741481  33wxj8ri  603931811100778560  808561348827292307
```

		problem	level	type	\
0	1 - Dans l'espace	[2022-3 white]	menu	load	
1	1 - Dans l'espace	[2022-3 white]	basic	load	
2	1 - Dans l'espace	[2022-3 white]	basic	submission	
3	1 - Dans l'espace	[2022-3 white]	basic	submission	
4	1 - Dans l'espace	[2022-3 white]	basic	submission	

		answer	score	\
0			0	
1			0	
2	<xml><block type="robot_start" id="space" x="0...		10	
3	<xml><block type="robot_start" id="space" x="0...		10	

```
4 <xml><block type="robot_start" id="space" x="0... 10
```

```
      date  time_session  time_attempt  time_load  context  grade  \  
0 2022-05-12 08:48:37      0           0           3    space   5em  
1 2022-05-12 08:48:40      3           3          16    space   5em  
2 2022-05-12 08:48:56     19          16           4    space   5em  
3 2022-05-12 08:48:56     19          16           4    space   5em  
4 2022-05-12 08:48:56     19          16           4    space   5em
```

```
      id_sol  id_prot  
0          0      0  
1          0      0  
2          1      1  
3          1      1  
4          1      1
```

```
[51]: dt_users = pd.read_csv('../data/data_these/users.csv')
```

```
[52]: dt_users.head(5)
```

```
[52]:   id_common  password  school_year  session  category  video  
0          1  8hm6qrhv      ce1 2021-1    blanc   True  
1          1  rfhu4i84      ce1 2021-2    blanc   True  
2          1  m4phaen4      ce1 2021-3    blanc   True  
3          2  f3avqege      ce1 2021-1    blanc   True  
4          2  k5gx9bfn      ce1 2021-2    blanc   True
```

```
[53]: dt_users['password'].nunique()
```

```
[53]: 909
```

```
[54]: dt_users['id_common'].nunique()
```

```
[54]: 381
```

```
[57]: dt_profils = pd.read_csv('../data/data_these/profils_nov23.csv')  
dt_profils.head(5)
```

```
[57]:   password      id_problem      problem  \  
0 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]  
1 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]  
2 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]  
3 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]  
4 33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]  
  
      level      profile  success  attempts_number  time_spent_on_question  \  
0  basic  accommodation  True           6                77
```



1	easy	expert	True	1	21
2	medium	accommodation	True	17	319
3	hard	échec	False	1	56
4	basic	expert	True	1	36

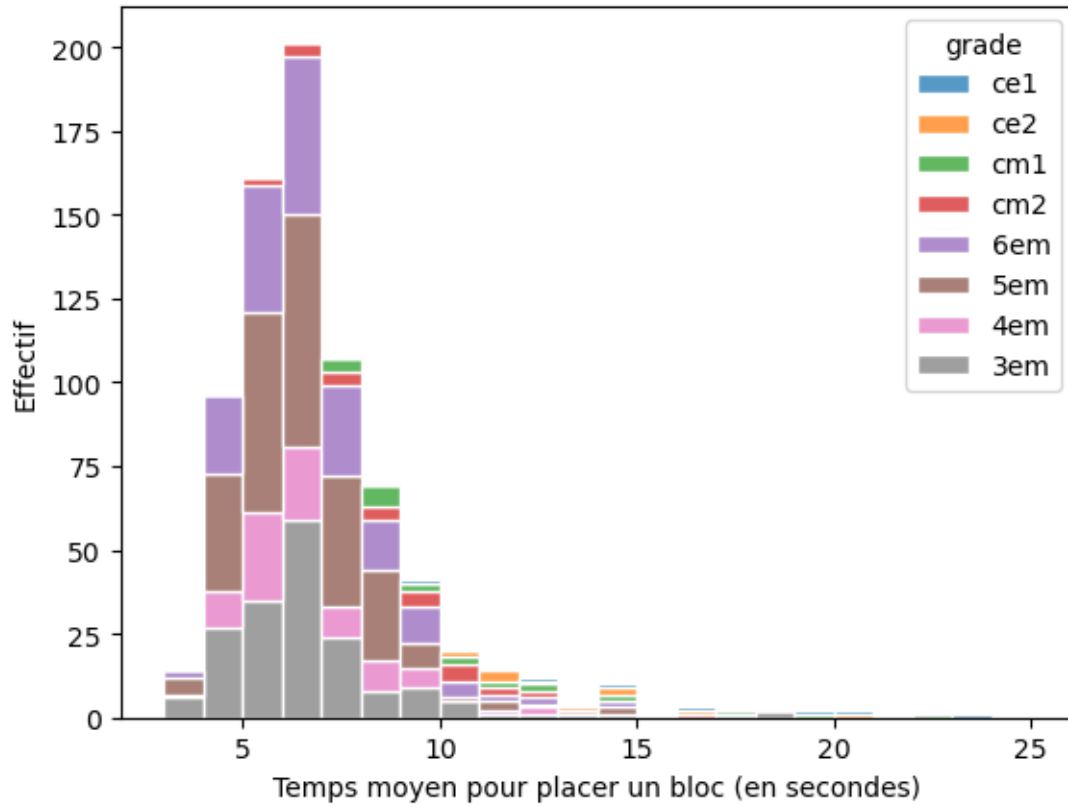
	mean_time_per_block	block_count_ref	ind_speed	grade	tx_reussite_concours
0	5.1	4.0	2.774510	5em	0.938
1	5.1	7.0	-0.411765	5em	0.913
2	5.1	13.0	3.811463	5em	0.799
3	5.1	11.0	-0.001783	5em	0.054
4	5.1	6.0	0.176471	5em	0.994

```
[56]: dt_profils['profile'].value_counts()
```

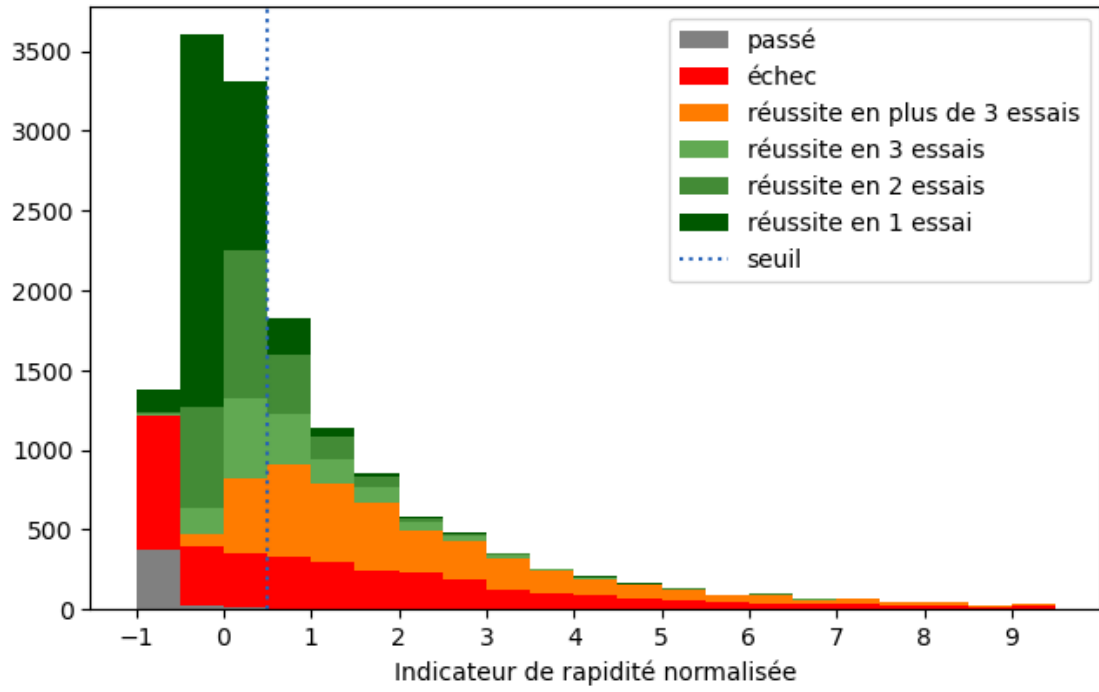
```
[56]: accommodation    4749
      échec              3590
      expert            3528
      ajustement        2249
      passé             434
      indéfini          341
      Name: profile, dtype: int64
```

```
[59]: dt_mean_time = dt_profils[['password', 'grade', 'mean_time_per_block']].
      ↳groupby(['password']).head(1)
```

```
[61]: sns.histplot(data=dt_mean_time,
                  x='mean_time_per_block',
                  hue='grade',
                  binwidth=1,
                  binrange=(3,25),
                  multiple='stack',
                  edgecolor='white',
                  hue_order=['ce1', 'ce2', 'cm1', 'cm2', '6em', '5em', '4em', '3em']
                  )
plt.gca().set_xlabel("Temps moyen pour placer un bloc (en secondes)")
plt.gca().set_ylabel("Effectif")
plt.savefig('hist_temps_moyen_par_bloc.pdf')
plt.show()
```



```
[62]: file_profiles = '../data/data_these/profils_nov23.csv'
      hist_rapidite (file_profiles, seuil = 0.5, debut= -1)
```



## **Annexe 8 : traitements relatifs aux programmes exécutés**

# traitements\_programmes\_executes\_annexe\_these

July 28, 2024

## 1 Annexe : Étude des programmes exécutés à l'échelle des classes

Les traitements regroupés dans cette annexe comprennent la généralisation de visualisations pour un puzzle particulier, la visualisation de frise chronologique, les traitements pour les programmes exécutés, le traitement pour guider le passage de l'échelle des classes à l'échelle du sujet.

```
[1]: import sys
      sys.path.append('../')

      from src import *
      import pandas as pd

      import re
      import seaborn as sns

      %load_ext autoreload
      %autoreload 2
```

pygame 2.1.2 (SDL 2.0.16, Python 3.9.15)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

2024-07-28 07:42:57.580039: I tensorflow/core/platform/cpu\_feature\_guard.cc:193]

This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2024-07-28 07:42:57.898085: E tensorflow/stream\_executor/cuda/cuda\_blas.cc:2981]

Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

2024-07-28 07:42:58.683545: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dLError: libnvinfer.so.7: cannot open shared object file: No such file or directory

2024-07-28 07:42:58.683636: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer\_plugin.so.7'; dLError: libnvinfer\_plugin.so.7: cannot open shared object file: No such file or directory

2024-07-28 07:42:58.683642: W

tensorflow/compiler/tf2tensorrt/utils/py\_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed properly.

### 1.0.1 Initialisation

```
[2]: classes_etude = ['cp', 'ce1', 'ce2', 'cm1', 'cm2', '6em', '5em', '4em', '3em']
```

```
[3]: classes_concours = ['cm1', 'cm2', '6em', '5em', '4em', '3em']
```

```
[4]: dt_concours = pd.read_csv('../data/data_these/resultats_concours.csv',  
    ↳dtype={"questionId sur chticode" : 'str'})
```

```
[5]: dt_tags = pd.read_csv('../data/data_these/tags_situations.csv',  
    ↳dtype={"questionId sur chticode" : 'str'})
```

```
[6]: dt_loop = dt_tags[dt_tags['classe_situation'].notna()]  
dt_loop = dt_loop.astype({'classe_situation': 'int',  
    ↳'nb_loops': 'int',  
    ↳'length_motif': 'int',  
    ↳'length_program': 'int'})
```

```
[7]: dt_loop.columns
```

```
[7]: Index(['id_situation', 'questionId sur chticode', 'Version', 'ref_svn',  
    ↳'Année', 'Tour', 'Catégorie', 'Numero problème', 'Effectif_all', 'all',  
    ↳'CM1', 'CM2', '6ème', '5ème', '4ème', '3ème', 'Contexte', 'Notion algo',  
    ↳'Nb cases du motif visuel', 'composition_motif', 'motif_case',  
    ↳'orientation', 'disposition', 'elt_id', 'adj_diff', 'rotation_mentale',  
    ↳'decor', 'classe_situation', 'nb_loops', 'length_motif',  
    ↳'length_program', 'before_loop', 'after_loop', 'last_instruction_loop',  
    ↳'answer_ref'],  
    dtype='object')
```

```
[8]: dt_users = pd.read_csv('../data/data_these/users.csv')  
dt_users.head(5)
```

```
[8]:
```

	id_common	password	school_year	session	category	video
0	1	8hm6qrhv	ce1	2021-1	blanc	True
1	1	rfhu4i84	ce1	2021-2	blanc	True
2	1	m4phaen4	ce1	2021-3	blanc	True
3	2	f3avqege	ce1	2021-1	blanc	True
4	2	k5gx9bfn	ce1	2021-2	blanc	True

```
[9]: caract_users_video = (dt_users['video']==True) & (dt_users['school_year'].  
    ↳isin(classes_etude))
```

```
list_passwords_video = dt_users[caract_users_video]['password'].to_list()
```

```
[10]: dt_profils = pd.read_csv('../data/data_these/profils_nov23.csv',  
    ↪dtype={'id_problem': str})
```

```
[11]: dt_profils.head(5)
```

```
[11]:
```

	password	id_problem	problem	\
0	33wxj8ri	36614660447619275	3 - Pousser les caisses [2022-3 white]	
1	33wxj8ri	36614660447619275	3 - Pousser les caisses [2022-3 white]	
2	33wxj8ri	36614660447619275	3 - Pousser les caisses [2022-3 white]	
3	33wxj8ri	36614660447619275	3 - Pousser les caisses [2022-3 white]	
4	33wxj8ri	613408804708837047	2 - Atteindre le drapeau [2022-3 white]	

	level	profile	success	attempts_number	time_spent_on_question	\
0	basic	accommodation	True	6	77	
1	easy	expert	True	1	21	
2	medium	accommodation	True	17	319	
3	hard	échec	False	1	56	
4	basic	expert	True	1	36	

	mean_time_per_block	block_count_ref	ind_speed	grade	tx_reussite_concours
0	5.1	4.0	2.774510	5em	0.938
1	5.1	7.0	-0.411765	5em	0.913
2	5.1	13.0	3.811463	5em	0.799
3	5.1	11.0	-0.001783	5em	0.054
4	5.1	6.0	0.176471	5em	0.994

```
[12]: list_levels = ['basic', 'easy', 'medium', 'hard']
```

```
[13]: list_problems = list(dt_profils['problem'].unique())
```

```
[14]: list_problems_2022 = list(dt_profils[dt_profils['problem'].str.  
    ↪contains('2022')]['problem'].unique())  
list_problems_2022
```

```
[14]: ['3 - Pousser les caisses [2022-3 white]',  
    '2 - Atteindre le drapeau [2022-3 white]',  
    '5 - Dans l'espace [2022-3 white]',  
    '1 - Dans l'espace [2022-3 white]',  
    '6 - Peindre le dessin [2022-3 white]',  
    '4 - Peindre le dessin [2022-3 white]',  
    '5 - Collecter les pierres [2022-1 white]',  
    '6 - Chauffer le château [2022-1 white]',  
    '4 - Planter des fleurs [2022-1 white]',  
    '3 - Planter des fleurs [2022-1 white]',  
    '1 - Chauffer le château [2022-1 white]',
```

```
'2 - Collecter les pierres [2022-1 white]',
'6 - Chauffer le château [2022-2 white]',
'5 - Distribuer le bois [2022-2 white]',
'2 - Distribuer le bois [2022-2 white]',
'3 - Ranger les billes [2022-2 white]',
'4 - Chauffer le château [2022-2 white]',
'1 - Ranger les billes [2022-2 white]',
'1 - Apporter du poisson [2022-2 yellow]',
'5 - Apporter du poisson [2022-2 yellow]',
'6 - Collecter les diamants [2022-2 yellow]',
'4 - Chauffer le château [2022-2 yellow]',
'2 - Chauffer le château [2022-2 yellow]',
'3 - Collecter les diamants [2022-2 yellow]',
'2 - Ranger les billes [2022-3 yellow]',
'5 - Distribuer le bois [2022-3 yellow]',
'3 - Dans l'espace [2022-3 yellow]',
'4 - Ranger les billes [2022-3 yellow]',
'1 - Distribuer le bois [2022-3 yellow]',
'6 - Dans l'espace [2022-3 yellow]',
'3 - Atteindre le drapeau [2022-1 yellow]',
'2 - Jouer aux dominos [2022-1 yellow]',
'5 - Préparer les colis [2022-1 yellow]',
'4 - Jouer aux dominos [2022-1 yellow]',
'1 - Préparer les colis [2022-1 yellow]'
```

## 1.0.2 Génération du diagramme de taux de réussite à l'échelle nationale pour un puzzle particulier

```
[ ]: def diagramme_reussite_categorie_puzzle(df, colonne_caract,
↳valeurs_caract_coul=[], puzzles=[], nom_fichier='', moyenne=False,
↳ecart_interquartile=True) :
    '''Procédure qui génère un diagramme avec la médiane et l'écart_
↳interquartile pour une série de puzzles

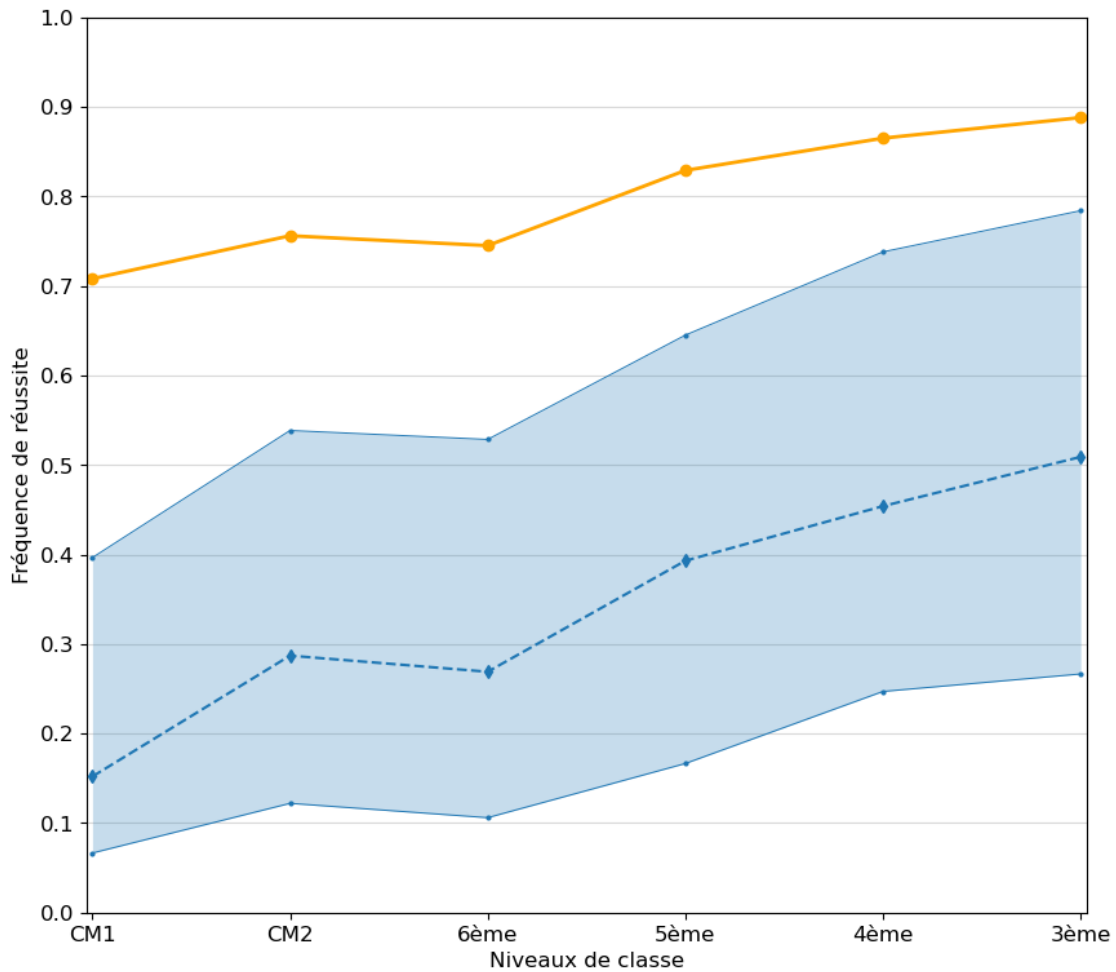
    Args:
        df : dataframe dans lequel sont stockées les données.
        colonne_caract: colonne qui contient la variable indépendante
        valeurs_caract_coul: liste de couples composés d'une modalité de la_
↳variable colonne_caract et d'une couleur
        puzzles: liste de couples composés de la valeur de l'id_situation_
↳pour lesquels on ajoute la courbe de fréquence de réussite, et d'une couleur
        nom_fichier: nom du fichier dans lequel exporter le diagramme
        moyenne: valeur booléenne qui indique s'il faut ajouter la courbe de_
↳la moyenne (défaut: False)
        ecart_interquartile: valeur booléenne qui indique s'il faut_
↳représenter l'écart interquartile (défaut: True)
```



```
'''
```

```
[15]: problem = 'white-05-veterinary-2'  
level = 'basic'  
id_puzzle = dt_loop[(dt_loop['ref_svn']==problem) &␣  
↳(dt_loop['Version']==level)].reset_index()['id_situation'][0]  
classe_puzzle = dt_loop[dt_loop['id_situation']==id_puzzle].  
↳reset_index()['classe_situation'][0]  
dt_classe_puzzle = dt_loop[dt_loop['classe_situation']==classe_puzzle]
```

```
[16]: diagramme_reussite_categorie_puzzle(dt_classe_puzzle,  
                                          'classe_situation',  
                                          valeurs_caract_coul=[(classe_puzzle,␣  
↳'#1f77b4')]),  
  
      puzzles=[(id_puzzle, 'orange')],  
      nom_fichier='',  
      moyenne=False,  
      ecart_interquartile=True  
      )
```



```
[17]: list_problems_loop = list(dt_loop['ref_svn'].unique())
```

```
[ ]: for problem in list_problems_loop :
      for level in list_levels :
          try :
              id_puzzle = dt_loop[(dt_loop['ref_svn']==problem) &
                                   (dt_loop['Version']==level)
                                   ].reset_index()['id_situation'][0]
              classe_puzzle = dt_loop[dt_loop['id_situation']==id_puzzle].
↳reset_index()['classe_situation'][0]
              dt_classe_puzzle =
↳dt_loop[dt_loop['classe_situation']==classe_puzzle]
              diagramme_reussite_categorie_puzzle(
                  dt_classe_puzzle,
                  'classe_situation',
                  valeurs_caract_coul=[(classe_puzzle, '#1f77b4')],
                  puzzles=[(id_puzzle, 'orange')],
                  nom_fichier = '../data/data_these/diag_reussite_national/' +
↳problem + '_' + level,
                  show = False,
                  moyenne=False,
                  ecart_interquartile=True
              )
          except :
              pass
```

### 1.0.3 Génération de frise chronologique de session

```
[18]: file_in = '../data/data_these/clean_nov23.csv'
      repertory_in = '../data/data_these'
      repertory_out = '../data/data_these/frises'
      password = '8w6qyjj5'
```

```
[ ]: def draw_user_timeline(file_in, repertoire_out, password, duree=45):
      '''Procédure qui génère la frise chronologique de la session d'un
↳utilisateur

      Args :
          file_in : fichier qui contient les champs password, time_session,
↳problem, level, type, grade
          repertoire_out : répertoire dans lequel sera généré la frise
          password : identifiant de session
          durée : durée de la session
      '''
```

```
[ ]: # La visualisation est générée avec pygame, elle est directement enregistrée
      ↪ dans un fichier,
      # sans être affichée dans le notebook
draw_user_timeline(file_in, repertory_out, password)
```

```
[19]: # Affichage de l'image dans le notebook à partir du fichier généré
img = mpimg.imread(repertory_out + '/' + password + '.png')
largeur_img = 15
rapport = img.shape[0] / img.shape[1]
hauteur_img = largeur_img * rapport

plt.figure(figsize=(largeur_img, hauteur_img))
plt.imshow(img)
plt.axis('off')
plt.show()
```



```
[ ]: list_passwords = ['6txeubkd',
                       '8hm6qrhv',
                       '8phfxfx9',
                       '9yvfarvb',
                       'e6jhmd63',
                       'eshsv5nf',
                       'ir7fix68',
                       'k8aeqdbd',
                       'mfijtyvb',
                       'mm77n8vq',
                       'r8yeffxu',
                       'sg9yuikt',
                       'unk4g5si',
                       'vt9d799f']
```

```
[ ]: len(list_passwords)
```

```
[ ]: for password in list_passwords :
      try :
          draw_user_timeline(file_in, repertory_out, password)
      except :
          print('erreur pour la session ' + password)
```

```
[ ]: draw_all_timelines(file_in, repertoire_out)
```

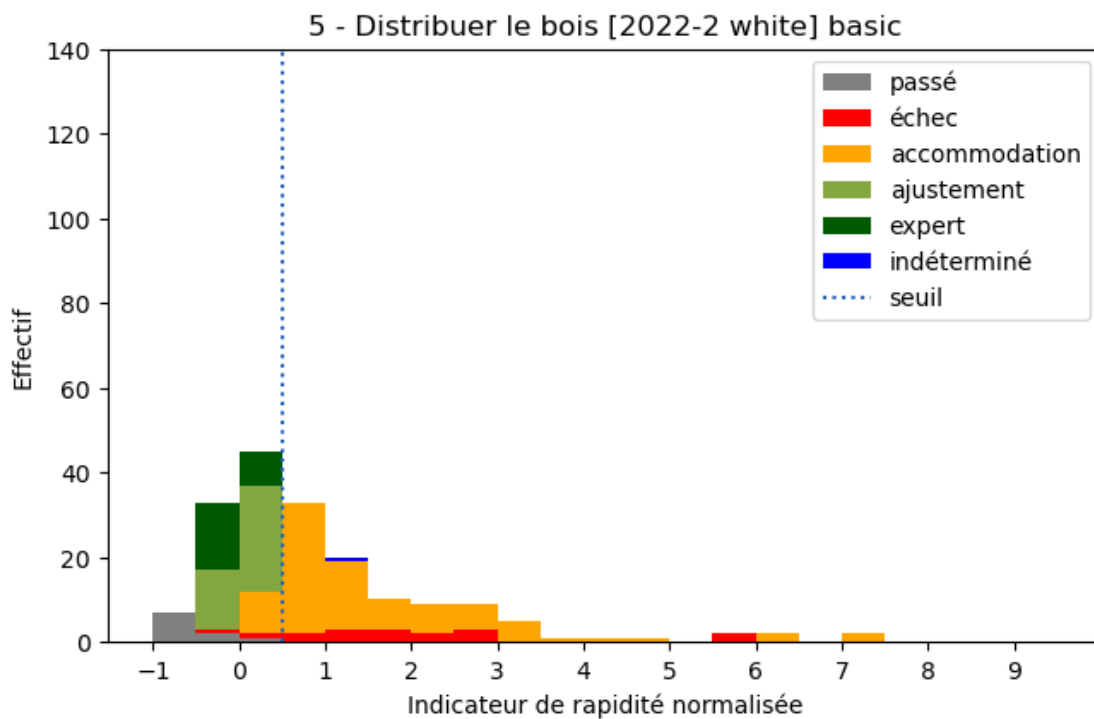
Pour générer un fichier pdf avec une visualisation de toutes les frises du répertoire : - enregistrer frises.tex dans le répertoire des frises - ouvrir un terminal dans ce répertoire et lancer la génération du fichier images : py2tex.py . images.tex - ouvrir le fichier frises.tex avec kile et le compiler - générer le pdf en cliquant sur PDFLatex dans kile

#### 1.0.4 Indicateur de rapidité normalisée

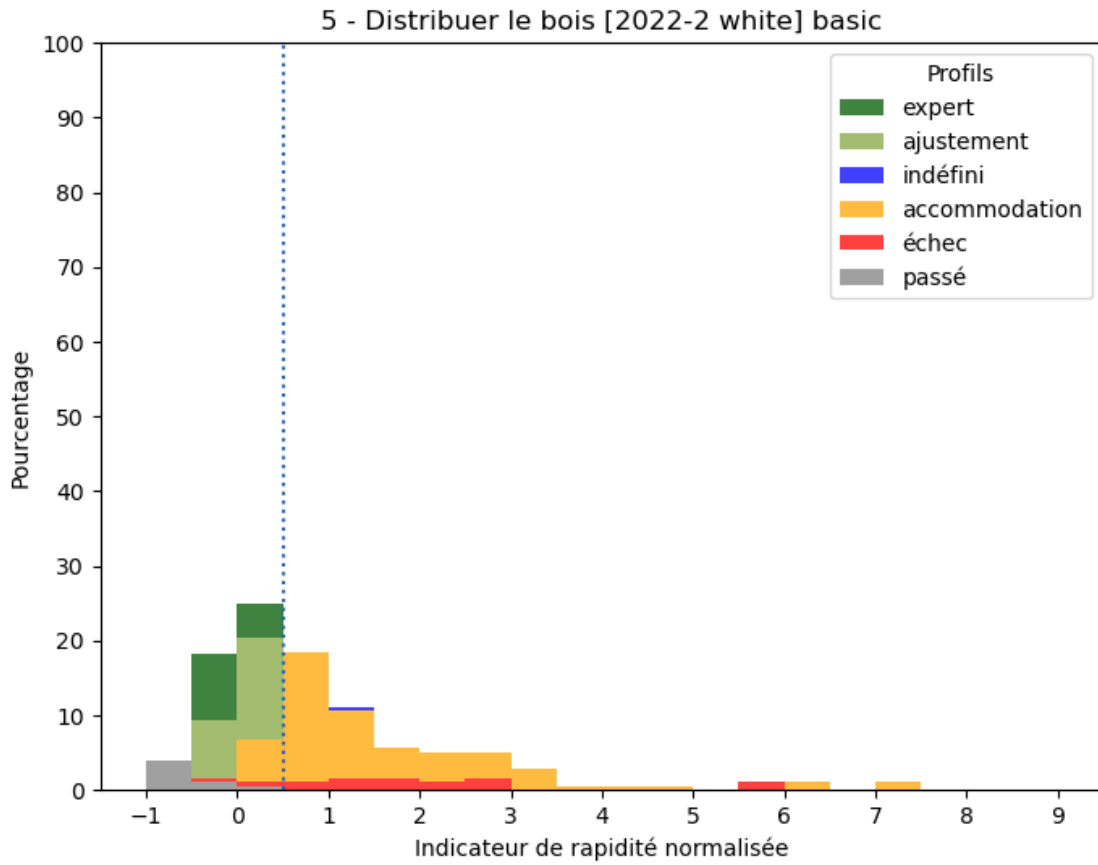
```
[20]: file_profiles = '../data/algorea_2022/profils_2022_sessions_removed.csv'
      repertory_out_hist_profils = '../data/algorea_2022/histogrammes_profils'
      repertory_out_hist_essais = '../data/algorea_2022/histogrammes_essais'
      problem = '5 - Distribuer le bois [2022-2 white]'
      level = 'basic'
```

Histogramme de la répartition des profils de confrontation sujet/puzzle pour un puzzle particulier

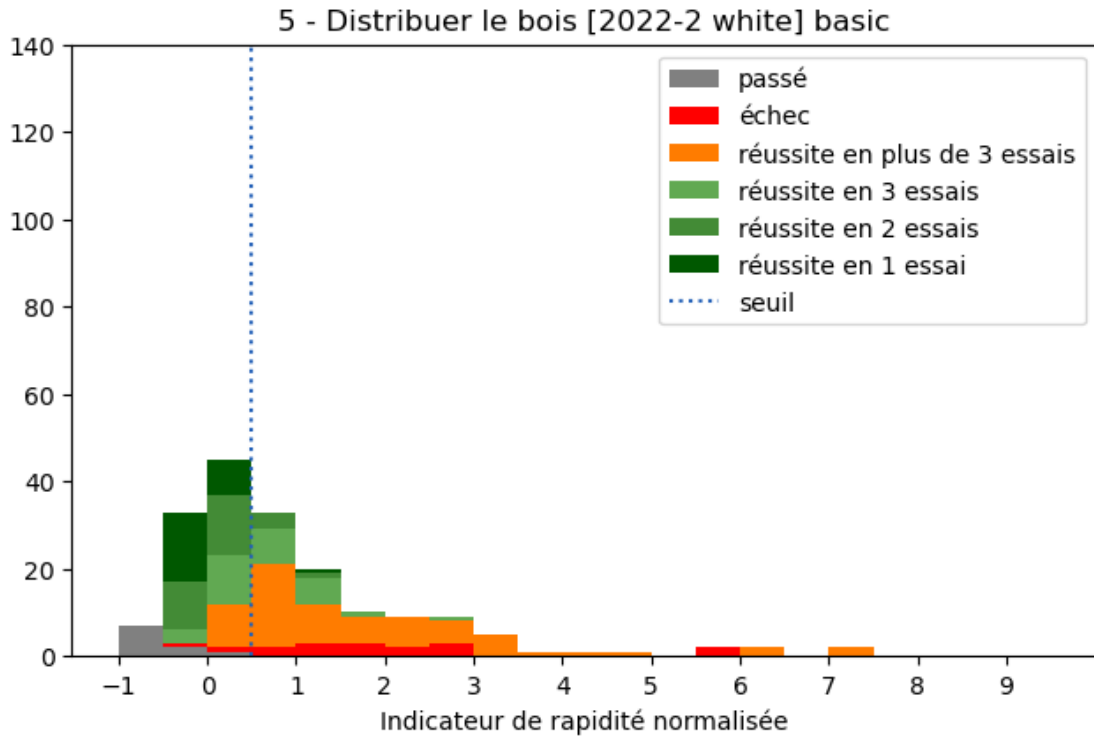
```
[21]: hist_effectifs_profils (file_profiles, repertory_out_hist_profils, problem, ↵
      ↪level,
      taille_effectif=150, pas_effectif=20)
```



```
[22]: hist_profils_puzzle_pourcentage(file_profiles, repertory_out_hist_profils, ↵
      ↪problem, level)
```



```
[23]: hist_rapidite(file_profiles, repertory_out_hist_essais, problem, level,
                 taille_effectif=150, pas_effectif=20)
```



```
[ ]: for problem in list_problems_2022 :
      for level in list_levels :
          try :
              hist_profils_puzzle_pourcentage (file_profiles,
↪repertory_out_hist_profils, problem, level,
                                              taille_effectif=150,
↪pas_effectif=20)
          except :
              print("erreur pour le problème" + problem + "version" + level)
```

```
[ ]: for problem in list_problems_2022 :
      for level in list_levels :
          try :
              hist_rapidite (file_profiles, repertory_out_hist_essais, problem,
↪level,
                           taille_effectif=150, pas_effectif=20)
          except :
              print("erreur pour le problème " + problem + " version " + level)
```

### 1.0.5 Répartition des différents profils de confrontation sujet / puzzle

Pour l'ensemble des puzzles

```
[54]: dt_rep = pd.DataFrame(dt_profils['profile'].value_counts())
dt_rep.rename(columns={'profile': 'all'}, inplace=True)
nb_confrontations = dt_rep['all'].sum()
dt_rep['all_percent'] = round((dt_rep['all']/nb_confrontations*100), 1)
dt_rep
```

```
[54]:
```

	all	all_percent
accommodation	4749	31.9
échec	3590	24.1
expert	3528	23.7
ajustement	2249	15.1
passé	434	2.9
indéfini	341	2.3

```
[45]: nb_confrontations
```

```
[45]: 14891
```

Répartition des différents profils de confrontation sujet / puzzle pour un puzzle particulier

```
[51]: dt_profil_puzzle = dt_profils[(dt_profils['problem']==problem) &
↳(dt_profils['level']==level)]
```

```
[56]: dt_rep_puzzle = pd.DataFrame(dt_profil_puzzle['profile'].value_counts())
label_column = 'p' + problem[0] + ' - ' + level
dt_rep_puzzle.rename(columns={'profile': label_column}, inplace=True)
nb_confrontations_puzzle = dt_rep_puzzle[label_column].sum()
dt_rep_puzzle[label_column + '_percent'] = round(dt_rep_puzzle[label_column]/
↳nb_confrontations_puzzle*100, 1)
dt_rep_puzzle
```

```
[56]:
```

	p5 - basic	p5 - basic_percent
accommodation	90	49.5
ajustement	40	22.0
expert	24	13.2
échec	20	11.0
passé	7	3.8
indéfini	1	0.5

```
[57]: nb_confrontations_puzzle
```

```
[57]: 182
```

Comparaison de la répartition pour un puzzle particulier et pour l'ensemble des puzzles

```
[58]: dt_compare = pd.concat([dt_rep, dt_rep_puzzle], axis=1)[['all_percent',
↳label_column + '_percent']]
dt_compare
```

```
[58]:
```

	all_percent	p5 - basic_percent
accommodation	31.9	49.5
échec	24.1	11.0
expert	23.7	13.2
ajustement	15.1	22.0
passé	2.9	3.8
indéfini	2.3	0.5

### 1.0.6 Visualisation des programmes soumis par ordre décroissant d'occurrences

```
[25]: nom_groupe = 'thèse'
problem = '5 - Distribuer le bois [2022-2 white]'
level = "basic"
nb_sol = 10
repertoire_out_freq = "../data/data_these/occurrences_prog_soumis"
file_tx_reussite_groupe = "../data/data_these/taux_reussite_groupe.csv"
file_sol_prot = '../data/data_these/table_sol_prot.csv'
file_ref = '../data/data_these/sol_ref.csv'
file_result = '../data/data_these/clean_nov23.csv'
file_profils = '../data/data_these/profils_nov23.csv'
```

```
[ ]: def visualisation_occurrences_solution (nom_groupe, file_tx_reussite_groupe,
↳file_result, file_profils, file_sol, problem, level, repertoire_out,
↳nb_sol=10) :
    '''Procédure qui génère la visualisation des programmes soumis à un
↳problème par fréquence décroissante du nombre d'occurrences

    Args :
        nom_groupe : chaine de caractère qui représente le nom du groupe pour
↳lequel la visualisation est générée
        file_tx_reussite_groupe : fichier csv qui contient les taux de réussite
↳aux différents puzzles pour le groupe considéré
        file_result : fichier de type result
        file_profils : fichier de type profile
        file_sol : fichier qui contient les identifiant de solutions
↳prototypique
        problem : chaine de caractère qui représente le nom du problème
        level : chaine de caractère qui contient la version du problème
        repertoire_out : répertoire dans lequel stocker la visualisation générée
        nb_sol : nombre de solutions les plus fréquentes à faire apparaître sur
↳la visualisation
    '''
```

```
[ ]: # La visualisation est générée avec pygame, elle est directement enregistrée
↳dans un fichier,
# sans être affichée dans le notebook
```





```
[ ]: for problem in list_problems :
      for level in list_levels :
          try :
              visualisation_occurrences_solution (nom_groupe,
                                                    file_tx_reussite_groupe,
                                                    file_result,
                                                    file_profils,
                                                    file_sol_prot,
                                                    problem,
                                                    level,
                                                    repertoire_out_freq)

          except :
              print("erreur pour le problème " + problem + " version " + level)
```

### 1.0.7 Générer une visualisation d'une liste de programmes prototypes

Cette visualisation est utilisée ponctuellement pour des vérifications manuelles

```
[27]: dt_prot = pd.read_csv(file_sol_prot)
      dt_prot.head(5)
```

```
[27]:
```

	id_problem	problem	context	level	\
0	1555753216143902	3 - Pousser les caisses	[2021 yellow]	sokoban	basic
1	1555753216143902	3 - Pousser les caisses	[2021 yellow]	sokoban	easy
2	1555753216143902	3 - Pousser les caisses	[2021 yellow]	sokoban	easy
3	1555753216143902	3 - Pousser les caisses	[2021 yellow]	sokoban	easy
4	1555753216143902	3 - Pousser les caisses	[2021 yellow]	sokoban	easy

	sol_prototypique	nb_occurrences	id_sol	\
0	<xml><block type="robot_start" id="sokoban" x=...	2	[8160]	
1	<xml><block type="robot_start" id="sokoban" x=...	1	[7554]	
2	<xml><block type="robot_start" id="sokoban" x=...	1	[7553]	
3	<xml><block type="robot_start" id="sokoban" x=...	1	[7552]	
4	<xml><block type="robot_start" id="sokoban" x=...	1	[18101]	

	id_prot	valide	freq	freq_users	mean_accomm	std_accomm	mean_essais	\
0	8160	True	1.00	1.000	-0.12	0.07	1.0	
1	7554	False	0.05	0.167	1.31	NaN	5.0	
2	7553	False	0.05	0.167	1.31	NaN	5.0	
3	7552	False	0.05	0.167	1.31	NaN	5.0	
4	18101	False	0.05	0.167	1.79	NaN	6.0	

	std_essais	id_sujets_visio	type_sol
0	0.0	{'aigvrpue': 'expert', 'pekx5imk': 'expert'}	optimal
1	NaN	{'9yiiqsjg': 'accommodation'}	erreur

2	NaN	{'9yiiqsjg': 'accommodation'}	erreur
3	NaN	{'9yiiqsjg': 'accommodation'}	erreur
4	NaN	{'pekk5imk': 'accommodation'}	erreur

```
[28]: problem = '5 - Distribuer le bois [2022-2 white]'
level = "basic"
titre= 'test'
file_out = '../data/data_these/erreurs_2022t2p5v1.png'
```

```
[29]: list_id_prot = list(dt_prot[(dt_prot['problem']==problem) &
(dt_prot['level']==level) &
(dt_prot['valide']==False) &
(dt_prot['nb_occurrences']>5)
].sort_values('nb_occurrences', ascending=False)['id_prot'].unique())
```

```
[ ]: def generer_visualisation_list_id_prot(title, dt_prot, problem, level,
↳list_id_prot, file_out) :
    """ Procedure qui génère la visualisation des images de solution pour la
↳liste fournie

    Args :
        title(string) : titre à donner à la visualisation
        dt_prot(pandas.dataframe): contient un champ avec les id_prot des
↳solutions prototypiques
        et un champ avec les sol xml
        problem(string) : puzzle pour lequel on considère les solutions
        level(string) : version du puzzle pour laquelle on considère les
↳solutions
        list_id_prot : liste des id de solutions prototypiques qui vont
↳apparaître sur la visualisation
        file_out : fichier d'enregistrement de l'image

    """
```

```
[ ]: # La visualisation est générée avec pygame, elle est directement enregistrée
↳dans un fichier,
# sans être affichée dans le notebook

generer_visualisation_list_id_prot(titre, dt_prot, problem, level,
↳list_id_prot, file_out)
```

```
[30]: # Affichage de l'image dans le notebook à partir du fichier généré
img = mpimg.imread(file_out)
largeur_img = 15
rapport = img.shape[0] / img.shape[1]
hauteur_img = largeur_img * rapport
```

```
plt.figure(figsize=(largeur_img, hauteur_img))
plt.imshow(img)
plt.axis('off')
plt.show()
```



### 1.0.8 Programmes non valides différents pour un puzzle donné

```
[ ]: file_sol_prot = '../data/data_these/table_sol_prot.csv'
      problem = '5 - Distribuer le bois [2022-2 white]'
      level = "basic"
```

```
[ ]: def list_id_prot_distincts(file_sol, problem, level, column_id='id_prot',
    ↪valide=False):
    '''Fonction qui renvoie par défaut la liste des id de solutions
    ↪prototypiques pour les programmes invalides

    Args :
        file_sol (fichier csv): fichier qui contient la colonne id_prot
        problem (string): nom du problème
        level (string): version du problème
        column_id (string): id_prot par défaut, mais en remplaçant par id_sol,
    ↪on obtient toutes les solutions
        valide (boolean): False par défaut, mais peut être passé à True pour
    ↪obtenir la liste des id de programmes valides
    '''
```

Nombre de programmes prototypiques non valides différents exécutés pour un puzzle donné

```
[33]: list_errors = list_id_prot_distincts(file_sol_prot, problem, level)
      len(list_errors)
```

[33]: 277

Nombre de programmes prototypiques erronés différents pour un puzzle donné (on enlève les programmes partiels)

```
[32]: len(dt_prot[(dt_prot['problem']==problem) &
      (dt_prot['level']==level) &
```

```
(dt_prot['type_sol']=='erreur')
])
```

[32]: 272

### 1.0.9 Recherche des programmes contenant un type d'erreur donné

Première phase de datégorisation manuelle à partir de la visualisation des programmes erronés soumis par ordre décroissant d'occurrences : - déterminer les labels - déterminer pour chaque erreur ce qui la spécifie, chaînes à inclure et exclure, contrainte sur le nombre de blocs

```
[2]: dt_prot = pd.read_csv('../data/data_these/table_sol_prot.csv')
```

```
[ ]: def rechercher_id_sol_erreur(dt_prot,
                                problem,
                                level,
                                liste_chaines_inclure,
                                liste_chaines_exclure,
                                liste_contraintes_bloc = []):
    """Fonction qui cherche les id_sol pour lesquelles on trouve dans le
    ↪ programme soumis
    les chaines de la liste à inclure mais pas celles de la liste à exclure, et
    ↪ qui respectent
    les contraintes données en nombre de blocs pour certains blocs spécifiques

    Args :
        dt_prot(pandas.dataframe): contient un champ avec les id_prot des
    ↪ solutions prototypiques
        et un champ avec les sol xml
        problem(string) : puzzle pour lequel on considère les solutions
        level(string) : version du puzzle pour laquelle on considère les
    ↪ solutions
        liste_chaines_inclure : liste des chaines dont l'une doit être présente
    ↪ dans la sol xml
        liste_chaines_exclure : liste des chaines de caractères qui ne doivent
    ↪ pas être présente
        dans la solution xml

    Returns :
        list_id_prot : liste des sol qui satisfont les contraintes

    """
```

```
[3]: problem = '4 - Chauffer le château [2022-2 white]'
id_problem = dt_prot[dt_prot['problem'] == problem].
    ↪ reset_index()['id_problem'][0]
context = dt_prot[dt_prot['problem'] == problem].reset_index()['context'][0]
```

```

level = 'medium'
label_erreur ="Boucle : manque une instruction en fin de boucle, qui correspond
↳à la remise du robot dans la même position par rapport au motif visuel qu'en
↳entrée de boucle."
liste_contraintes_bloc = [
]
liste_chaines_inclure = [
    '<block type="robot_start" id="castle" x="0" y="0"><next><block
↳type="control_repeat"><value name="TIMES"><shadow type="math_number"><field
↳name="NUM">(\d+)</field></shadow></value><statement name="SUBSTACK"><block
↳type="dropPlatformInFront">(<next><block type="forward">)?<next><block
↳type="forward"><next><block type="forward"><next><block
↳type="withdrawObject"><next><block type="backwards"><next><block
↳type="dropObject"></block>(?: (?!<statement>.)*)</statement></block>',
    '<block type="robot_start" id="castle" x="0" y="0"><next><block
↳type="control_repeat"><value name="TIMES"><shadow type="math_number"><field
↳name="NUM">(\d+)</field></shadow></value><statement name="SUBSTACK"><block
↳type="dropPlatformInFront"><next><block type="control_repeat"><value
↳name="TIMES"><shadow type="math_number"><field name="NUM">[2-3]</field></
↳shadow></value><statement name="SUBSTACK"><block type="forward"></block></
↳statement><next><block type="withdrawObject"><next><block
↳type="backwards"><next><block type="dropObject"></block>'
]
liste_chaines_exclure = [
]

liste_id_prot = rechercher_id_sol_erreur(dt_prot,
                                     problem,
                                     level,
                                     liste_chaines_inclure,
                                     liste_chaines_exclure,
                                     liste_contraintes_bloc)

nb_id_prot = len(liste_id_prot)

liste_id_prot

```

[3]: [229, 228, 430, 1296, 10393, 19799, 2219, 19798, 14443, 19939, 14444]

[5]: *#génération et affichage de la visualisation des programmes contenant l'erreur*  
↳spécifiée

```

file_out = '../data/data_these/provisoire/erreur.png'

generer_visualisation_list_id_prot(
    problem + '_' + level,
    dt_prot,

```

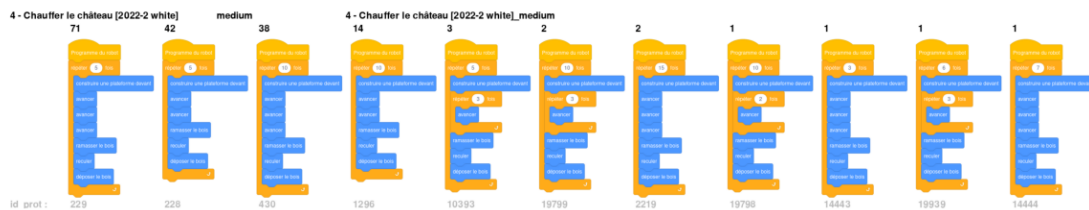
```

    problem,
    level,
    liste_id_prot,
    file_out
)

img = mpimg.imread(file_out)
largeur_img = 15
rapport = img.shape[0] / img.shape[1]
hauteur_img = largeur_img * rapport

plt.figure(figsize=(largeur_img, hauteur_img))
plt.imshow(img)
plt.axis('off')
plt.show()

```



Utilisateurs qui ont commis l'erreur : leur nombre, leur fréquence, la liste des passwords

```
[6]: dt_clean = pd.read_csv('../data/data_these/clean_nov23.csv')
```

```
[ ]: def calculer_freq_utilisateurs_liste_id_prot(dt_clean, problem, level,
→liste_id_prot) :
    """fonction qui calcule la fréquence des utilisateurs qui ont soumis au
→moins une solution
    parmi la liste d'id donnée en entrée

    Args :
        dt_clean(pandas.dataframe): contient un champ avec les id_prot des
→solutions prototypiques
        problem(string) : puzzle pour lequel on considère les solutions
        level(string) : version du puzzle pour laquelle on considère les
→solutions
        liste_id_prot : liste des id de solutions prototypiques pour laquelle la
→fonction calcule
        la fréquence d'utilisateurs

    Returns :
        (nb_users, freq_users, list_passwords) : triplet qui contient :
```

```

        - un entier représentant le nombre d'utilisateur qui ont soumis au
↳moins une solution
        parmi la liste d'id donnée en entrée
        - un flottant représentant la fréquence des utilisateurs qui ont soumis
↳au moins une solution
        parmi la liste d'id donnée en entrée
        - une liste de chaînes de caractère contenant les identifiants des
↳utilisateurs qui ont
        soumis au moins une solution parmi la liste d'id donnée en entrée
        """

```

```

[7]: nb_users, freq_users, liste_passwords =
↳calculer_freq_utilisateurs_liste_id_prot(dt_clean, problem, level,
↳liste_id_prot)

```

```

[8]: nb_users

```

```

[8]: 108

```

```

[9]: freq_users

```

```

[9]: 0.617

```

```

[ ]: liste_passwords

```

Ajout de l'erreur documentée au dataframe des erreurs

```

[13]: dt_err = pd.read_csv('../data/data_these/doc_erreurs.csv')
colonne = ['id_problem', 'problem', 'level', 'context', 'label_erreur',
↳'freq_users', 'nb_users', 'nb_id_prot', 'list_id_prot', 'list_passwords']

```

```

[14]: erreur = pd.Series([id_problem,
                           problem,
                           level,
                           context,
                           label_erreur,
                           freq_users,
                           nb_users,
                           nb_id_prot,
                           liste_id_prot,
                           liste_passwords
                           ], index=colonne)

```

```

[15]: erreur

```

```

[15]: id_problem          980641712306666163
      problem           4 - Chauffer le château [2022-2 white]

```



```

level                                medium
context                              castle
label_erreur      Boucle : manque une instruction en fin de bouc...
freq_users                0.617
nb_users                108
nb_id_prot                11
list_id_prot      [229, 228, 430, 1296, 10393, 19799, 2219, 1979...
list_passwords      [34r3vvnw, 375ptvqv, 38bcjvyt, 3auspvrn, 3jnck...
dtype: object

```

```
[16]: dt_err = pd.concat([dt_err, erreur.to_frame().T], ignore_index=True)
```

```
[18]: dt_err.tail(5)
```

```
[18]:
```

	id_problem	problem	level	context	label_erreur	freq_users	nb_users	nb_id_prot	list_id_prot	list_passwords
315	938454838520278869	2 - Distribuer le bois [2022-2 white]	basic	veterinary	erreur dans le sens de pivotement	0.063	12	10	[2022, 3595, 9111, 7746, 1276, 22670, 13346, 2...	['4as76ya5', '4dagm4an', '5k39dtsc', '63nr9gis...
316	808561348827292307	1 - Dans l'espace [2022-3 white]	easy	space	Il manque un déplacement avant ou après une in...	0.221	29	11	[7317, 1977, 5119, 6558, 15747, 25924, 9984, 1...	['48c6kxyy', '4m5wuvkk', '7eepfppc', '7ijwabr5...
317	808561348827292307	1 - Dans l'espace [2022-3 white]	medium	space	Il manque un déplacement avant ou après une in...	0.045	6	3	[6922, 16054, 6923]	['8yhiq3yd', 'kwiq9giu', 'mi5knkgw', 'qwhqin9g...
318	808561348827292307	1 - Dans l'espace [2022-3 white]	medium	space	erreur dans le sens de pivotement de pivotement	0.301	40	21	[360, 6171, 6922, 11867, 14406, 8993, 9473, 51...	['37vpf37c', '7eepfppc', '88e76ywa', '89x39b3a...
319	980641712306666163	4 - Chauffer le château [2022-2 white]	medium	castle	Boucle : manque une instruction en fin de bouc...	0.617	108	11	[229, 228, 430, 1296, 10393, 19799, 2219, 1979...	[34r3vvnw, 375ptvqv, 38bcjvyt, 3auspvrn, 3jnck...

Erreurs les plus fréquentes pour une classe de problèmes données

```
[21]: dt_cl3 = dt_err[(dt_err['problem'].str.contains('5 -')) &
                    (dt_err['level'].isin(['basic', 'easy'])) &
                    (dt_err['problem'].str.contains('2022')) &
```

```

(dt_err['freq_users']>=0.05)
]
dt_cl3 = dt_cl3.sort_values(by='freq_users', ascending=False)
dt_cl3.head(5)

```

```

[21]:
      id_problem      problem  level \
184  74670111826351495  5 - Distribuer le bois [2022-2 white]  easy
171  74670111826351495  5 - Distribuer le bois [2022-2 white]  basic
196  748042633662361316      5 - Dans l'espace [2022-3 white]  basic
153  169929086869040532  5 - Collecter les pierres [2022-1 white]  basic
183  74670111826351495      5 - Distribuer le bois [2022-2 white]  easy

      context      label_erreur  freq_users \
184  veterinary  Boucle : il manque les instructions après la b...  0.602
171  veterinary  Boucle : manque une instruction de pivotement ...  0.594
196      space  Boucle : nombre d'itérations laissé à la valeu...  0.449
153      gems  Boucle : manque une instruction de pivotement ...  0.439
183  veterinary  Boucle : nombre d'itérations laissé à la valeu...  0.385

      nb_users  nb_id_prot      list_id_prot \
184      97      165  [461, 1232, 3158, 1626, 1625, 234, 4495, 935, ...
171     104      49  [232, 1273, 2838, 1227, 4098, 450, 4126, 925, ...
196      57      94  [85, 6135, 12513, 2013, 1895, 43, 13099, 13716...
153      83      27  [2742, 143, 6407, 5216, 20700, 3730, 4264, 323...
183      62      74  [461, 3158, 4495, 5814, 10968, 11305, 1278, 20...

      list_passwords
184  ['34r3vvnw', '38bcjvyt', '3auspvrn', '3avv4smh...
171  ['34r3vvnw', '38bcjvyt', '3hfvtf44', '3jnckjwb...
196  ['33wxj8ri', '45b4q98f', '48c6kxyy', '4d67scfh...
153  ['33yjhik9', '4m9c5jh7', '56d9fp4u', '5r4k7qna...
183  ['38bcjvyt', '3auspvrn', '3avv4smh', '3dbsp8ac...

```

### 1.0.10 Passage de l'échelle des classes à l'échelle du sujet

Recherche du profil des sujets pour lesquels on dispose d'un enregistrement vidéo, en vue de guider l'analyse de l'activité sur un problème

```

[34]: def profils_users_video(dt_profils, list_passwords_video, problem, level) :
      return dt_profils[(dt_profils['problem']== problem) &
          (dt_profils['level'] == level) &
          (dt_profils['password'].isin(list_passwords_video))
          ][['password', 'grade', 'profile', 'attempts_number',
          ↪'time_spent_on_question', 'mean_time_per_block', 'ind_speed']]

```

```

[35]: problem = '5 - Distribuer le bois [2022-2 white]'
      level = 'basic'

```

```
profils_users_video(dt_profils, list_passwords_video, problem, level)
```

```
[35]: password grade      profile  attempts_number  time_spent_on_question  \  
63      363spjfx  6em      expert           1                        48  
712     43tt49bc  3em      expert           1                        33  
1077    4m6cq598  cm2      expert           1                        74  
1151    4r4hjrce  cm2      ajustement       2                        81  
1377    58mipipc  4em      expert           1                        37  
1918    6maj8qwh  cm1      ajustement       3                        102  
5090    ctdje8pj  cm2      expert           1                        77  
7315    haxyd6pf  cm1      échec            9                        283  
7649    i6nmn46d  6em      expert           1                        47  
7941    ipwtpr4u  cm1      ajustement       3                        129  
8629    k5iurkg4  cm1      accommodation    2                        190  
9180    m6p7i7xh  6em      accommodation    3                        130  
10113   nnk99dr   5em      expert           1                        74  
10227   nyav8mk8  ce2      accommodation    5                        284  
10258   p4nyn4fc  ce1      accommodation    12                       554  
11342   r7xxmtqw  6em      accommodation    6                        809  
11693   rufgman8  5em      expert           1                        57  
11892   scaifywq  6em      ajustement       2                        38
```

```
mean_time_per_block  ind_speed  
63                    5.9 -0.186441  
712                   4.2 -0.214286  
1077                  5.7  0.298246  
1151                  10.1 -0.198020  
1377                  6.0 -0.383333  
1918                  8.4  0.214286  
5090                  9.1 -0.153846  
7315                  12.8  1.210938  
7649                  6.9 -0.318841  
7941                  10.5  0.228571  
8629                  7.8  1.435897  
9180                  7.1  0.830986  
10113                 5.4  0.370370  
10227                 10.3  1.757282  
10258                 6.7  7.268657  
11342                 4.9  15.510204  
11693                 7.0 -0.185714  
11892                 3.7  0.027027
```

Génération de la liste des identifiants de session pour un profil particulier

```
[36]: def list_users_video_profil_puzzle(dt_profils, list_passwords_video, problem,   
    ↪level, profil) :  
    return list(dt_profils[(dt_profils['problem']== problem) &
```

```
(dt_profil['level'] == level) &
(dt_profil['password'].isin(list_passwords_video)) &
(dt_profil['profile'] == profil)
]['password']
```

```
[37]: profil = 'expert'
list_users_video_profil_puzzle(dt_profil, list_passwords_video, problem,
↳level, profil)
```

```
[37]: ['363spjfx',
'43tt49bc',
'4m6cq598',
'58mipipc',
'ctdje8pj',
'i6nmn46d',
'nnk99drr',
'rufgman8']
```

Programmes exécutés par les sujets pour lesquels on dispose d'un enregistrement vidéo

```
[ ]: def sessions_and_profil_list_users_video(file_sol, problem, level,
↳list_id_prot, nb_id_prot=10) :
    '''Fonction qui renvoie un array avec les nb_id_prot les plus fréquents de
↳la liste list_id_prot
    associé aux utilisateurs pour lesquels existe un enregistrement vidéo et
↳qui ont un profil ajustement, accommodation ou échec

    Args :
        file_sol (fichier csv): fichier dans lequel se trouve un champ id_prot,
↳nb_occurences, id_sujets_visio
        problem (string): nom du problème
        level (string): version du problème
        list_id_prot:
        nb_erreurs (int): nombre d'erreurs les plus fréquentes pour lesquelles
↳on veut ce résultat
    '''
```

```
[38]: sessions_and_profil_list_users_video(file_sol_prot, problem, level,
↳list_errors, nb_id_prot=10)
```

```
[38]: array([[232,
{'4r4hjrce': 'ajustement', '6maj8qwh': 'ajustement', 'k5iurkg4':
'accommodation', 'nyav8mk8': 'accommodation', 'r7xxmtqw': 'accommodation'}],
[777, '{"ipwtpr4u': 'ajustement'}"],
[1273, '{"ipwtpr4u': 'ajustement'}"],
[449, '{}'],
[2838,
```

```

    '{"nyav8mk8': 'accommodation', 'r7xxmtqw': 'accommodation'}"],
    [3152, '{"haxyd6pf': 'échec', 'm6p7i7xh': 'accommodation'}"],
    [1227, '{}'],
    [450, '{"r7xxmtqw': 'accommodation'}"],
    [4098, '{}'],
    [4127, '{"p4nyn4fc': 'accommodation'}"]], dtype=object)

```

### 1.0.11 Visualisation de la suite des programmes soumis par un sujet sur un puzzle

```

[42]: repertory = '../data/data_these'
      file_result = 'clean_nov23.csv'
      file_profils = 'profils_nov23.csv'
      file_prot = 'table_sol_prot.csv'
      password = 'nyav8mk8'
      problem = '5 - Distribuer le bois [2022-2 white]'
      level = 'basic'

```

```

[ ]: def visualisation_prog(repertory, file_result, file_profils, file_prot,
    ↪password, problem, level):
    '''Procédure qui génère la suite des programmes soumis par un utilisateur,
    ↪sur un problème donné

    Args :
        repertory : répertoire qui contient les fichiers file_result et
    ↪file_profils, et dans lequel est généré la visualisation
        file_result : fichier qui contient les champs password, problem, level,
    ↪grade
        file_profils : fichier qui contient les champs password, problem,
    ↪level, time_spent_on_question, ind_speed, mean_time_per_block
        file_prot : fichier qui contient la banque de solutions prototypiques
        password : code de session
        problem : nom du problème dont on veut générer la suite de programmes
    ↪soumis
        level : version du problème
    '''

```

```

[ ]: # La visualisation est générée avec pygame, elle est directement enregistrée
    ↪dans un fichier,
    # sans être affichée dans le notebook

    visualisation_prog(repertory, file_result, file_profils, file_prot, password,
    ↪problem, level)

```

```

[43]: # Affichage de l'image dans le notebook à partir du fichier généré
      img = mpimg.imread(repertory + '/visualisation_prog_soumis/' + password + '/' +
    ↪password + '_' + problem[0] + '_' + level + '.png')

```

```

largeur_img = 15
rapport = img.shape[0] / img.shape[1]
hauteur_img = largeur_img * rapport

plt.figure(figsize=(largeur_img, hauteur_img))
plt.imshow(img)
plt.axis('off')
plt.show()

```

nyav8mk8 ce2

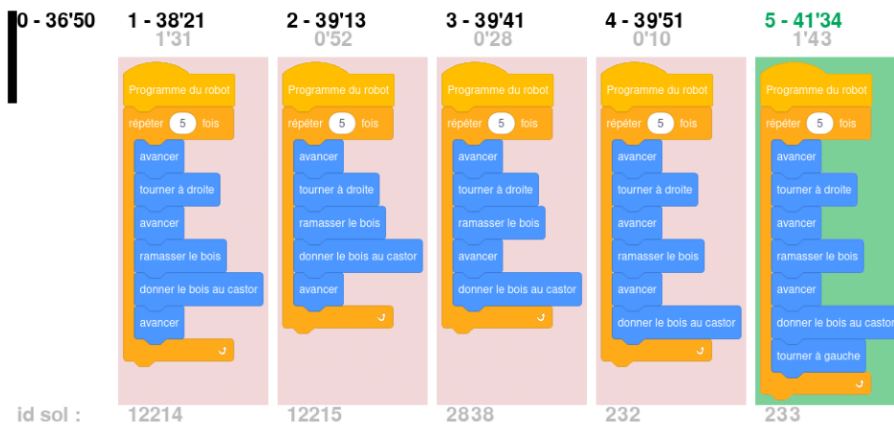
score global : 120

temps moyen par bloc : 10.1

5 - Distribuer le bois [2022-2 white] basic

durée : 4'44

ind. rapidité normalisée : 1.81



```

[ ]: for password in list_passwords_video :
      for problem in list_problems :
          for level in list_levels :
              try:
                  visualisation_prog(repertory, file_result, file_profils,
↪file_prot, password, problem, level)
              except:
                  pass

```

## **Annexe 9 : analyse préalable des extraits vidéo**

# Document d'analyse des enregistrements vidéos

Cette annexe donne un aperçu du travail réalisé à partir des enregistrements vidéo.

Nous avons collecté 155 sessions vidéo de 45 minutes d'activité de programmation chacune. Il faut y ajouter entre 15 et 30 minutes d'entretien après chaque session de programmation.

Nous avons découpés les vidéos de ces sessions par puzzles, en ne gardant que les puzzles étudiés dans notre recherche doctorale, ce qui représente 259 extraits pour l'édition 2021 du concours Algoréa, 492 extraits pour l'édition 2022 et 192 extraits pour l'expérimentation complémentaire de l'été 2022.

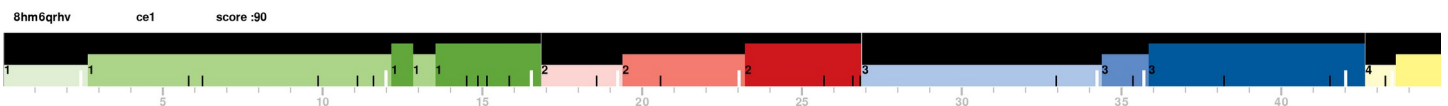
Nous avons ensuite associé ces extraits vidéos à des visualisations générées de manière automatisée : frises de sessions, suite des programmes exécutés.

Nous avons ensuite transcrit et décrit une partie de ces extraits vidéos, en fonction du profil de confrontation calculé. Les documents originaux de transcription sont classés par puzzles et représentent en tout 543 pages.

Nous regroupons dans cette annexe le travail d'analyse préalable ainsi que les liens vers les extraits vidéo mentionnés dans le document de thèse. Ceux-ci sont classés chronologiquement pour chaque sujet.

## Tom, CE1 en 2021

### Tom, 2021 tour 1



### Tom, 2021t1p1v1 - profil indéfini

[lien vers l'extrait vidéo](#)

### Tom, 2021t1p1v2 - profil accommodation

[lien vers l'extrait vidéo](#)

8hm6qrhv ce1

score global : 90

temps moyen par bloc : 26.2

1 - Planter des fleurs [white 2021]

easy

durée : 9'23

ind. rapidité normalisée : 2.07

0 - 2'39

1 - 5'50  
3'11

2 - 6'16  
0'26

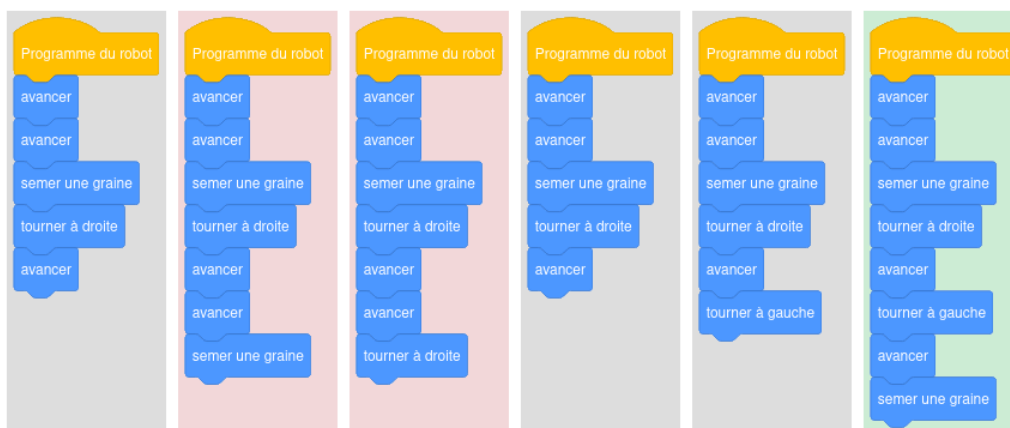
3 - 9'53  
3'37

4 - 11'07  
1'14

5 - 11'37  
0'30

6 - 12'02  
0'25

7 - 12'50



id sol :

6436

6437

6438

6436

6439

3649



## Tom, 2021t1p3v1 – profil accommodation

[lien vers l'extrait vidéo](#)

8hm6qrhv ce1

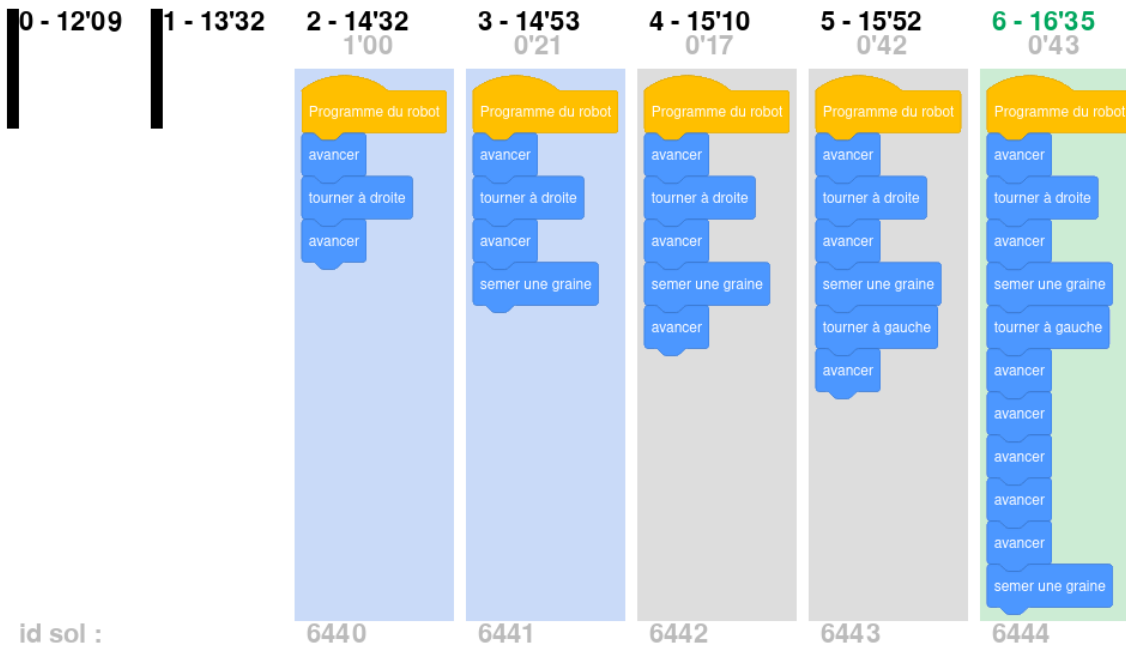
score global : 90

temps moyen par bloc : 26.2

1 - Planter des fleurs [white 2021]

medium durée : 3'44

ind. rapidité normalisée : -0.15



## Tom, 2021t1p3v2 – profil ajustement

[lien vers l'extrait vidéo](#)

8hm6qrhv ce1

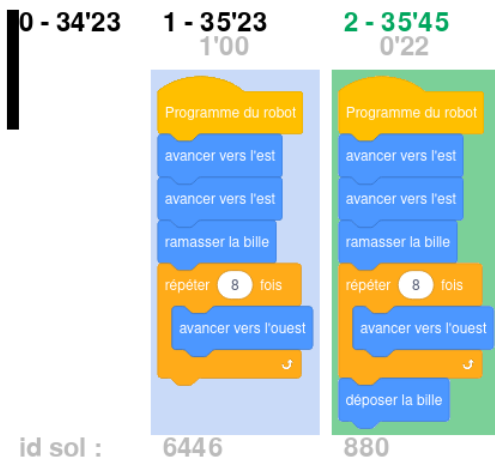
score global : 90

temps moyen par bloc : 26.2

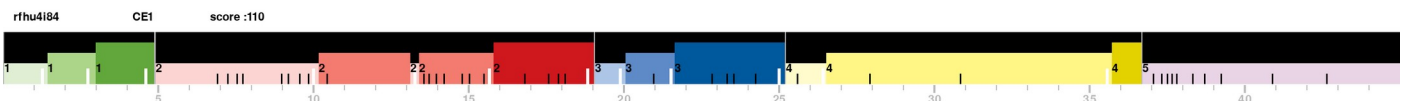
3 - Ranger les billes [white 2021] easy

durée : 1'22

ind. rapidité normalisée : -0.61



## Tom, 2021 tour 2



## Tom, 2021t2p1v3 – profil expert

[lien vers l'extrait vidéo](#)

# Tom, 2021t2p4v2 – profil accomodation

[lien vers l'extrait vidéo](#)

rfhu4i84 ce1

score global : 110

temps moyen par bloc : 14.1

4 - Jouer aux dominos [white 2021]

easy

durée : 9'06

ind. rapidité normalisée : 6.74

0 - 26'31	1 - 27'57 1'26	2 - 30'52 2'55	3 - 35'37 4'45
id sol : 25344	25370	22460	

Le sujet a ouvert le tutoriel, échangé avec l'expérimentatrice et utilisé le mode pas à pas, ce qui rend la visualisation précédente lacunaire.

Tutoriel accessible en cliquant sur le bouton Plus de détails


1 - On reprend le même type de programme que pour la version 1 étoile, qui fait ramasser des dominos au robot.


2 - On remarque que le même groupe de blocs est utilisé plusieurs fois.


3 - On obtient le même résultat en plaçant ce groupe de blocs à l'intérieur d'un bloc

Utiliser le bloc vous permet d'économiser des blocs !

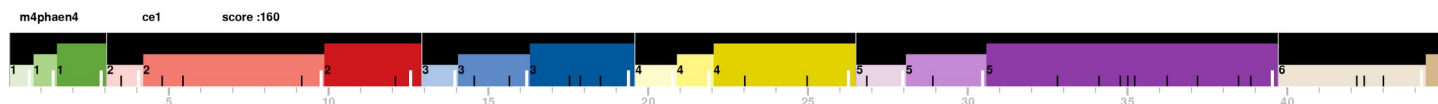
Maintenant, le robot doit ramasser 11 dominos. À vous de jouer !

0'00		<p>Il commence par compter à haute voix jusque 11 sans pointage avec la souris  <b>S - 11 dominos ! Mais comment je fais pour ramasser 11 dominos ? »</b>      Il place des blocs dans l'éditeur, compte 1 lorsqu'il place le premier bloc <i>ramasser le domino</i>, ensuite ne dit plus rien. Il s'arrête quand il a placé 8 blocs et recompte en chuchotant les blocs <i>ramasser le domino</i> en les pointant avec sa souris.      Il ajoute encore un bloc avancer et un bloc ramasser le domino.</p>	<p>Pointe-t-il directement sur l'écran avec son doigt ?</p> <p>La remarque à propos du nombre 11 indique que le sujet sent la rupture au niveau de la situation.</p> <p>Mais il mobilise le même schème que pour la version précédente :</p> <p>Prendre l'information sur le nombre d'éléments en les comptant sur la grille, placer les blocs dans l'éditeur en se repérant au bloc d'action saillante.</p> <p>Une règle de contrôle apparaît : recompter les blocs d'action saillante déjà placés dans l'éditeur</p> <p>Le sujet reprend la même stratégie que lors de la version précédente.</p>
1'00		<p>À ce moment, le feedback <i>0 blocs restants sur 10 autorisés</i> apparaît. Le sujet recompte les blocs ramasser le domino de son programme, toujours en chuchotant et en pointant avec la souris, puis reste perplexe :  <b>S - Maintenant comment je fais, j'ai pas tout ramassé. »</b></p>	<p>L'artifice didactique joue son rôle : il met en défaut le schème mobilisé par le sujet. Celui-ci voit la nécessité de changer de stratégie pour aboutir.</p> <p>Il est à noter que le sujet n'a pas encore lancé d'exécution à ce moment, il anticipe, a une représentation de l'exécution de son programme.</p>
1'19		<p>Le sujet lance une exécution de son programme.  <b>E – Oui, tu n'as pas tout ramassé et tu n'as plus assez de blocs, t'as utilisé tous les blocs... Alors, regarde tu as un bouton « Plus de détails » pour t'aider</b></p>	<p>Le sujet (CE1) n'est pas encore autonome avec les fonctionnalités de la plateforme, il a besoin d'être guidé.</p>
1'31		<p>Le sujet clique sur le bouton <b>Plus de détails</b>, ce qui ouvre le tutoriel.  <b>S – Ah avancer, ramasser le domino, avancer, ramasser le domino, oui 2 avancer ramasser le domino avancer.. mais j'comprends pas vraiment beaucoup.</b>  <b>E – Est-ce que tu as lu ?</b>  <b>S – Ah oui là ici.. utiliser le bloc.. le robot doit ramasser.. accompagne la lecture des deux dernières phrases du tutoriel d'un mouvement de souris qui suit le texte lu.</b>  <b>E – Regarde en haut, je ne t'ai pas entendu lire :1 on reprend.. non juste à droite (la souris était sur le texte de l'énoncé) 1 on reprend le même type de programme que pour la version 1 étoile, qui fait ramasser des dominos au robot, on remarque que le même.. 2 on remarque que le même groupe de blocs est utilisé plusieurs fois</b>  <b>S – 3 on obtient le même résultat en plaçant ce groupe de blocs à l'intérieur d'un bloc..</b>  <b>E – répéter</b></p>	<p>Au niveau ergonomique, l'oeil est attiré tout de suite par les images, ce qui entraîne que les deux premières lignes ne sont pas lues.      → inverser, mettre les images en premier.      → ou grouper l'image avec le texte qui lui correspond</p>

2'34		<p><b>S – Alors..</b> Le sujet ferme le tutoriel, et vide l'éditeur (en deux temps). Il place un bloc répéter dans l'éditeur et renseigne correctement le compteur de boucle à 11. Que signifie ce 11 pour lui ? Il place ensuite une séquence de blocs à l'intérieur du bloc répéter jusqu'à ne plus avoir de blocs disponibles (feedback du système), puis lance une exécution. Comme il n'y a pas un nombre entier de motifs dans la boucle, deux blocs ne sont pas ramassés par le robot, ce qui laisse le sujet dépité. <b>S – Mais j'ai pas pu tout faire..</b></p>	<p>Le sujet reprend en fait le même schème, mais en plaçant sa séquence à l'intérieur du bloc répéter.</p> <p>Le sujet n'abandonne pas la représentation analogique. Les deux représentations du nombre de motifs, analogique et symbolique coexistent.</p>
4'24		<p><b>E – Alors tu vois, tu as une petite croix marron.. sur la bande marron en bas</b> <b>S – Oui</b> <b>E – Appuie sur cette petite croix.. Voilà.. Et à côté du bouton avec le triangle, tu en as un autre, avec un petit triangle et un trait.. Voilà celui-ci.. Celui-ci tu te rappelles, tu t'en étais déjà servi la dernière fois, c'est le mode pas à pas.. à chaque fois que tu cliques dessus, il exécute une instruction.. donc il faut cliquer plusieurs fois de suite dessus, pour bien voir ce que fait le robot.</b> Le sujet clique plusieurs fois sur le mode pas à pas. <b>E – Voilà.. En bleu foncé, tu as la prochaine instruction que le robot va exécuter.</b> Le sujet exécute le premier passage dans la boucle et s'arrête plusieurs secondes avant la première instruction lors du deuxième passage, celle qui pose problème. Il continue encore quelques clics et s'arrête à nouveau <b>S – Je ne sais plus quoi faire maintenant</b></p>	<p>Le temps d'arrêt avant d'entrer pour la deuxième fois dans la boucle indique que le sujet perçoit qu'à cet endroit il y a un problème.</p> <p>Mais, il y a blocage. Le sujet n'a pas de stratégie de recherche autonome, il demande indirectement l'étayage de l'adulte.</p>
5'45		<p><b>E – Alors retourne voir le bouton Plus de détails, y'a peut-être quelque chose que tu n'as pas bien regardé.</b> <b>S – Il est très dur celui-là..</b> <b>E – Ah oui il est plus dur .. regarde là le bloc orange avec le numéro 3.. Combien de dominos il va ramasser ? Là sur l'image ?</b> <b>S – 11 !</b> <b>E – Là maintenant tu dois en ramasser 11, mais dans le programme.. dans le programme qui est dans l'exemple.. qui est entouré ?</b> <b>S – Oui ?</b> <b>E – Combien de dominos le robot va ramasser ?</b> <b>Le sujet compte tous les blocs ramasser du tutoriel, sans tenir compte de la numérotation !</b> <b>S – 7</b> <b>E – 7 ?</b> <b>S – Oui</b> <b>E – Ah oui, alors je vois ce que tu n'as pas compris.. en fait c'est le même programme, mais qu'on a écrit différemment. C'est le même programme en 1, en 2 ou en 3.. mais en 1.. regarde tous les blocs sont collés : avancer, ramasser, avancer, ramasser, avancer, ramasser.. en 2, on a séparé les blocs pour montrer que c'était pareil.. et en 3, on a mis un morceau qui était pareil à l'intérieur du bloc répéter, et, on a dit ce morceau là de programme, il faut le répéter 3 fois.. pour ramasser 3 dominos.. Donc à chaque fois qu'il y a dans le bloc répéter une instruction.. à chaque fois les instructions du bloc répéter vont s'exécuter les unes après les autres.. donc la première fois, ça va faire avancer, ramasser un domino.. la deuxième fois qu'on va répéter ça va encore avancer, ramasser un domino.. et la troisième fois encore</b></p>	<p>Il ne fait pas sens pour le sujet que les images du tutoriel sont des images séquentielles. Ils les voient comme une même image ce qui entraîne un blocage. → mettre une séparation entre les images → revenir à un tutoriel vidéo : certes le rythme est imposé, raison pour laquelle on a changé, mais le caractère séquentiel est pris en charge par le flux de la vidéo.</p>

		<p><i>avancer et ramasser un domino. Si je veux ramasser un quatrième domino, qu'est-ce que je fais ?</i></p> <p><i>S – Je mets quatre fois</i></p> <p><i>E – Voilà.. Et si je veux en ramasser 11 ?</i></p> <p><i>S – Je mets 11 fois ?.. Ah oui j'ai compris</i></p>	
8'13		<p>Le sujet ferme le tutoriel et vide l'éditeur.</p> <p>Il conçoit le programme attendu.</p> <p><b>S - Voilà !</b></p> <p>Il exécute le programme et valide</p>	<p>Le fait qu'il exprime son contentement avant d'exécuter le programme prouve qu'il anticipe le résultat de l'exécution.</p> <p>Il sera intéressant de regarder si la stratégie de ce sujet évolue sur le même type de problème par la suite.</p>

## Tom, 2021 tour 3



### Tom, 2021t3p4v3 – profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

m4phaen4 ce1

score global : 160

temps moyen par bloc : 12.2

4 - Rejoindre la fusée [white 2021]

medium durée : 4'16

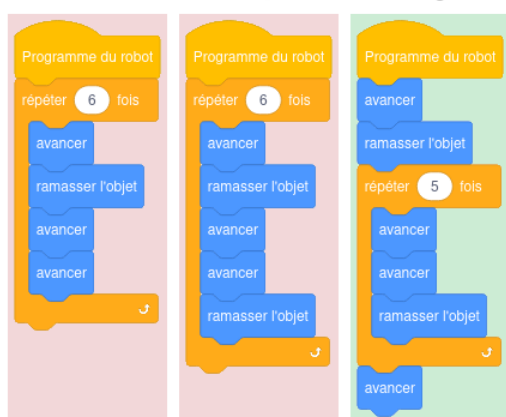
ind. rapidité normalisée : 2.5

0 - 22'03

1 - 23'03  
1'00

2 - 25'00  
1'57

3 - 26'19  
1'19





id sol :

16270

9924

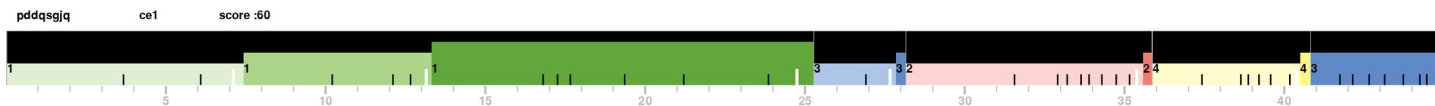
529

Capture	Description	Interprétation
	<p><b>2021t3s4v3_m4phaen4_CE1.mp4</b></p> <p>0'00 S – <i>Ah oui d'accord..</i></p> <p>(2) il place un bloc répéter dans l'éditeur</p> <p>(3v) S – <i>avancer, ramasser l'objet, avancer</i> (en plaçant les blocs correspondant dans le bloc répéter).</p> <p>Il s'arrête quelques secondes, reste silencieux</p>	<p>À ce moment le motif est correct mais le sujet ne l'évalue pas comme tel.</p>
	<p>0'27 (3a) Il place un bloc avancer supplémentaire dans le bloc répéter</p> <p>(1v) Il compte à mi-voix jusque 6, la souris restant immo-</p>	<p>Dès ce moment, le sujet a bien repéré l'unité visuelle, l'objet. C'est le remplacement du robot pour aborder l'itération suivante qui pose problème.</p>

	<p>bile</p> <p>(4) Il renseigne la valeur 6 dans le compteur de boucle.</p> <p><b>S – <i>J’suis pas sûr.. pas sûr</i></b></p> <p>(5a) Soumission 1 → le robot ramasse le premier objet et s’arrête sur la case suivant le deuxième. Le message « Il n’y a rien à ramasser ! » est affiché</p>	
	<p>1’00 (X) Le sujet vide l’éditeur.</p> <p><b>S – (9v) <i>Avancer, ramasser l’objet</i></b> (en accrochant les blocs correspondants au programme du robot).</p> <p>(2) Il accroche un bloc répéter en dessous de ces deux blocs,</p> <p>Il tente de le déplacer pour qu’il englobe les deux blocs, il n’y arrive pas, et (X) vide de nouveau l’éditeur.</p>	<p>Problème de dextérité avec la souris</p> <p>Le sujet se repère aux objets. Il traite le premier objet afin de se trouver dessus pour continuer.</p>
	<p>1’44 (2) Il accroche un bloc répéter dans l’éditeur</p> <p>(3v) <b>S – <i>Avancer</i></b> (place un bloc répéter dedans)</p> <p>Il continue en plaçant les blocs ramasser l’objet et avancer. Il veut ajouter un bloc avancer supplémentaire mais ne le lâche pas à l’endroit voulu. Il enlève alors les 3 derniers blocs.</p> <p><b>S – <i>Ah, ça veut plus</i></b></p> <p>(3v) <b>S – <i>Avancer, ramasser l’objet</i></b> (place les blocs avancer, ramasser l’objet, avancer, avancer, ramasser l’objet, en finissant en chuchotant)</p> <p>(1v) La souris est immobile et il compte à mi-voix jusque 6.</p> <p>(4) Il renseigne la valeur 6 dans le compteur de boucle.</p> <p><b>S – <i>Je sais pas si ça va marcher</i></b></p> <p>(5a) Soumission 2 → le robot ramasse les deux premiers objets et s’arrête sur la case suivante. Le message « Il n’y a rien à ramasser ! » est affiché</p>	<p>Le sujet allonge le motif qui devient trop grand. Pour dénombrer les motifs, il ne tient pas compte du fait qu’il ramasse deux objets à chaque itération.</p> <p>Les éléments comptés sur la grille et le motif dans le programme ne sont plus en correspondance.</p>
	<p>2’57 (X) Il vide l’éditeur.</p> <p><b>S – (9v) <i>Je vais tout recommencer.. Avancer, ramasser l’objet</i></b> (accroche ces deux blocs dans l’éditeur)</p> <p>(2) Il accroche un bloc répéter en-dessous de ces deux blocs</p> <p>(3v) <b>S – <i>Avancer, avancer, ramasser l’objet</i></b></p> <p>(1) La souris reste immobile et il y a un temps de silence</p> <p>(4) Il renseigne la valeur du compteur de boucle à 5</p> <p>(6) Il complète son programme par un bloc avancer après la boucle.</p> <p>(5a) soumission 3 valide</p>	<p>Le sujet se repère aux objets. Il traite le premier objet afin de se placer dessus en entrée de boucle. Il contrôle la position du robot avant la boucle.</p> <p>Il renseigne bien le compteur de boucle, signe que la procédure est maîtrisée.</p> <p>Il distingue bien les instructions qui sont dans et hors de la boucle.</p> <p>Il ne se sert pas du mode pas à pas. Sa stratégie de débogage est sommaire : il recommence.</p>

# Paola, CE1 en 2021, CE2 en 2022

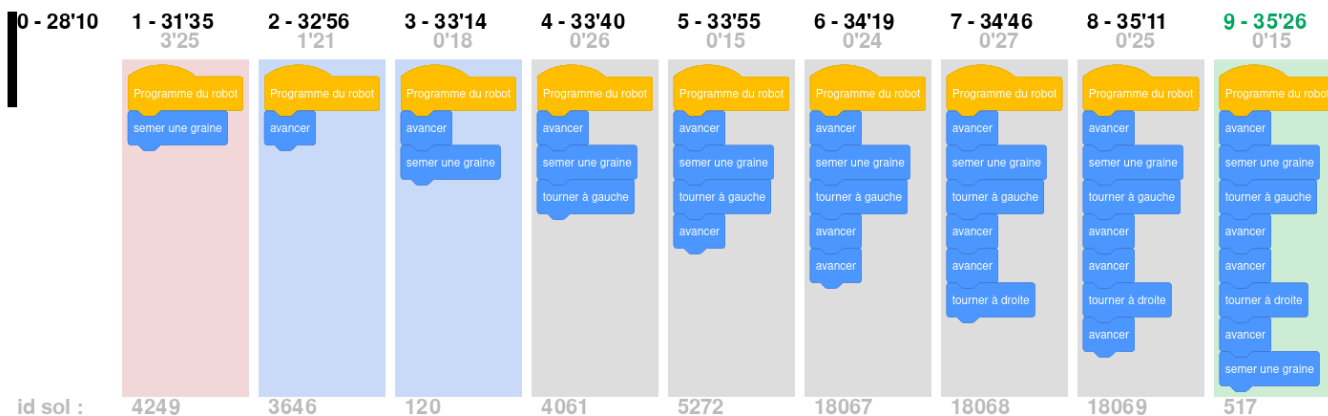
## Paola, 2021 tour 3



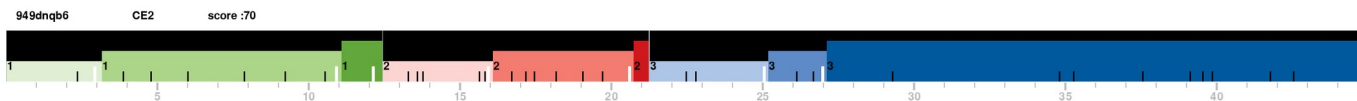
### Paola, 2021t3p2v1 – profil accommodation

[lien vers l'extrait vidéo](#)

pddqsgjq ce1 score global : 60 temps moyen par bloc : 39.0  
2 - Planter des fleurs [white 2021] basic durée : 7'16 ind. rapidité normalisée : 0.6



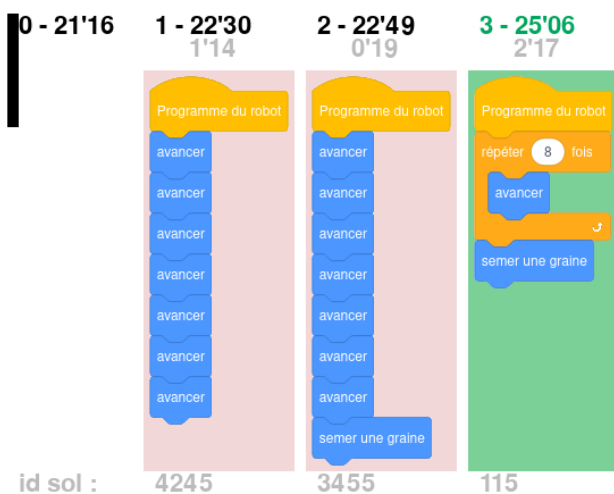
## Paola, 2022 tour 1



### Paola, 2022t1p3v1 – profil accommodation

[lien vers l'extrait vidéo](#)

949dnqb6 ce2 score global : 70 temps moyen par bloc : 10.6  
3 - Planter des fleurs [2022-1 white] basic durée : 3'50 ind. rapidité normalisée : 3.34



C'est la deuxième session pour Paola, élève de CE2, qui a commencé l'expérimentation lors du tour 3 de 2021.

L'extrait débute par un temps de silence de 20'' qui correspond très probablement à la lecture de l'énoncé, suivi de l'ouverture spontanée du tutoriel.



Lorsqu'elle referme le tutoriel, Paola interpelle l'expérimentatrice

**Paola – J'comprends pas trop trop**

**E – Alors..**

L'expérimentatrice s'interrompt car Paola commence un programme en séquence. Elle place 7 blocs avancer sans survol de la grille puis lance l'exécution du programme (id 4039).

Bien que le robot ne soit pas arrivé sur la case avec de la terre, elle ajoute un bloc semer une graine. A-t-elle anticipé la contrainte en nombre de blocs ? ou pense-t-elle que le robot sème sur la case devant lui.

**Paola – Ah d'accord**

**E - Mm**

Étant donné sa remarque au résultat de l'exécution suivie d'un ajout de bloc avancer, il est très probable que ce soit la deuxième option qui soit la bonne : la représentation de l'action engendrée par le bloc semer une graine était erroné.

Paola est alors confrontée à la fenêtre pop-up qui lui indique que la limite en nombre de blocs est dépassée.

**E – Alors regarde bien.. regarde bien**

**Paola – J'ai trop de blocs**

**E – Voilà..** (Paola décroche la séquence de blocs) **Non non non laisse pour l'instant.. y'a trop de blocs.. mais quand tu cliques sur Plus de détails on t'explique comment économiser des blocs**

Paola ouvre Plus de détails et reste silencieuse, la souris ne bougeant pas, puis referme

**E – Est-ce que tu as compris ?**

Paola supprime les deux derniers blocs de la séquence, puis **elle essaie de placer un bloc avancer dans le bloc répéter de la réserve.**

**Paola – Euh non**

**E – Alors qu'est-ce que tu penses qu'il faut faire ?**

**Paola – faut mettre un bloc là (pointe le bloc répéter)**

**E – Oui.. alors tu peux le prendre celui-là**

Paola essaie à nouveau de placer un bloc avancer dans le bloc répéter de la réserve.

**E – D'accord.. je vois ce que tu n'as pas compris.. tu voudrais placer un bloc avancer dans ce bloc orange là.. qui s'appelle répéter.. et le bloc avancer il disparaît quand tu le mets.. c'est ça qui t'embête ?**

**Paola – Oui**

**E – Oui.. alors c'est parce qu'il faut d'abord déplacer le bloc orange dans la partie où tu as le droit..** (Paola place un bloc répéter dans l'éditeur) **voilà.. et là maintenant tu as le droit de mettre des choses dedans**

Paola place un bloc avancer dans le bloc répéter mais elle n'enlève pas les blocs avancer déjà présents.





**Paola – Je peux changer le nombre ?**

**E - Oui**

Paola clique sur le nombre 10, le clavier virtuel apparaît, elle clique sur 8 qui est effectivement le nombre de déplacements à faire réaliser au robot, et enregistre.

**Paola – Et du coup j'enlève tout ça ?** (pointe les blocs avancer avant la boucle)

**E - Oui**

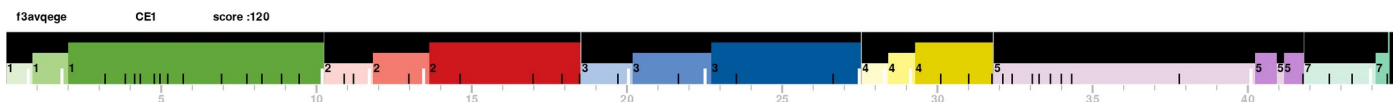
Par manque de dextérité, elle vide l'éditeur

**Paola – Ah**

Elle refait le programme correct rapidement et valide.

## Léa, CE1 en 2021, CE2 en 2022

### Léa, 2021 tour 1



### Léa, 2021t1p4v3 – profil échec

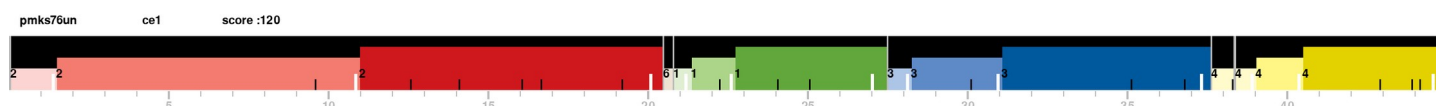
[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

f3avqege	CE1	score global : 120
problème 4	medium	durée : 2'30
0 - 29'17	1 - 30'08	2 - 31'02    3 - 31'47
id sol : 6	100	5

Description	Interprétation
<p><b>Tour 1 version medium : 2021t1s4v3_f3avqege_CE1.mp4</b></p> <p>(2) Le sujet accroche un bloc répéter dans l'éditeur,</p> <p>(4) renseigne le compteur de boucle avec la valeur 2, et</p> <p>(3) place un bloc avancer et un bloc semer une graine dans le bloc répéter.</p> <p>(5a) Il lance une exécution (11aX) (soumission 1).</p>	<p>Le sujet mobilise la même stratégie que lors des versions précédentes : compter les zones de terre et utiliser une boucle pour semer une graine sur chacune d'elle.</p> <p>Il fait bien la différence entre les instructions dans et hors de la boucle, le bloc avancer est placé sans hésitation hors de la boucle.</p> <p>La verbalisation indique que le motif est identifié, que l'unité est créée.</p> <p>Mais le sujet n'accède pas à la représentation de cette unité en utilisant le bloc répéter. Il est probable que le bloc répéter déjà utilisé dans le programme constitue un obstacle, car il est nécessaire de placer celui-ci à l'intérieur d'un autre, c'est à dire à imbriquer deux boucles.</p> <p>Il s'agit d'une extension du schème de répétition au cas où le motif comporte lui-même une répétition. En l'état actuel, le sujet applique le théorème-en-acte « Le motif est constitué d'une séquence »</p>
<p>(7g) Il parcourt la grille avec le pad,</p> <p>(6) puis place un bloc avancer sous la boucle et dit :</p> <p><i>S – Du coup je fais ça plusieurs fois</i></p> <p>Il continue avec un bloc (2s) répéter (4s) 2 fois (3s) (avancer, semer une graine), (6) un bloc avancer en-dessous, (2s) un bloc répéter (4s) 2 fois qu'il laisse vide.</p> <p>(5a) Il marque un temps d'arrêt et lance une exécution (soumission 2)</p>	
<p>(3s) Il déplace un bloc avancer dans l'éditeur, ce qui provoque l'affichage du message d'alerte sur le nombre de blocs disponibles. (0bloc)</p> <p>(X) Après avoir fermé le pop-up, le sujet vide l'éditeur et retourne au menu</p> <p>(visiblement, la soumission 3 est un bug d'enregistrement)</p>	

### Léa, 2021 tour 3



### Léa, 2021t3p4v3 – profil accommodation

[lien vers l'extrait vidéo](#)

0 - 40'30

1 - 42'56  
2'26

2 - 43'56  
1'00

3 - 44'11  
0'15

4 - 44'37  
0'26

id sol :


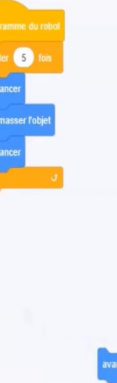

18388

9586

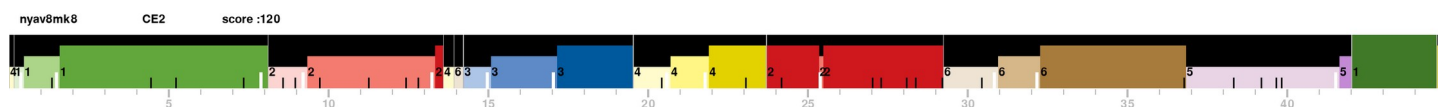
18389

18390

	Description	Interprétation
	<p><b>Tour 3 version medium :</b> <b>2021t3s4v3_pmks76un_CE1.mp4</b></p> <p>0'00 La souris parcourt longuement la grille, en pointant certaines cases : case du robot, mais surtout cases contenant des objets.</p> <p>Détail :</p> <p>(7pg) Il pointe a case du robot puis les 3 premiers objets.</p> <p>Il recommence : pointe le robot puis les 2 premiers objets.</p> <p>Il recommence : pointe les 3 premiers objets.</p> <p>Il recommence : pointe le robot, puis les 2 premiers objets, puis les 2 cases vides suivantes.</p> <p>Il recommence : pointe le robot puis les 4 premiers objets, puis revient sur le 3ème objet.</p> <p>(1) Il recommence : pointe le robot puis les 6 objets</p>	<p>Cette phase correspond à une prise d'information à la fois pour l'identification du motif et pour le dénombrement.</p> <p>Comme le sujet pointe d'abord le robot, on peut supposer qu'il se représente mentalement les actions du robot.</p> <p>Il cherche à prendre un repère pour construire le motif.</p>
	<p>1'13 (4) Il ouvre le clavier numérique après plusieurs clics parasites. La souris s'arrête sur les touches 5 et 6, puis il entre la valeur 5.</p> <p>(2) Il accroche le bloc répéter dans l'éditeur.</p> <p>(7pg) Il pointe le dernier objet</p> <p>(7pr) Il place sa souris sur le bloc avancer, qui reste immobile pendant quelques secondes à cet endroit.</p> <p>(3) Il place les blocs avancer et ramasser l'objet dans le bloc répéter.</p> <p>(6) Il accroche un bloc avancer en-dessous du bloc répé-</p>	<p>L'hésitation au moment d'entrer la valeur indique que le sujet ne se rappelle plus bien le résultat de son dénombrement, ce qui finalement aboutit à une erreur.</p> <p>On peut supposer qu'au cours de la phase précédente, le sujet a essayé de traiter à la fois le dénombrement et l'identification du motif, ce qui engendre une surcharge cognitive.</p> <p>Le pointage du dernier objet et le fait que le sujet ajoute un bloc avancer après la boucle indique qu'il se représente le robot sur la case du dernier objet</p>

	<p>ter.</p> <p>(5a) Soumission 1 → Le premier objet a disparu, le robot est placé sur la case suivante, et le message « Il n’y a rien à ramasser ! » est affiché dans la zone de feedback.</p>	<p>après le dernier passage dans la boucle.</p> <p>C’est cohérent avec le motif placé dans le bloc répéter qui finit par le bloc ramasser l’objet.</p> <p>Le repère pour l’identification du motif sur la grille est l’objet. Dans le programme, c’est le bloc ramasser l’objet placé en dernière position.</p> <p>Le remplacement du robot pour aborder l’itération suivante n’est pas gérée lors de ce premier essai.</p>
	<p>2’34 (7pe) Il pointe le bloc ramasser l’objet de son programme, puis le bloc avancer après la boucle.</p> <p>(6X) Il détache ce bloc avancer et le place en bas de l’éditeur, puis (6) le remet à sa place initiale.</p> <p>(3) Il ajoute un bloc avancer dans le bloc répéter après le bloc ramasser l’objet.</p> <p>(5a) Soumission 2 → le robot ramasse les 5 premiers objets et s’arrête sur la case du dernier objet. Le message « Votre robot doit rejoindre la fusée est renvoyé »</p>	<p>Suite au feedback visuel, le sujet ajoute le remplacement adéquat du robot.</p> <p>Mais il ne prend pas en compte la conséquence de cet ajout à la position du robot après le dernier passage dans la boucle.</p>
	<p>3’26 (6X) Il détache à nouveau le bloc avancer qui est hors de la boucle, et le laisse en bas de l’éditeur.</p> <p>(5a) Soumission 3 → le robot ramasse les 5 premiers objets et s’arrête sur la case vide suivante. Le message « Votre robot doit rejoindre la fusée est renvoyé »</p>	<p>L’erreur de dénombrement complique la prise de conscience de l’erreur.</p> <p>Le sujet ne remet pas en cause son dénombrement.</p> <p>Est-ce que cette exécution est une prise d’information de la position du robot après la boucle ?</p> <p>Le fait que le sujet mette de côté le bloc avancer sans le supprimer amène cette hypothèse</p>
	<p>3’42 (6) Le sujet ajoute la séquence avancer, ramasser l’objet, avancer en-dessous de la boucle.</p> <p>(5a) Soumission 4 valide</p>	<p>Complétion du programme à partir du feedback visuel : le sujet repart de la position du robot à la fin de l’exécution et ajoute les blocs pour résoudre la sous-tâche engendrée.</p> <p>→ Débogage rustine</p>

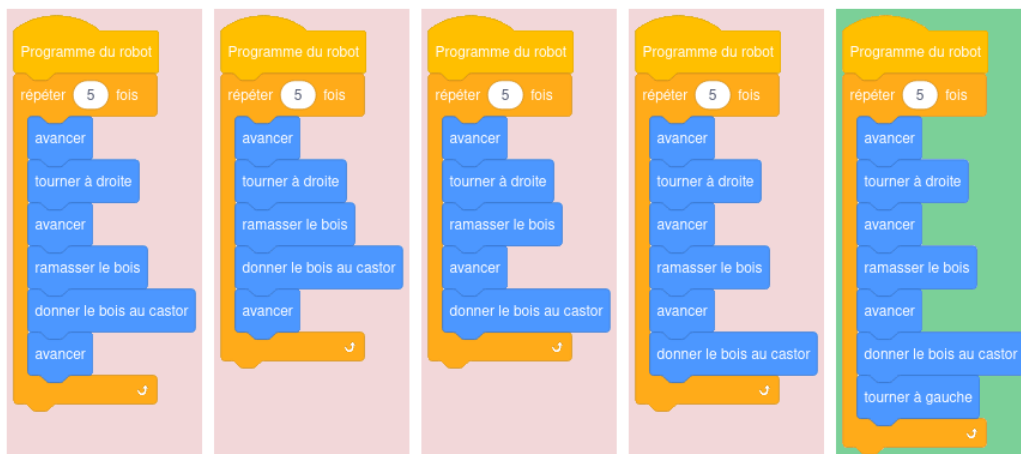
## Léa, 2022 tour 2



### Léa, 2022t2p5v1 – profil accommodation

[lien vers l’extrait vidéo](#)

0 - 36'50

1 - 38'21  
1'312 - 39'13  
0'523 - 39'41  
0'284 - 39'51  
0'105 - 41'34  
1'43

id sol :

12214

12215

2838

232

233

Léa fait la session sur tablette, nous ne disposons pas des mouvements de souris pour cette confrontation.

Léa – *Ouh.. à l'aide..*

E – *Mm*

Léa – *Ah je crois c'est pas si difficile en fait*

ajREP

E – *Non j'crois pas*

Léa compte tout haut jusque 5

Léa – *Ok répéter.. cinq (comp 5) en fait c'est .. quand on le voit le parcours.. quand on l'ouvre.. on croit que c'est difficile.. mais en fait c'est pas très.. difficile*

E – *Mm*

Léa – *C'est cinq fois..*

E – *Alors pourquoi au début tu crois que c'est difficile et après non ?*

repAV

Léa – *Et ben parce que quand on voit tous les trucs là.. tous les castors.. tous les bois.. et tout fff..*

E – *Et à quel moment tu te dis que finalement ce n'est pas très difficile ?*

Léa – *Ben quand j'ai trouvé comment on fait..*

E – *Et alors comment on fait ?*

Léa – *Ben là il suffit de.. de répéter plusieurs marches*

E – *Mm.. bah oui*

Léa – *Avancer.. tourner.. ben mon robot il est [incompréhensible] ce sera plus pratique.. tourne à droite (repTD) avancer tourner à droite.. avancer (chuchoté, repAV) ramasser le bois (repRAM) avancer (repAV) et donner le bois au castor.. là (repDEP mais ne le place pas au bon endroit à cause d'une erreur de manipulation)*

Le manque de dextérité peut générer des erreurs qui interrompent un raisonnement adéquat et compliquent le débogage. Ici, l'erreur de manipulation fait perdre le fil de la simulation mentale.

Exécution du programme d'id.12214

Léa – *Mais hé attends.. j'y comprends rien*

exécution en mode pp

Léa – *Répéter cinq fois (pp) avancer (pp) tourner à droite (pp, pp) ah d'accord.. j'ai mis un truc en trop (décroche la séquence sauf le premier AV) soit avancer soit tourner à droite (enlève le AV après TD) maintenant..*

Exécution du programme d'id.12215

Détache les trois derniers blocs de la séquence, puis le TD, supprime le AV après DEP

Léa – *Ah d'accord*

E – *Sers-toi bien du mode pas à pas*

intercale un bloc AV après RAM

Léa – *Voilà je vais tout rattacher* (reconstitue la séquence)

Exécution du programme d'id.2838

Léa – *Ah oui je vais remettre avancer là* (intercale AV entre TD et RAM)

Exécution du programme d'id.232

Léa – *Mm.. mm d'accord.. répéter cinq fois (pp) répétition un.. (pp) avancer (pp) je tourne à droite (pp) j'avance (pp jusqu'au feedback ; le robot essaie de se déplacer sur un obstacle !) mais où est-ce que j'en suis là.. ah récapitulons.. (recommence l'exécution en pp) je commence.. j'avance.. je suis là.. je tourne.. attends là je fais le robot..*

E – *Oui.. mais fais le mode pas à pas aussi en même temps.. là tu as cliqué une fois sur le mode pas à pas.. et donc voilà maintenant..*

pp

Léa – *Avance.. je tourne (pp) j'avance (pp) je prends le bois (pp) j'avance (pp) je donne au castor.. ah j'ai oublié un truc que je viens de comprendre*

E – *Alors qu'est-ce que tu as oublié et que tu viens de comprendre ?*

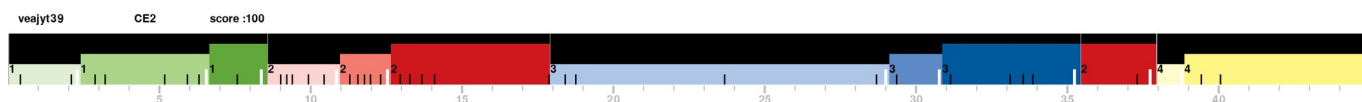
Léa – *Tourner à gauche* (repTG) *car sinon il va foncer..* j crois que c'est bon maintenant

Exécution du programme d'id.233

Léa – *Ah si.. finalement c'est pas si difficile que ça*

## Gina, CP en 2021, CE1 en 2022

### Gina, 2022 tour 1



### Gina, 2022t1p2v1 – profil accommodation

[lien vers l'extrait vidéo](#)

veajyt39 ce1

score global : 100

temps moyen par bloc : 8.2

2 - Collecter les pierres [2022-1 white] basic

durée : 2'19

ind. rapidité normalisée : 1.83

0 - 8'35

1 - 9'02  
0'27

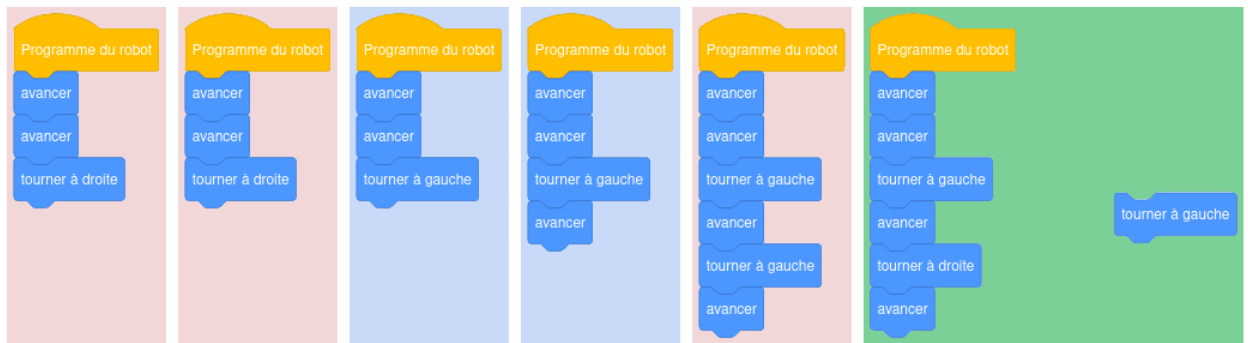
2 - 9'14  
0'12

3 - 9'25  
0'11

4 - 9'57  
0'32

5 - 10'28  
0'31

6 - 10'54  
0'26



id sol :

103

103

3233

104

2589

21687

### Gina, 2022t1p3v1 – profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

[lien vers l'extrait vidéo 3](#)

veajyt39 ce1

score global : 100

temps moyen par bloc : 8.2

3 - Planter des fleurs [2022-1 white] basic

durée : 11'08

ind. rapidité normalisée : 15.19

0 - 17'55

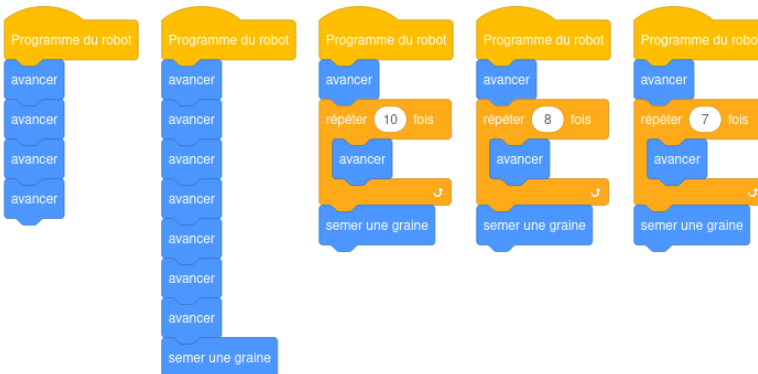
1 - 18'26  
0'31

2 - 18'47  
0'21

3 - 23'42  
4'55

4 - 28'43  
5'01

5 - 29'03  
0'20



id sol :

10050

3302

20029

20030

13533

C'est la deuxième session pour Gina, qui a commencé lors du tour 2021-3. La boucle avait été abordée lors de ce premier tour. Cela fait environ 9 mois que Gina n'a pas été confrontée avec l'environnement de programmation.

Gina arrive sur le sujet sans savoir utiliser le bloc répéter (cela a bloqué son avancée en version medium du problème 2). Elle commence son programme en séquence.

Elle soumet d'abord un programme avec 4 blocs avancer (id 10050), puis elle ajoute 3 blocs avancer et un bloc semer une graine (id 3302). Il n'y a pas d'arrêt entre le placement du dernier bloc avancer et le bloc semer une graine, pas d'hésitation qui pourrait laisser penser que Gina met un bloc de moins pour respecter la limite de blocs. La réaction de Gina au résultat de l'exécution est d'ajouter un bloc avancer, ce qui montre que l'erreur est en rapport avec le dénombrement. L'hypothèse est que Gina dénombre les cases vides au lieu de dénombrer les déplacements du robot. Nous ne pouvons pas le vérifier sur cet extrait car Gina ne pointe pas les cases avec la souris.

De ce fait, Lina se retrouve confrontée à la limite de blocs : fenêtre pop-up qui indique que le programme contient trop de blocs, et bandeau rouge empêchant l'exécution du programme.

**E – Alors qu'est-ce qui se passe ?**

**Gina – Eh ben j’ai utilisé trop de blocs**

**E – Eh bien voilà, t’as le même problème..** [que pour le puzzle précédent qui a été quitté sans être résolu] **t’as utilisé trop de blocs.. et donc qu’est-ce que tu sais qu’il faut faire.. pour pas utiliser trop de blocs.**

Gina identifie bien le problème ce qui est repris par l’expérimentatrice. Elle a identifié le bloc répéter comme pouvant solutionner son problème mais elle ne sait pas l’utiliser. Les schèmes d’usage ne sont pas construits.

**Gina – Euh répéter 10 fois** (commence à déplacer le bloc répéter)

**E – Alors maintenant regarde tu as le bouton Plus de détails.. et on va t’expliquer comment on utilise ce bloc répéter**

Gina ouvre le tutoriel et le parcourt avec sa souris, qui suit probablement la lecture ou le point d’attention.

**Gina – Ah le bloc répéter permet d’utiliser moins de blocs**

**E – Mm**

**Gina – Et comment je fais pour retourner si je veux faire l’exercice ?**

Gina a confirmation de son intuition sur la fonctionnalité du bloc et cela lui suffit à ce moment. Elle demande à retourner sur l’éditeur. On note au passage que Gina n’est pas autonome avec la navigation au sein de l’environnement de programmation. Elle n’a probablement pas construit le concept de fenêtre et est perdue lorsque tout n’est pas sur la même fenêtre.

**E – Alors tu cliques sur Masquer les détails**

De retour sur l’exercice, Gina essaie de combiner les blocs répéter et avancer mais en restant dans la réserve. Elle n’a pas construit le schème d’usage de l’interface de programmation : **« La construction du programme se fait dans la zone d’édition »**.

[observer s’il y a déjà eu des tentatives en ce sens auparavant]

**E – Alors si tu veux l’utiliser il faut le mettre dans la partie blanche le bloc répéter.. il faut que tu le déplaces.. dans la partie blanche.. l’accrocher ou pas.. et là maintenant tu peux faire ce que tu veux avec.**

**Gina – Et comment je fais pour effacer la bande rouge ici ?**

**E – Alors pour effacer la bande rouge il faut que tu enlèves des blocs**

Gina vide l’éditeur.

Elle amène le bloc répéter dans l’éditeur puis se reprend et accroche d’abord un bloc avancer, le bloc répéter, puis le bloc semer une graine en-dessous.

La non construction des schèmes d’usage de l’environnement de programmation a coupé et perturbé la prise en main du bloc répéter : alors qu’elle essayait de combiner les blocs répéter et avancer dans la réserve, elle laisse le bloc répéter vide une fois celui-ci placé dans l’éditeur.

Elle exécute le programme en mode pas à pas (soumission non enregistrée).

**Gina – Euh j’ai pas trop compris** (pointe le pop-up avec « répétition 3 », la i-ème itération en cours).

L’expérimentatrice ne voit pas l’attention portée à ce pop-up et questionne Gina sur un autre aspect : sa représentation de l’exécution.

**E – Alors qu’est-ce que tu penses qu’il va faire ton programme ? Qu’est-ce que tu aurais voulu qu’il fasse ?**

**Gina – Ben qu’il fasse avancer le robot**

Le but de Gina est explicité nettement. L’action est finalisée.

**E – Oui qu’il fasse avancer le robot. Et ton bloc répéter il fait avancer le robot ? Tu penses qu’il fait avancer le robot ton bloc répéter ?**

**Gina – Euh non**



*E – Pourquoi ?*

*Gina – Je sais pas trop*

*E – Alors regarde bien le Plus de détails encore.. Essaie de m’expliquer comment il fonctionne ce bloc répéter quand tu vois le dessin.*

*Gina – Il permet d’utiliser 8 fois le même bloc par exemple*

*E – Alors si tu utilises 8 fois le même bloc.. ça c’est le dessin... alors on a enlevé des blocs parce que c’était trop long.. et alors ça ça veut dire quoi quand je mets le bloc répéter juste à côté ?*

*Gina – Mm bah ça fait rien*

*E – Ça fait rien*

Le tutoriel ne fait pas sens. Gina le referme.

Gina ferme le tutoriel et ajoute un bloc avancer dans le bloc répéter. Malgré qu’elle n’ait visiblement pas compris le schéma, elle a repéré qu’il y avait un bloc à l’intérieur du bloc répéter et reproduit la disposition dans son programme. Mais le bloc avancer déjà là avant la boucle n’est pas remis en cause.

Gina exécute son programme en mode pas à pas jusqu’à amener le robot sur la case juste avant la terre, elle s’arrête (donc soumission non enregistrée) et pointe à nouveau la fenêtre pop-up marqueur d’itérations. A-t-elle compris que le nombre s’incrémente de 1 à la fois et que le robot allait dépasser la zone de terre ?

*Gina – Maintenant je fais quoi ?*

Elle décroche le bloc semer une graine, le fait survoler le programme avant de le remettre à la même place. Ensuite la souris erre sur l’interface, ce qui pousse l’expérimentatrice à intervenir.

*E – Quand tu as exécuté avec le mode pas à pas, qu’est-ce qui s’est passé ?*

*Gina – Eh ben j’pouvais plus mettre semer une graine*

*E – Pourquoi ?*

*Gina – Et bien parce qu’après ça recommence à chaque fois.. je dois avancer 10 fois*

*E – Mm*

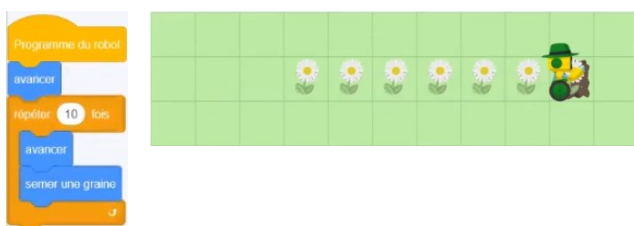
Le sujet signale qu’elle ne peut pas semer une graine sur la zone de terre parce que le robot va aller trop loin. **Le schème d’usage pour changer le nombre de répétitions n’est pas disponible** et elle pense qu’il faut utiliser le bloc répéter avec la valeur 10 par défaut, ce qui empêche d’aboutir.

Le théorème-en-acte appliqué est « **Un bloc pris dans la réserve n’est pas modifiable** »

Gina relance une exécution en mode pas à pas pour montrer. L’expérimentatrice l’incite à poursuivre l’exécution au-delà de la zone de terre jusqu’à ce que le message d’erreur apparaisse (solution 20229).

*E – Encore.. vas-y.. encore.. encore*

Après la fin de l’exécution, Gina place le bloc semer une graine à l’intérieur de la boucle, après le bloc avancer et lance une exécution en mode pas à pas jusqu’à ce que le robot ait semé une graine sur la zone de terre (non enregistrée).



*E – Alors comment il fonctionne ce bloc répéter ?*

*Gina – Euh à chaque fois que j’avance je plante une graine*

*E – Bah oui là à chaque fois que tu avances tu plantes une graine*

*Gina – Bah ça fait joli*

*E – C'est vrai*

*Gina – Mais c'est pas très pratique*

*E – Mm..*

[temps de silence de plusieurs secondes où Gina fait des mouvements de souris sur l'éditeur qui ne semblent pas finalisés]

*E – Alors retourne voir le Plus de détails peut-être..*

Gina ouvre le tutoriel

*E – Il y a quelque chose là que t'as compris.. qu'est-ce que tu as compris ?*

*Gina – Que j'pouvais mettre semer une graine juste en-dessous de avancer.. que ben*

*E – Là c'est ce que tu as fait mais.. est-ce que ça c'est ce que tu voulais ?*

*Gina – Pas trop*

*E – Non.. mais par contre là ce que tu as compris c'est que.. le bloc.. C'est comment fonctionne le bloc répéter.. c'est à dire qu'il faut mettre des blocs à l'intérieur*

*Gina – Mm*

*E – Au début tu ne le faisais pas fonctionner parce que tu ne mettais pas de bloc à l'intérieur.. maintenant tu arrives à le faire fonctionner.. alors pas encore comme tu voudrais.. mais parce que tu mets des blocs à l'intérieur.. mais regarde aussi.. regarde le reste.. qu'est-ce que.. en-dessous tu vois.. c'est quoi ce petit écran ? [le clavier numérique virtuel]*

*Gina – Mm j'sais pas trop (survole le clavier virtuel avec sa souris)*

*E – Alors est-ce que tu peux lire ce qui est écrit ?*

*Gina – Les boucles c'est pratique..*

*E – Non juste au-dessus*

Gina – Pour changer le nombre de répétitions, cliquez sur le nombre, puis utilisez le clavier

*E – Ça veut dire quoi ?*

Temps de silence pendant lequel Gina survole le clavier virtuel avec sa souris

*Gina – Ben je sais pas*

*E – Tu sais pas.. ça veut.. changer le nombre de répétitions.. est-ce que tu comprends tous les mots ?*

*Gina – Euh.. ben j'comprends pas trop le nombre de répétitions en fait*

*E – Alors tu vois.. le nombre de répétitions c'est le nombre de fois que le robot il va répéter ce que tu mets à l'intérieur du bloc*

*Gina – Ok*

*E – Et donc là c'est 10 et donc toi tu ne veux peut-être pas que ce soit 10.. tu veux que ce soit un autre nombre*

*Gina – Ok*

*E – Et donc si tu veux que ce soit un autre nombre il faut changer.. et on t'explique comment il faut faire pour changer.. comment il faut faire pour changer ?*

*Gina – Euh j'appuie sur le petit écran*

***E – Voilà.. tu cliques d’abord sur le nombre pour faire apparaître l’écran et ensuite tu choisis le nombre sur la petite.. sur le clavier.. avec le nombre qui apparaît.. là tu peux faire Masquer les détails puis essayer de changer le nombre***

De retour sur l’éditeur, Gina pointe le nombre du bloc répéter dans la réserve, ce qui provoque l’intervention de l’expérimentatrice

***E – C’est dans celui que tu as déjà mis***

Gina sort les blocs qui étaient dans le bloc répéter et les laisse dans l’éditeur, ensuite elle fait des mouvements saccadés avec sa souris.

***E – Si tu cliques sur le nombre.. sur le 10 dans le bloc que tu as déjà mis***

***Gina – celui-là*** (le survole avec la souris)

***E – Tu cliques sur le 10.. vas-y..*** (ce que fait Gina) ***voilà et là tu peux choisir***

Gina pointe le 8, puis va survoler quelques cases de la grille avec sa souris, avant de revenir cliquer sur le 8 puis du bouton pour enregistrer le nombre entré. Le mouvement vers la grille indique un besoin de vérification du dénombrement.

Suit un moment où Gina essaie de déplacer les blocs laissés dans l’éditeur. Elle vise mal et déplace tout l’éditeur, ce qui engendre quelques difficultés à replacer le programme correctement dans l’éditeur.

***E – Reprends le programme du robot.. pour le remettre bien.. voilà***

Après cela, Gina replace les deux blocs, avancer et semer une graine dans le bloc répéter. Elle se ravise et déplace le bloc semer une graine après la boucle. Puis elle lance une exécution en mode pas à pas.

Le robot sème la graine une case plus loin que la zone de terre (id 20030). Dans cette confrontation, l’erreur ne semble pas liée directement au dénombrement, mais au bloc avancer placé avant la boucle qui n’a pas été remis en cause après la consultation du tutoriel.

***E – Ah c’est presque ça***

La réaction de Gina est d’ajuster le nombre d’itération en retirant une, elle entre 7 à la place de 8, ce qui donne la solution non optimale d’id 13533. La validation se fait via une exécution en mode pas à pas.

### Synthèse

La manipulation des blocs et les exécutions successives ont permis à Gina de progresser dans la prise en main du bloc répéter. C’est cet aspect qui a semblé posé le plus de problème et non le concept de répétition lui-même.

Si on replace cette confrontation dans la progression de Gina, elle correspond au moment où les schèmes d’usage du bloc répéter sont assimilés. En effet :

- Lors de sa première confrontation 10 mois auparavant, nous avons un extrait similaire avec essai en séquence, consultation du tutoriel avec étayage de l’expérimentatrice, mais visiblement il n’y a pas eu rétention sur un temps si long.
- En revanche, après la résolution de ce problème, les schèmes d’usage du bloc de boucle sont mobilisés de manière experte
  - dans la résolution de la version 2 étoiles du même problème juste après, puzzle résolu en 1’42 alors qu’il est un peu plus difficile, car il est nécessaire de placer des instructions avant la boucle.
  - environ deux mois plus tard, le 26/02/2002, Gina mobilise spontanément le bloc de boucle dans la version 3 du problème 1 alors que la contrainte en nombre de blocs ne l’impose pas, et elle résout le puzzle équivalent au présent puzzle avec un profil expert (en 1’04 au lieu de 11’08)
  - lors du troisième tour, Gina mobilise à nouveau spontanément le bloc répéter de manière experte, la confrontation avec le puzzle équivalent dans le parcours n’est pas comparable en termes de temps de résolution car elle rencontre une difficulté liée à la prise en main d’un nouveau contexte.

## Gina, 2022t1p3v3 – profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

veajyt39 ce1

score global : 100

temps moyen par bloc : 8.2

3 - Planter des fleurs [2022-1 white]

medium durée : 4'25

ind. rapidité normalisée : 0.8

0 - 30'52

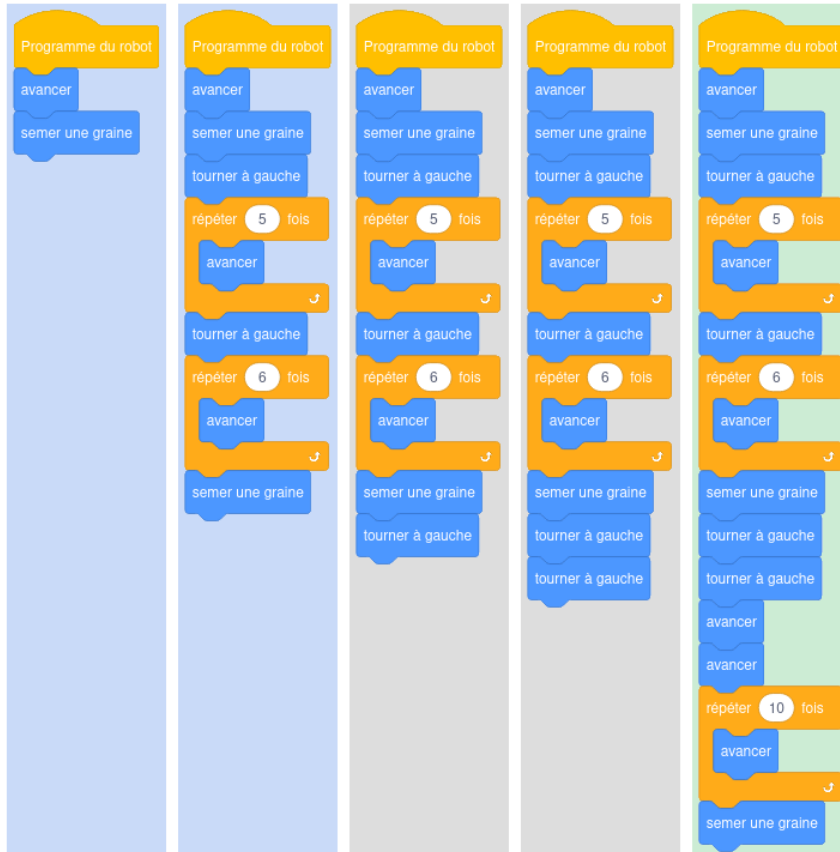
1 - 31'10  
0'18

2 - 33'08  
1'58

3 - 33'34  
0'26

4 - 33'53  
0'19

5 - 35'17  
1'24



id sol :

120

4153

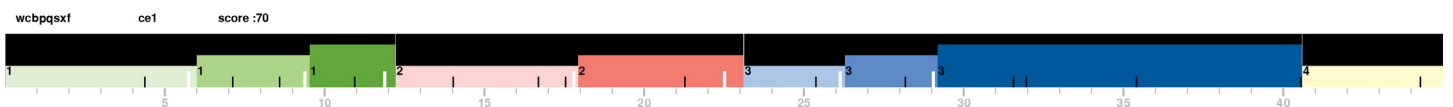
21691

18305

21692

## Inès, CE1 en 2022

### Inès, 2022 tour 1



## Inès, 2022t1p4v1 – profil échec

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

[lien vers l'extrait vidéo 3](#)

wcbpqsxf ce1

score global : 70

temps moyen par bloc : 20.7

4 - Planter des fleurs [2022-1 white] basic

durée : 3'43


ind. rapidité normalisée : 1.15

0 - 40'36 1 - 44'19

3'43



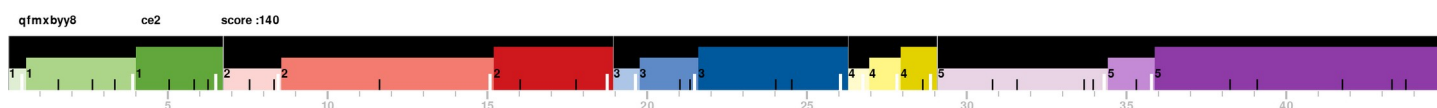
id sol : 22502

	Description	Interprétation
	<p><b>2022t1s4v1_wcbpqsxf_CE1.mp4</b></p> <p>0'00 S - <i>Ouh là là</i></p> <p>0'04 pt T1</p> <p>0'07 S – <i>Avancer de une fois.. là il y aura besoin de plein de [?]. euh là</i> (pt REP)</p> <p>0'19 ouvre comp</p> <p>0'22 S – <i>Euh ça va être dur</i></p> <p>0'23 comp 1dans la réserve</p> <p>0'27 aj REP</p> <p>0'30 S – <i>Une fois.. avancer</i> (aj AV avant REP).. <i>quoi ? non mais</i> (vide l'éditeur)</p> <p>0'37 S – <i>on va mettre une fois</i> (aj REP avec comp à 1).. <i>avancer..</i> (rep AV) <i>semer la graine</i> (aj SG).. <i>une fois</i> (aj REP 1) <i>avancer</i> (rep2 AV)</p> <p>0'54 E – <i>Alors quans tu mets répéter une fois.. tu n'as pas besoin de mettre le répéter</i></p> <p>1'01 S – <i>Je n'ai pas besoin de mettre le répéter ?</i></p> <p>1'03 E – <i>Ben quand tu le fais qu'une fois.. tu mets seulement un bloc répéter.. c'est quand tu veux le faire plus que tu mets le répéter.. répéter une fois avancer ou alors avancer tout seul ça fait pareil</i></p> <p>1'17 S – <i>Mais alors comment on fait pour enlever le répéter du coup ?</i></p> <p>1'20 E – <i>Ah ben tu dois enlever les deux et tu remets le avancer après.. ou tu sors le avancer.. tu le mets sur le</i></p>	<p>Inès ressent la difficulté du puzzle, qu'elle attribue au nombre de graines à planter</p> <p>dextérité qui entrave l'action</p>

	<p><b>côté.. et puis après tu enlèves le répéter</b></p> <p>1'30 sup AV2, sup REP2</p> <p>1'35 S – <b>Allez [?] enlève un répéter.. donc là.. répéter</b> (pt REP1)</p> <p>1'45 mvt</p> <p>1'49 S – <b>Donc en fait.. si j'ai bien compris.. on doit faire une, deux, trois, quatre, cinq, six, sept, huit</b> (pt zones de terre).. <b>huit trucs ?.. on doit fait huit répéter ?</b></p> <p>2'10 E – <b>Huit répéter.. qu'est-ce que tu veux dire par huit répéter ?</b></p> <p>2'13 S – <b>Ben euh.. pour semer des graines on met euh.. on met répéter huit fois avancer.. et puis après on met huit fois semer des graines ?</b></p> <p>2'27 E – <b>Eh bien vas-y, essaie</b></p> <p>2'30 S – <b>Ben je sais pas.. je..</b></p> <p>2'31 mvt</p> <p>2'34 comp 8</p> <p>2'40 S – <b>Je vais essayer.. avancer.. et (aj 4 SG) un, deux</b> [perturbation par le petit frère] <b>un deux.. j'ai droit de le mettre dans le répéter ?</b> (pt SG1)</p> <p>3'06 E – <b>Oui tu as le droit.. mais tu as vu.. tu pouvais mettre.. ce que tu veux dans le répéter</b></p> <p>3'10 déplace les 5 SG dans REP</p> <p>3'12 S – <b>Ouh.. donc là ça fait une deux trois quatre cinq..</b> (pt SG1 à SG5) <b>six..</b> (repSG,SG,SG) <b>non là</b> [perturbations du papa] <b>c'est semer des graines</b></p> <p>3'42 S – <b>Faut faire tout</b></p> <p>3'43 soum 1 → « Il y a déjà une fleur ici ! »</p>	
	<p>3'47 E – <b>Il faut que tu prennes le mode pas à pas pour voir l'erreur que tu as fait</b></p> <p>3'51 S – <b>Ok</b></p> <p>3'52 ferme feedback</p> <p>3'58 pp</p> <p>4'06 E – <b>Alors.. regarde bien</b></p> <p>4'10 S – <b>Du tas de terre.. du tas de terre</b></p> <p>4'12 fin de la session</p>	

## Élana, CE2 en 2021, CM1 en 2022

### Élana, 2021 tour 3



# Élana, 2021t3p5v1 – profil accommodation

[lien vers l'extrait vidéo](#)

qfmxbyy8 ce2

score global : 140

temps moyen par bloc : 11.5

5 - Atteindre le drapeau [white 2021]

basic

durée : 5'15

ind. rapidité normalisée : 4.48

0 - 29'06

1 - 30'50  
1'44

2 - 31'36  
0'46

3 - 33'42  
2'06

4 - 33'59  
0'17

5 - 34'21  
0'22

id sol :

18847

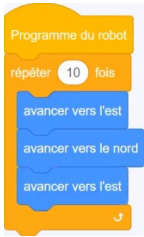
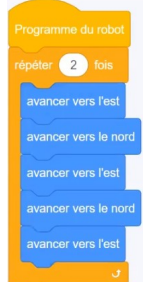

18848

18849

18850

18851

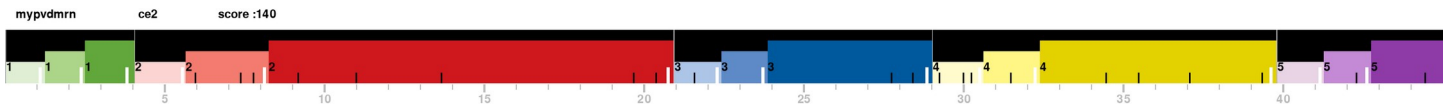
	Description	Interprétation
	<p><b>Version basic : 2021t3s5v1_qfmxbyy8_CE2.mp4</b></p> <p>(10g) 0'00 survole la grille</p> <p>(9, X9) 0'04 pr-rel S</p> <p>(2) 0'10 ajREP</p> <p>(7g) 0'13 survole la grille en diagonale de bas en haut</p> <p>(7r, lat) 0'15 survole la réserve, lat 8s</p> <p>(7g) 0'25 survole la grille (cases accessibles au robot)</p> <p>(7pr) 0'31 ptS</p> <p>(7pg) 0'33 pt la case de gauche de chaque marche</p> <p>(7r) 0'40 survole la réserve</p> <p>(7pg) 0'44 pt marche 0 (celle du robot) D, m1D, m0D, m1G, m0D, m1G</p> <p>(+det) 0'56 det+</p> <p>(7pg) 1'02 pt robot sur la capture de gauche, suit le fond rose</p> <p>(-det) 1'07 det-</p> <p>(7g) 1'14 survole les cases de gauche de chaque marche</p> <p>(7r) 1'17 survole la réserve</p> <p>(3) 1'20 repS</p> <p>(7pg) 1'22 pt m0D, m1D</p> <p>(3) 1'26 repN</p> <p>(7pg) 1'34 pt un buisson sur m2</p>	<p>est induit en erreur par le mot Sud à proximité de la case cible</p> <p>→ faire des grilles rectangulaires ou mettre les directions sur une rose des vents</p>

	<p>(3) 1'37 repE  (5a) 1'43 soum1 → buisson !</p>	
	<p>(lat) 1'45 lat 7s  1'52 feedback-  (7pg) 1'53 ptrib  (7e) 1'55 survole le bas du programme  (7pg) 1'59 pt rob, m0D  (X3) 2'05 supS et replace N et E dans REP  (3) 2'19 rep E en première position  (5a) 2'28 soum 2 → buisson !</p>	
	<p>(12) 2'35 pt m0G (le robot est resté en position de fin d'exécution), m0D, m1G, buisson à droite du robot, m2G  (7r) 2'46 survole la réserve  2'50 pt det  2'54 feedback-  (+det) 3'00 det+  (7g, 8) 3'02 survole la zone autour du robot sur les deux captures, pt texte de droite, pt texte de gauche, pt texte de droite, pt rob sur capture de droite  (-det) 3'25 det-  (7pg) 3'28 pt rob  (3) 3'40 repE  (4) 3'44 comp 2  (7pg) 3'54 pt m1G, m0D, m1G  (X3) 4'02 sup E3  (7pg) 4'05 pt rob, m0D, m1G, rob, m0D, m1G  (3) 4'15 repN  (7pg, lat) 4'20 pt coin supérieur gauche de m2G, lat 5s  (3) 4'25 repE  (5a) 4'33 soum 3 → buisson !</p>	<p>À quoi correspond ce 2 ?</p>
	<p>4'40 feedback-  (X3) 4'48 enlève le dernier E du bloc répéter et le laisse dans l'éditeur  (5a) 4'50 soum 4 → pas arrivé</p>	



	<p>(12) 4'58 sur E resté dans l'éditeur          (6) 5'03 ajE, ajN en-dessous du bloc REP          (5a) 5'11 soum 5 valide          5'17 fin</p>	<p>débugage par complétion du programme</p>
--	--	---

## Élana, 2021 tour 2

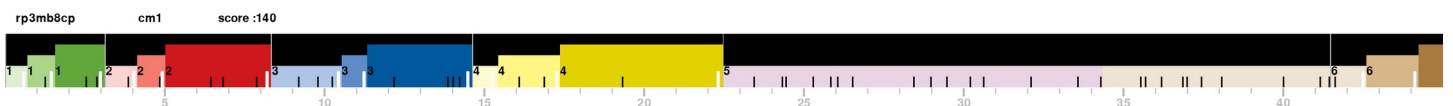


### Élana, 2021t2p5v1 - profil expert

[lien vers l'extrait vidéo](#)

pas de verbalisation mais confrontation intéressante pour montrer que les mouvements de souris même seuls nous permettent d'accéder au raisonnement

## Élana, 2022 tour 3



### Élana, 2022t3p5v1 - profil échec

[lien vers l'extrait vidéo](#)

rp3mb8cp cm1 score global : 140 temps moyen par bloc : 7.5  
 5 - Dans l'espace [2022-3 white] basic durée : 11'50 ind. rapidité normalisée : 9.52

id sol :	1 - 23'28 0'59	2 - 24'21 0'53	3 - 24'28 0'07	4 - 25'19 0'51	5 - 25'52 0'33	6 - 26'05 0'13	7 - 26'33 0'28	8 - 28'28 1'55	9 - 29'00 0'32	10 - 29'30 0'30	11 - 30'14 0'44	12 - 30'39 0'25	13 - 32'08 1'29	14 - 33'36 1'28	15 - 34'19 0'43

Noté pendant la session : tâtonnement dans l'identification du motif ; se heurte à la limite de blocs ; la consultation du tutoriel ne permet pas de débloquer ; **ne maîtrise pas le caractère cyclique de la boucle, le fait que faire entrer des instructions en début de boucle implique d'enlever ces mêmes instructions en fin de boucle.**

survole brièvement la grille sur la diagonale

ajREP

met AV avant REP

pointe les objets ou la limite entre objet et astéroïde

comp 5

pointe la case avant le robot

rep AV RAM AV RAM TG AV

→ nous faisons l'hypothèse qu'Élana considère les astéroïdes comme des objets, ce qui serait cohérent avec le fait qu'elle place un bloc avant la boucle et qu'elle mette deux blocs RAM dans le motif. Avec cette hypothèse, il y aurait seulement une erreur de remplacement du robot dans la bonne orientation

Exécution du programme d'id.19844

ferme le feedback

E – [prénom] ***est-ce que tu pensais que l'astéroïde était un objet à ramasser ?***

pointe l'astéroïde puis survole son programme

E – ***Hein ?***

décroche la séquence après le premier RAM

survole la grille, pointe la case devant le robot

repTG AV

Exécution du programme d'id.19845

enlève le dernier bloc AV

Exécution du programme d'id.19846

ferme le feedback

exécute le programme jusqu'à l'endroit de l'erreur en pp

→ **le fait qu'elle ne stoppe pas à la fin de la première itération montre qu'elle n'a pas fait le lien entre le motif visuel et le motif algorithmique**

passe le premier avancer de la boucle avant la boucle

Exécution du programme d'id.19847

pointe le robot

rep TG

pointe la case au-dessus de l'objet

rep AV

Exécution du programme d'id.19848 en mode pp

remplace le deuxième TG par TD

Exécution du programme d'id.16028

repAV

→ complète en allongeant le motif

Exécution du programme d'id.44 en mode pas à pas pour la première itération, en mode normal ensuite

pointe les quatre premiers astéroïdes

ferme le feedback

relance l'exécution en mode normal et stoppe lorsque le robot est sur la case en-dessous de la fusée

comp 4

Exécution du programme d'id.45 jusqu'à ce que le robot soit sur la case devant la quatrième astéroïde (non enregistré)

fait un aller-retour entre la grille et le programme

aj TG puis le supprime

Exécution du programme d'id.45 jusqu'à ce que le robot soit sur la case devant la deuxième astéroïde (non enregistré)

comp 5

déplace le curseur dans la diagonale, s'arrête sur le dernier objet et déplace lentement le pointeur vers la fusée

Exécution du programme d'id.44 jusqu'à ce le robot soit sur la fusée (non enregistré)

→ **à ce moment, Élana aurait pu se rendre compte que le robot étant arrivé, les deux dernières instructions sont en trop, et alors décaler le motif en faisant entrer à leur place les deux blocs placés avant la boucle (image de la fenêtre glissante : où faut-il placer la fenêtre pour avoir le plus de motifs possibles ?)**

pointe le robot, puis les deux blocs avancer en fin de boucle, décroche puis raccroche ces deux blocs, en place un après la boucle

→ il manque le schème décrit précédemment

Exécution du programme d'id.45

remet le bloc AV dans la boucle

ouvre le tutoriel

le pointeur parcourt le tutoriel → lecture

tente de fermer le tutoriel avec l'icône à côté de Détails de la missions, qu'elle prend pour un bouton



ferme le tutoriel

Exécution du programme d'id.44

parcourt la fin du trajet avec le pointeur

comp 4

aj AV

Exécution du programme d'id.46

ferme le feedback

Nouvelle exécution du programme d'id.46 en mode pas à pas

sup le bloc AV après la boucle

Exécution du programme d'id.45

pointe le robot

aj TG AV

pop-up trop de blocs

supprime le dernier bloc AV

Exécution du programme d'id.2089

Survole les deux blocs AV avant la boucle

aj AV → trop de blocs

décroche et raccroche la boucle

ouvre et referme plus de détails

le pointeur parcourt l'interface signe qu'Élana est bloquée, elle a dans l'éditeur un programme qui permettrait au robot de remplir la mission, mais qui utilise trop de blocs

supprime le dernier bloc AV

début d'exécution en pas à pas et remet en situation initiale

Exécution du programme d'id.2089

aj AV

fait passer le premier AV dans la boucle

supprime le dernier AV (qui était après la boucle) → ne compense pas j'ajout du précédent

Exécution du programme d'id.19850

mouvements saccadés du pointeur

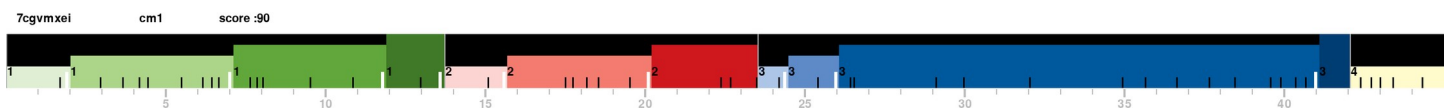
survole le milieu de la séquence dans la boucle, décroche le trois derniers blocs

imbrique un REP avec AV AV dedans, manipule les blocs sans que je comprenne ce qu'elle fait, pour arriver au programme d'id.19851 qu'elle exécute

exécute à nouveau ce programme en pp

quitte le puzzle

## Élana, 2022 expérimentation complémentaire



### Élana, 2022t4p1v4 – profil ajustement

[lien vers l'extrait vidéo](#)

7cgvmxci cm1

score global : 90

temps moyen par bloc : 11.3

1 - Planter des fleurs [uLille v2] hard

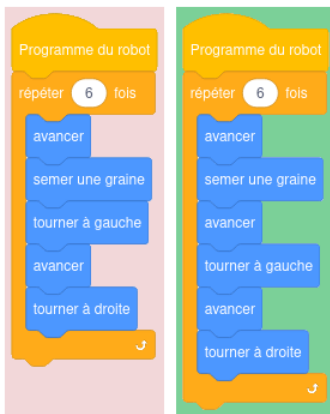
durée : 1'44

ind. rapidité normalisée : 0.02

0 - 11'54

1 - 13'00  
1'06

2 - 13'38  
0'38



id sol :

5039

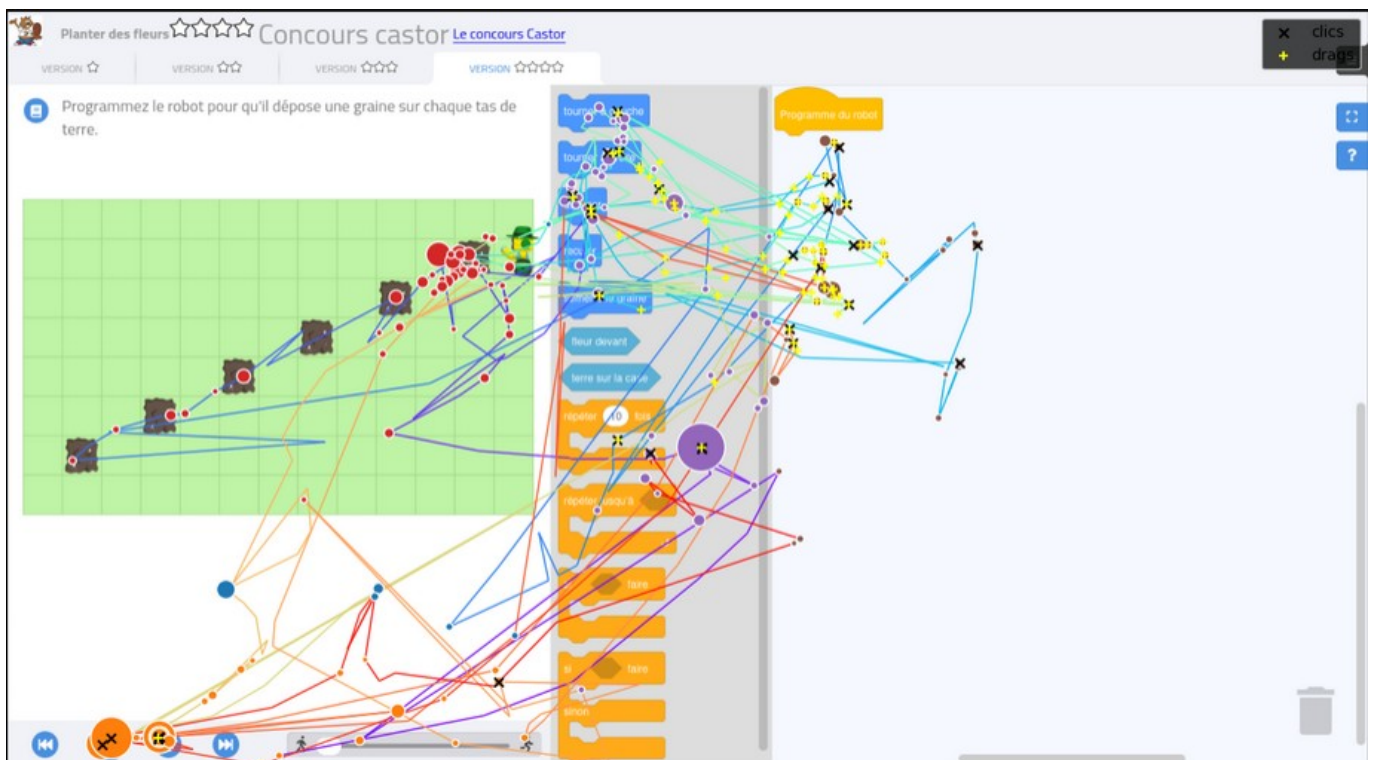
3400

pas de verbalisation

parcours de la diagonale de zones de terre avant de renseigner le compteur de boucle, **quelques pointages sur la grille en cours d'édition du motif**

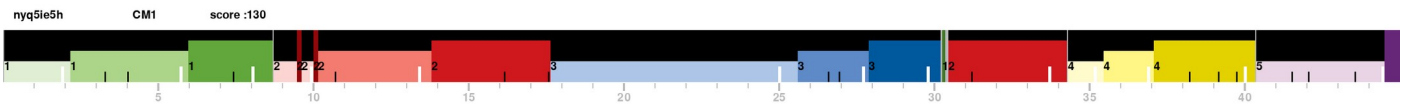
le pointeur reste sur le bouton du mode pas à pas pendant la phase de débogage

voir ce que ça donne comme visualisation des mouvements de souris



Ali, CM1 en 2021, CM2 en 2022

Ali, 2021 tour 1



**Ali, 2021t1p1v1 – profil indéfini**

[lien vers l'extrait vidéo](#)

**Ali, 2021t1p1v3 – profil ajustement**

[lien vers l'extrait vidéo](#)

nyq5ie5h cm1

score global : 130

temps moyen par bloc : 14.4

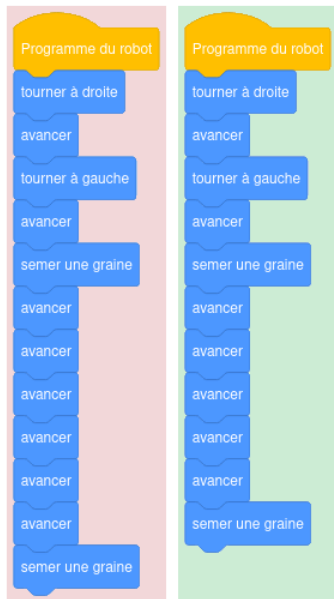
1 - Planter des fleurs [white 2021] medium durée : 2'08

ind. rapidité normalisée : -0.11

0 - 5'58

1 - 7'27  
1'29

2 - 8'06  
0'39



id sol :

4503

4504

**Ali, 2021t1p3v1 – profil indéfini**

[lien vers l'extrait vidéo](#)

**Ali, 2021t1p3v2 – profil ajustement**

[lien vers l'extrait vidéo](#)

nyq5ie5h cm1

score global : 130

temps moyen par bloc : 14.4

3 - Ranger les billes [white 2021] easy

durée : 2'10

ind. rapidité normalisée : 0.13

0 - 25'36

1 - 26'37  
1'01

2 - 26'58  
0'21

3 - 27'46  
0'48



id sol :

17867

17868

880

### Ali, 2021t1p4v2 – profil expert

[lien vers l'extrait vidéo](#)

nyq5ie5h cm1

score global : 130

temps moyen par bloc : 14.4

4 - Planter des fleurs [white 2021] easy

durée : 1'30

ind. rapidité normalisée : 0.25

0 - 35'27

1 - 36'57  
1'30



id sol :

883

Cette confrontation est intéressante car elle suit celle sur la version 1 étoile que le sujet a validée avec une séquence.

Tour 1 version easy : 2021t1s4v2\_nyq5ie5h\_CM1\_passage\_reconnaissance\_de\_motif\_s4v2.mp4

Ali – *Aïe, aïe, aïe ! Ouais.. C'est pas juste..*

E – *Qu'est-ce qui t'embête ?*

Ali – *ben, il faudrait répéter*

E – *Ben oui, qu'est-ce que tu dois répéter ?*

Ali – *Bah.. une fois.. chaque fois.. Attends je fais Plus de détails*

Ali maîtrise déjà l'utilisation du bloc répéter avec une instruction dedans (validation des trois premières versions du problème 3), et il a trouvé le motif (version 1 étoile de ce problème) donc nous supposons que ce qu'il n'a pas compris, c'est qu'il peut placer plusieurs instructions dans le bloc répéter.

Pour le tutoriel générique : montrer un bloc répéter avec plusieurs instructions dedans en enlevant les labels.

E – *Ah, bonne idée !*

ouvre le tutoriel en cliquant sur *Plus de détails* et lit en silence pendant plusieurs secondes.

E – *Regarde bien*

Ali – *Ah d'accord j'ai compris.. Ah d'accord, d'accord, d'accord.. d'accord, j'ai compris.. j'ai compris*

Le sujet ferme le tutoriel

Ali – *Alors !*

(2b) compte les zones de terre à voix haute en les pointant avec sa souris

(3a) place un bloc répéter dans l'éditeur

(4b) renseigne le compteur de boucle avec le nombre 12.

Ali - *12 fois..avancer..semer une graine*

(6b) place les blocs dans le bloc répéter en même temps qu'il énonce les actions.

Ali – *Donc là si je fais ça ?*

(11a) lance l'exécution en cliquant sur le bouton play et valide

### Ali, 2021t1p4v3 – profil accommodation

[lien vers l'extrait vidéo](#)

nyq5ie5h cm1

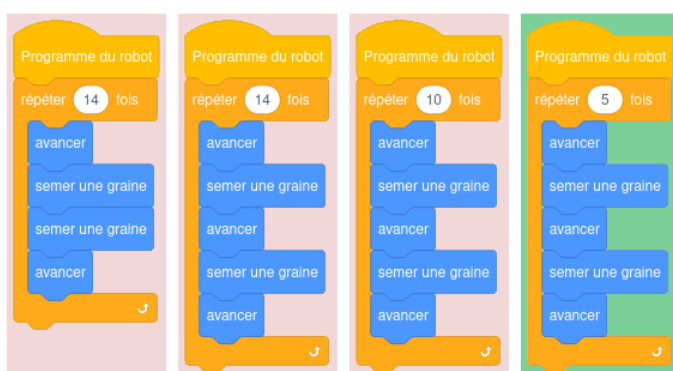
score global : 130

temps moyen par bloc : 14.4

4 - Planter des fleurs [white 2021] medium durée : 3'00

ind. rapidité normalisée : 0.56



0 - 37'04 1 - 38'15 2 - 39'11 3 - 39'46 4 - 40'04  
1'11 0'56 0'35 0'18




id sol : 17869 15450 884 885

	Description	Interprétation
	<p><b>Tour 1 version medium :</b> 2021t1s4v3_nyq5ie5h_CM1.mp4</p> <p>0'00 S – <i>Passer à la suite.. ou..uh.. ouh.. ouh.. ouh..</i></p> <p>(ExMT) E – <i>Alors est-ce que tu vois comment le faire ce-lui-là ?</i></p> <p>S – <i>Oui je sais.. bah attends.. je crois savoir comment faire.. en gros..</i></p> <p>(2) accroche un bloc répéter dans l'éditeur.</p> <p>(1p) pointe les cases avec sa souris,</p>	<p>Pour le schème de dénombrement, l'unité est la case.</p> <p>Pour l'identification du motif, il oublie un bloc, mais l'unité visuelle semble repérée.</p> <p>Le schème du dénombrement et celui de l'identification de motif ne sont pas articulés.</p> <p>geste / schème ? : ouvre le clavier numérique après avoir accroché le bloc répéter, puis se rend compte qu'il n'a pas l'information disponible en tête, donc déplace la souris vers la grille pour le schème de comptage.</p>



	<p>(Cl) ouvre le clavier numérique,</p> <p>(1) retourne survoler la grille</p> <p>(4) renseigne le compteur de boucle avec le nombre 14.</p> <p>(3) place un bloc avancer, deux blocs semer une graine et un bloc avancer dans le bloc répéter</p> <p>(5a) puis lance une exécution.</p>	
	<p>1'13 Le robot s'arrête sur la première zone de terre. Le feedback « Il y a déjà une fleur ici ! » apparaît à l'écran.</p> <p>(ExMT) E – <i>Est-ce que tu comprends ce qui se passe là ?</i></p> <p>S – <i>Y'a déjà une fleur ici.</i></p> <p>(ExPP) E- <i>Oui regarde, si tu mets la petite croix orange.. voilà.. et le bouton pour faire un à la fois.</i></p> <p>(5b) Le sujet exécute en mode pas à pas.</p> <p>(ExDE) – <i>Il va avancer.. il va semer une fleur.. une graine</i></p> <p>S – <i>Ah d'accord j'ai compris j'ai compris.. alors.. alors je vais faire ça</i></p> <p>(3) Il intercale un bloc avancer entre les deux blocs semer une graine.</p> <p>S – <i>Je vais le mettre là.. et si je fais ça..</i></p> <p>(5a) Il lance une exécution.</p>	<p>Phase de correction du motif.</p> <p>La séquence qui constitue le motif est correcte. Il le constate à l'exécution.</p> <p>Il y a discordance entre l'unité constituée dans le programme (séquence d'instructions) et l'unité du dénombrement (cases). C'est cette mise en correspondance des unités dans les deux représentations qui fait défaut. Le sujet ne se rend pas compte qu'il ne compte pas la même chose.</p> <p>Une hypothèse est que pour le schème du dénombrement, le théorème-en-acte suivant soit tenu pour vrai : « les éléments à compter sont les plus petites unités visibles », ce qui est la situation la plus courante.</p>
	<p>2'08 (ExDE) E – <i>Il a bien tout rempli mais..</i></p> <p>S – <i>Alors attends..</i></p> <p>(1p) Le sujet parcourt les zones de terre en les pointant avec sa souris.</p> <p>S – <i>Ah d'accord il faut 10 fois.</i></p> <p>(4v) Il change le nombre dans le compteur de boucle à 10.</p> <p>(ExMT) E – <i>Et pourquoi il faut 10 fois ?</i></p> <p>(5a) Il lance l'exécution.</p> <p>S – <i>Voilà 10 fois..</i></p>	<p>Cette fois, pour le dénombrement, l'unité est la zone de terre. Il y a accommodation, remise en cause partielle du théorème-en-acte précédent, mais pas de manière correcte : il y a sélection entre les unités précédentes, ne compte que ce qui est saillant visuellement.</p>

	<p>2'42 S – <i>Ah d'accord..</i></p> <p>(ExDE) E – <i>Le robot sort de la grille</i></p> <p>S – <i>Ah d'accord, j'ai compris</i></p> <p>(4v) Il change le nombre dans le compteur de boucle à 5.</p> <p>S – <i>5 fois ça.. voilà</i></p> <p>(5a) Il lance l'exécution et valide.</p> <p>E – <i>Et pourquoi il faut 5 alors ?</i></p> <p>S – <i>Parce qu'il y en a cinq.</i></p>	<p>Enfin, l'unité est adéquate, un ensemble de deux zones de terre.</p>
---	---	---

## Ali, 2021 tour 2



### Ali, 2021t2p1v3 – profil indéfini

[lien vers l'extrait vidéo](#)

knb7bv65 cm1

score global : 120

temps moyen par bloc : 8.6

1 - Peindre le motif [white 2021] medium durée : 2'05

ind. rapidité normalisée : 0.82

0 - 2'50

1 - 4'55  
2'05



id sol : 5254

### Ali, 2021t2p5v1 – profil accommodation

[lien vers l'extrait vidéo](#)

knb7bv65 cm1

score global : 120

temps moyen par bloc : 8.6

5 - Planter des fleurs [white 2021] basic

durée : 6'15

ind. rapidité normalisée : 4.45



verbalisation mais beaucoup de bruit familial derrière

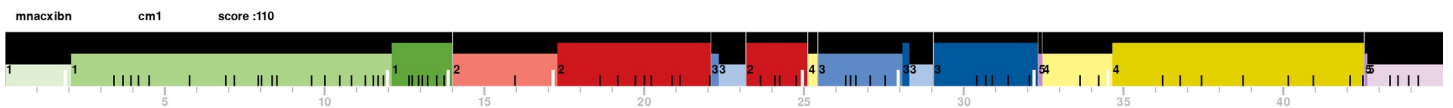
représentation de tourner = pivoter + avancer

la correction lors du premier passage consiste à allonger la séquence qui constitue le motif algorithmique, et aboutit à un abandon

lors du second passage tutoriel qui amène la compréhension mais pas d'explicitation de celle-ci explicitation lors de la jonction intéressante

## Alix, CM1 en 2021, CM2 en 2022

### Alix, 2021 tour 1 passage 1



### Alix, 2021t1p1v1 - profil indéfini

[lien vers l'extrait vidéo](#)

### Alix, 2021t1p1v2 - profil accommodation

[lien vers l'extrait vidéo](#)



### Alix, 2021t1p1v3 - profil accommodation

[lien vers l'extrait vidéo](#)

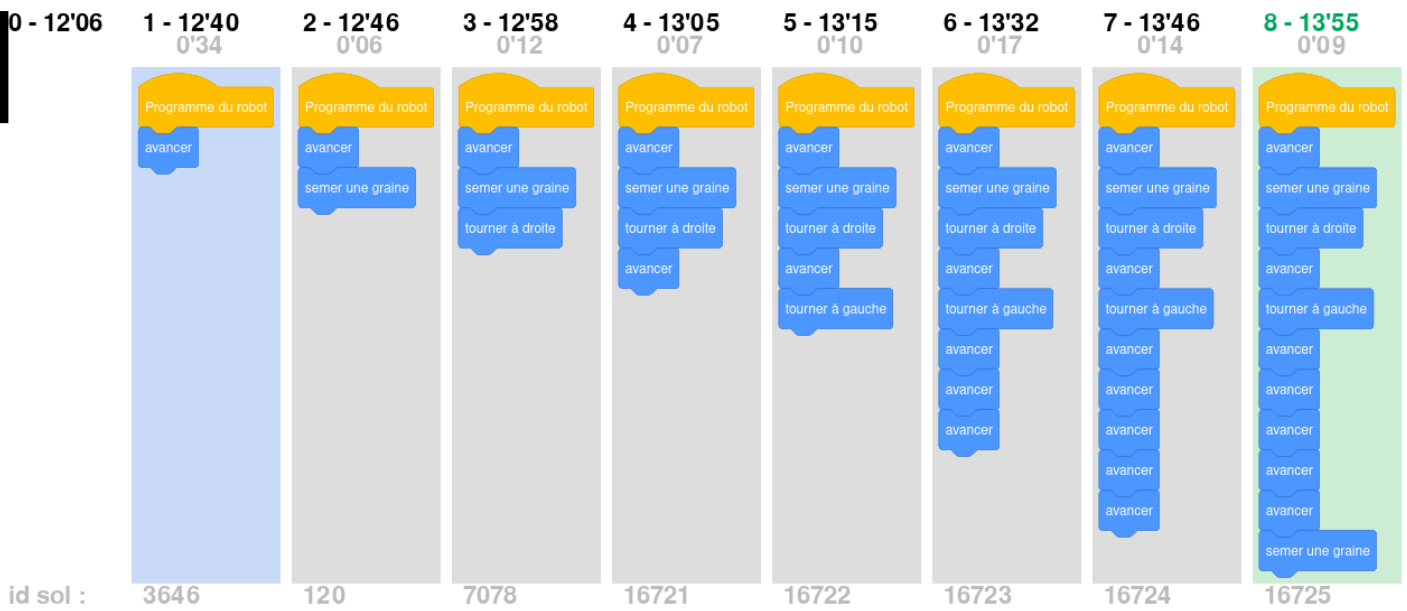
mnacxibn cm1

score global : 110

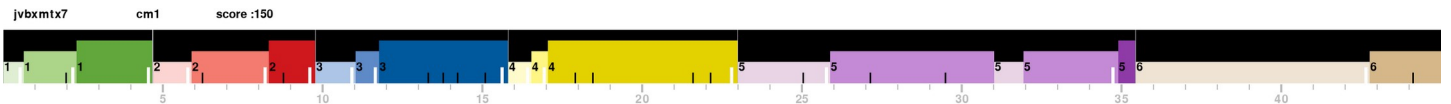
temps moyen par bloc : 22.9

1 - Planter des fleurs [white 2021] medium durée : 1'49

ind. rapidité normalisée : -0.52



### Alix, 2021 tour 1 passage 2



### Alix, 2021t1p4v3 – profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

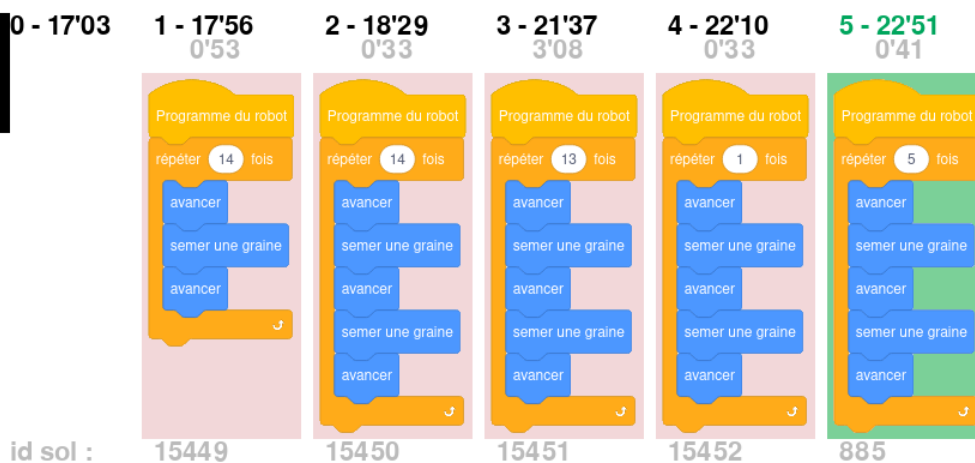
jvbxmtx7 cm1

score global : 150



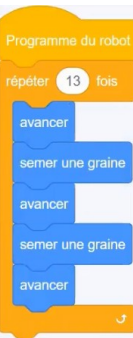
temps moyen par bloc : 9.7


4 - Planter des fleurs [white 2021] medium durée : 5'48


ind. rapidité normalisée : 3.48



	Description	Interprétation
--	-------------	----------------

	<p><b>Tour 1 version medium :</b>  <b>2021t1s4v3_jvbxmtx7_CM1.mp4</b></p> <p>0'00 (7pvg) pointe le robot, puis les trois premières cases en chuchotant : avancer, semer une graine, avancer, semer une graine, avancer</p> <p>(1pv) pointe les cases de la ligne où sont les zones de terre en comptant jusque 14</p> <p>(2) accroche un bloc répéter dans l'éditeur</p> <p>(4) renseigne le nombre de répétitions à 14</p> <p>(3) place les blocs avancer, semer une graine, et avancer dans le bloc répéter</p> <p>(5a) soumission 1 → une fleur apparaît une case sur deux et le message « Le robot sort de la grille ! » est affiché</p>	<p>Le feedback de la quantième itération n'est pas encore présente, il a été développé après le premier tour. Le sujet ne repère pas que lorsque le robot sort de la grille, il n'a fait que 6 itérations (c'est le début de la 7ème)</p> <p>Le sujet avait repéré le bon motif verbalement (7b) mais la transcription de ce motif en termes de blocs n'est pas faite aussitôt, le dénombrement est intercalé, et lorsque le sujet code le motif, il ne refait pas une prise d'information et ce n'est pas celui qui a été initialement repéré qui est codé.</p> <p>[TODO : regarder si l'ordre des phases a un impact : est-ce que le moment de la prise d'information a un impact ? ]</p>
	<p>0'53 S – <b>Angh.. avancer, semer une graine, avancer</b></p> <p>remet la grille dans l'état initial</p> <p>(7v) S – <b>Avancer, semer une graine, avancer, semer une graine, avancer</b> (chuchote, la souris reste immobile)</p> <p>(3v) ajoute un bloc semer une graine et un bloc avancer dans le bloc répéter en nommant les blocs.</p> <p>(5a) Soumission 2 → les fleurs apparaissent sur les zones de terre mais le message « Le robot sort de la grille ! » est à nouveau affiché.</p>	<p>Le sujet corrige la séquence dans le corps de la boucle. Pour cela, il verbalise la séquence d'actions à faire réaliser par le robot.</p> <p>Il ne touche pas au compteur de boucle.</p> <p>Il ne fait toujours pas le lien entre le nombre de motifs et</p>
	<p>(Q) 1'26 S – <b>Qu'est-ce qu'il faut que je fasse maintenant ?</b></p> <p>(ExDE) E – <b>Ben tu vois bien qu'il sort de la grille ton robot donc..</b></p> <p>S – <b>Ah oui..</b></p> <p>(X3) enlève le dernier bloc avancer du corps de la boucle sans le lâcher</p> <p>(ExFP, ExPP) E – <b>Remets-le ton bloc et fais avec le mode pas à pas</b></p> <p>(3) remet le bloc avancer à sa place</p> <p>(Q) S – <b>Euh.. il est où ?</b></p> <p>(ExGD) E – <b>Ben le mode pas à pas c'est le petit triangle juste à droite du bouton d'exécution.. le bouton play.. c'est un bouton bleu en bas..</b></p> <p>S – <b>Bleu.. en bas</b></p> <p>E – <b>C'est celui.. le troisième.. non non non non tu vois tu as ton bouton..</b></p> <p>S – <b>Ça ?</b></p> <p>E – <b>Oui c'est ça</b></p> <p>(5b) clique une fois sur le mode pas à pas, puis la souris sort de la zone,</p> <p>(lat) quelques secondes de latence</p>	<p>[TODO : dans la description, une couleur pour ce qui relève de la prise en main de l'EIAH (approche instrumentale), une couleur pour les actions instrumentées]</p> <p>prégnance du schème « dénombrer les cases » ou du théorème-en-acte « les unités à compter sont les éléments les plus élémentaires », ie ici les cases, indépendamment de ce qui se trouve dans le corps de la boucle.</p> <p>Il n'y a pas création mentale d'unités, le sujet prend les unités déjà disponibles.</p>

	<p>(ExGD) E – <b>Donc il faut que tu cliques plusieurs fois de suite.. et à chaque fois il va exécuter une seule instruction.. et tu vas pouvoir voir.. là la prochaine fois il va faire avancer</b></p> <p>(5aX) clique sur le bouton play</p> <p>E – <b>Ah non non non là tu as..</b></p> <p>S – <b>Ah ok.. ouh là</b></p> <p>(ExGD) E – <b>Oui, il faut que tu fasses plusieurs fois sur le bouton pas à pas.. celui-ci.. Voilà.. donc là il te fait une instruction à la fois.. donc là il fait ce que tu veux..</b></p> <p>S – <b>semer une graine.. avancer..</b></p> <p>(5b) exécute en mode pas à pas jusqu'à ce que le robot ait semé sur la dernière zone de terre</p> <p>(ExPP) E – <b>Encore une fois</b></p> <p>(5b) clique sur le bouton du mode pas à pas</p> <p>(ExMT) E – <b>Et là t'aimerais bien qu'il s'arrête</b></p> <p>S – <b>Ouais</b></p> <p>E – <b>Ouais.. et il va continuer</b></p> <p>S – <b>Oui</b></p> <p>E – <b>Ça veut dire quoi ?</b></p> <p>S – <b>Euh.. il faut enlever avancer.. le dernier</b></p> <p>(ExFP) E – <b>Ben non.. regarde c'était bien ça</b></p> <p>S – <b>Oui</b></p> <p>(ExSO) E – <b>Tu le fais trop de fois</b></p> <p>S – <b>Ah d'accord !</b></p> <p>(4) met la valeur du compteur à 13 au lieu de 14</p> <p>S – <b>Alors..</b></p> <p>(1pv) recompte les cases à voix haute jusque 14, une première fois de gauche à droite, puis une nouvelle fois de droite à gauche</p> <p>S – <b>Bon.. mm.. on va mettre ça (laisse la valeur à 13)</b></p> <p>(5a) soumission 3 → même feedback que lors de la soumission précédente</p>	<p>Cette remarque montre que le sujet ne tient pas compte du théorème-en-acte « Si j'enlève une instruction dans le corps de la boucle, cette instruction est enlevée pour toutes les itérations » ; Il ne raisonne qu'avec la dernière itération.</p>
	<p>(ExSO) 4'28 E – <b>Mais regarde.. c'est tout ce que tu as mis à l'intérieur du bloc répéter.. à l'intérieur du bloc orange.. ça compte pour un</b></p> <p>(7e) parcourt la séquence de blocs de haut en bas à l'intérieur du bloc répéter</p> <p>(X3) enlève les deux derniers blocs</p> <p>(ExFP) E – <b>Non non ça c'était bien</b></p> <p>(3) S – <b>Ouh là</b> (ne replace pas le blocs au bon endroit dans un premier temps)</p> <p>(ExSO) E – <b>Vas-y .. mets répéter 1 fois.. au lieu de 13,</b></p>	

	<p><b><i>mets 1 fois</i></b></p> <p>(4) met le compteur à la valeur 1</p> <p>E – <b><i>Voilà, et tu le lances maintenant.</i></b></p> <p>(5a) soumission 4</p>	
	<p>(ExDE) 5'03 E – <b><i>Tu regardes où il arrive.. Tu fais une fois, il fait ça</i></b></p> <p>S – <b><i>Une.. alors</i></b></p> <p>(7pvg) pointe les deux zones de terre avec une fleur et le robot</p> <p>S – <b><i>Un,</i></b></p> <p>(1pv) pointe les cases vides entre les zones de terre et compte en chuchotant</p> <p>(4) met le compteur à 5</p> <p>(5a) soumission 5 valide</p> <p>S – <b><i>Mais mince.. ah oui c'est bon c'est bon</i></b></p>	<p>C'est le passage par l'unité qui débloque le sujet, mais on sent que c'est fragile (réaction à l'exécution)</p> <p>fait le lien entre l'identification de motif et le dénombrement de ces motifs pour renseigner le compteur de boucle</p>

**Alix, 2021t1p5v2 – profil accommodation**

[lien vers l'extrait vidéo](#)

jvbxmtx7 cm1

score global : 150

temps moyen par bloc : 9.7

5 - Collecter les pierres précieuses [white 2021]

easy

durée : 6'29

ind. rapidité normalisée : 4.0

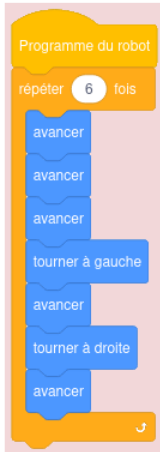
0 - 25'53

1 - 27'10  
1'17

2 - 29'31  
2'21

3 - 31'56

4 - 34'47  
2'51



id sol :

15453

15454

3655

verbalisation

difficulté à identifier le motif : trop long

étude de la jonction entre les deux premiers motifs

difficulté à placer une séquence dans le bloc répéter ( difficulté d'ordre instrumental)



Alix – *Alors donc.. avancer.. avancer.. avancer* (simule le déplacement avec le pointeur) *oui bon euh* (ajREP) *on va mettre ça.. donc ça fait un deux* (pointe la 2ème, la 3ème pierre)

pointe les groupes de deux pierres en comptant tout haut jusque 6

Alix – *Je vais mettre six* (comp 6) *donc.. avancer* (repAV) *avancer* (repAV, repAV) *avancer avancer avancer* (simule le déplacement avec le pointeur) *tourner à gauche* (repTG) *donc tourner à gauche* (survole brièvement la case) *avancer* (repAV) *avancer* (simule le déplacement avec le pointeur) *tourner à droite* (repTD) *avancer* (repAV) *ah mince ça va être trop* (parcourt le trajet)

Se rend compte qu'il y a un problème mais n'a pas été en mesure d'y remédier. **La question posée est la limite du motif algorithmique, la correspondance entre motif visuel de la grille (les deux pierres) et le motif algorithmique (les actions du robot sur cette grille). Suivant que le sujet considère les cases vides dans le motif visuel, cela peut faciliter la mise en correspondance avec le motif algorithmique. Ici le sujet ne prend pas en compte de case vide dans le motif visuel, le motif algorithmique établi ne peut pas être mis en correspondance avec ce motif visuel.**

**Contrairement au déplacement qui peut être simulé par un déplacement du pointeur, les actions de pivotement doivent être simulées mentalement.**

Alix – *Ah oui mince.. non attends c'est bon.. qu'est-ce que j'ai fait ça..*

exécution du programme d'id.15453

Alix – *Euh..*

ferme le feedback

Alix – *Alors..*

remet la grille dans l'état initial

**Après une exécution, deux manières de réfléchir, soit en remettant la grille dans l'état initial, soit en réfléchissant à partir de l'état de la grille à l'issue de l'exécution, ce qui entraîne un raisonnement par complétion, sur un nouveau problème.**

Alix – *C'est bon..* (pp) *avancer* (pp) *avancer* (pp) *avancer* (pp) *tourner à gauche* (pp) *avancer* (pp) *avancer* (alors que le robot n'avance pas mais tourne à droite, pp, pp après temps de latence de 5s) *et donc là ça doit recommencer donc ça fait avancer avancer avancer*

temps de latence : 10s

**Alix analyse la jonction entre les deux premières itérations et se rend compte du problème, sans être en mesure d'y remédier**

Alix – *Euh.. donc.. c'est où qu'on peut avoir plus de détails ?*

E – *Ah bah là y'en a pas.. c'était.. les plus de détails y'en a pas tout le temps*

Alix – *Ok*

E – *C'était sur la version un et t'en as pas eu besoin alors.. mais tu peux peut-être le regarder quand même.. peut-être que ça peut t'aider.. il y avait peut-être quelque chose.. un indice*

Alix – *Quatre blocs restants sur douze autorisés* (le pointeur suit le texte lu)

compte tout haut les blocs du corps de la boucle jusque sept, dit huit en chuchotant et en pointant le bloc répéter, marquant ainsi que le bloc répéter n'a pas le même statut que les autres

Alix – *Là je peux en mettre douze dedans ?*

E – *Tu peux en mettre douze en tout.. donc le bloc orange il compte pour un.. donc tu peux en mettre onze dedans*



Alix – *Ah.. donc.. ah la la la la la.. comment je fais..*

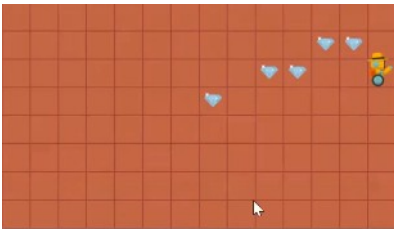
pr-rel le corps de la boucle, sort la séquence du bloc répéter, supprime le premier AV, cherche à remettre la séquence dans le bloc répéter, mais n'a pas compris que c'est le premier bloc qu'il faut mettre en regard de l'espace disponible pour que l'espace s'agrandisse automatiquement. Au lieu de cela, Alix vise au milieu, elle centre la séquence (schème d'usage non adéquat)

Alix – *Ouh là comment je fais..* (déplace la séquence au-dessus du bloc répéter) *ouh là qu'est-ce que j'ai fait* (parvient à remettre la séquence dans le bloc REP) *voilà..*

**Dans le cas de l'environnement par blocs, l'apprentissage de la syntaxe est remplacé par l'apprentissage des gestes, des schèmes d'usage des blocs.**

Alix – *Ça ça fait quoi ?*

exécution du programme d'id.15454



Alix – *Ouh là.. ça commençait bien mais maintenant ça termine mal.. donc ça fait*

Alix a corrigé le trajet du robot mais le motif algorithmique est encore trop long

ferme le feedback, pp

Alix – *Avancer* (pp) *avancer* (pp) *tourner à gauche* (pp) *avancer* (pp) *tourner à droite* (pp) *avancer* (pp) *avancer* (pp) *euh après c'est avancer* (pp) *tourner à gauche* (pp) *avancer* (pp) *tourner à droite* (pp) *avancer* (pp) *tourner.. ah la la* (parcourt le programme de bas en haut) *mm.. comment je fais.. mm..*

**Alix ne prend pas conscience que le robot ne fait pas le même parcours pour ramasser les deux premiers groupes de pierres.** Qu'il y ait une trace du parcours effectué de laissé sur la grille aurait été probablement aidant dans ce cas de figure. **Caractère aidant des éléments de décor lorsqu'ils sont présents.**

Elle continue l'exécution en pas à pas, mais on sent que c'est parce qu'elle ne voit pas autre chose à faire.

Alix – *hum* (pp) *avancer* (pp) *encore avancer* (pp) *tourner à gauche* (pp) *avancer* (pp, pointe la pierre non ramassée)



Alix – *Est-ce que vous pouvez me donner un indice ?*

E – *Et ben retourne sur la version une.. que tu as déjà résolue* (Alix survole les boutons d'exécution) la petite étoile jaune

Alix – *Ça ?* (pointe l'étoile dans le titre)

E – *Non.. mm.. plus à droite..* (pointe l'onglet de la version une étoile) *voilà ici* (ouvre l'onglet) *et ouvre le plus de détails.. peut-être ça va te donner un indice*

ouvre le plus de détails

Alix – *Alors..*

E – *Il va falloir que tu descendes aussi.. il est assez long* (Alix scrolle)

Alix lit à voix haute (le pointeur suit le texte lu)

Alix – *Un deux trois* (pointe les motifs sur l'image du haut) *ben oui..*

E – *mm*

scrolle vers le bas

Alix lit à voix haute (le pointeur suit le texte lu)

Alix – *Han..* (lit tout haut le texte des deux premières images, le pointeur est immobile sous le texte de l'image du milieu) *ok* (scrolle vers le haut) *donc là il faut que je fasse..* (retourne sur la version 2 étoiles) *version deux*

E – *Il faut que tu repères où est ton motif.. parce que il faut que tu replaces ton robot..*

survole le programme, le premier motif, pp

Alix – *Avancer* (pp) *avancer* (pp, pp, pp) *avancer* (pp) *tourner à droite avancer* (pp) *avancer* (pp)

E – *Alors est-ce que.. regarde quand tu as tout parcouru est-ce qu'il est dans la même position par rapport au deuxième motif.. qu'il était par rapport au premier ?..* (remet dans l'état initial) *regarde bien comment il est.. la position de ton robot..*

Alix – *Oui..*

E - *Tu vois ton motif en fait.. le motif ce sont les deux diamants.. donc tu vois au début il est juste avant les deux diamants*

Alix – *Oui*

E – *Donc quand tu vas.. quand tu as fini de parcourir ta boucle.. il faut qu'il soit juste avant les deux diaants qui sont juste au-dessus*

E étaie fortement la prise d'information. **Ce qui est particulier dans cette prise d'information, c'est qu'elle doit être faite à un certain endroit (état en début de boucle), et que la vérification se fait à un autre endroit de la grille (en fin de boucle). on a donc une opération mentale qui est de l'ordre de la translation.**

Alix – *Ah.. d'accord alors donc avancer.. on va faire..* (pp) *avancer* (pp, pp) *avancer* (pp) *avancer* (pp) *tourner à gauche* (pp) *avancer* (pp) *tourner à droite* (pp, pp) *avancer avancer* (pp) *avancer.. oui mais pourquoi ça marche sur les deux premiers mais pas aux autres ?* (parcourt le reste du trajet)

E – *Non ça ne marche pas sur les deux premiers.. c'est parce que tu es allée trop vite.. il faut que tu t'arrête au moment où tu es devant tes deux.. les deux diamants du.. du deuxième étage..*

Alix – *Ok..* (pp)

E – *Stop*

Alix – *Et voilà*

E – *Et là regarde.. regarde bien ce que tu as dans ta boucle.. normalement tu devrais avoir fini ta boucle..*

Alix – *Oui*

E – *Mais là tu n'as pas fini*

Alix – **Ah bon ?**

**Ici est mis en évidence le caractère non évident de l'interprétation du surlignage pour le sujet : intervient-il avant ou après que l'action soit effectuée ? Peut-être qu'une barre entre les deux blocs pour séparer ce qui a été exécuté de ce qui ne l'a pas été serait plus efficace** (même si je conçois bien que cela est peut-être largement plus compliqué du point de vue développement)

E – **Ben regarde.. quelle est la prochaine instruction que.. qui va être exécutée**

Alix – **Euh.. en fait c'est où la boucle.. c'est ici ?** (pas de pointage)

E – **La boucle.. c'est ce qu'il y a à l'intérieur du bloc orange**

Alix – **Ça..** (survole le corps de la boucle de haut en bas) **ah oui**

E – **Mais normalement il devrait avoir tout fait.. et là est-ce qu'il a tout fait ?**

Alix – **Non..**

E – **Non**

Alix – **Donc ça c'est pas bon** (supprime le dernier bloc AV)

E – **Mm**

Alix – **Mm donc je vais essayer comme ça**

Exécution du programme d'id.3655

Alix – **Ah enfin ça marche**

→ validation

E – **Oui mais est-ce que tu as compris pourquoi ça marche ?**

Alix – **Oui**

E – **Hein il faut replacer le robot exactement pareil par rapport au motif**

Alix – **D'accord**

## **Alix, 2021 tour 2**

---

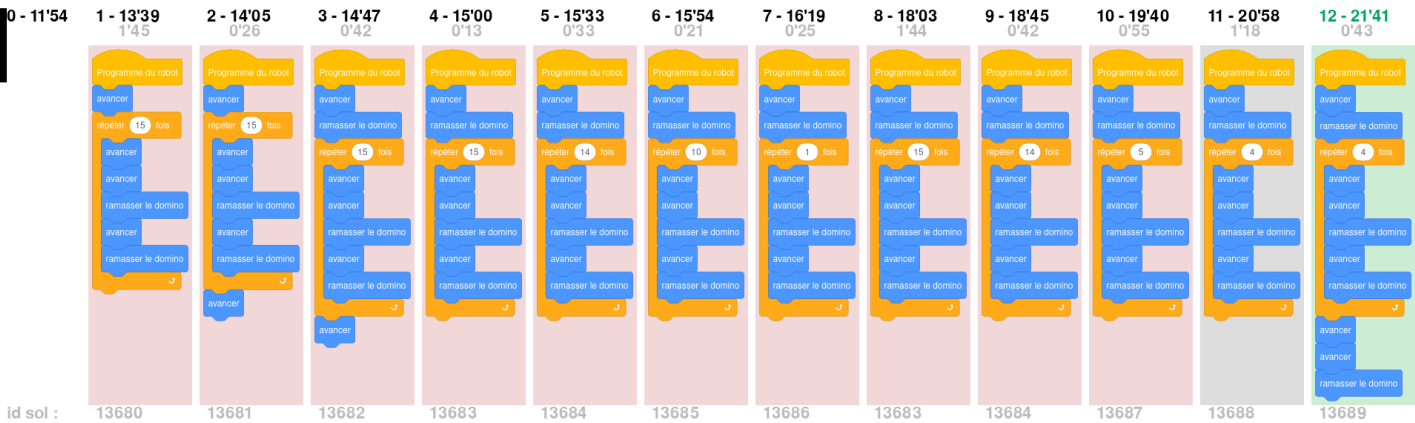
### **Alix, 2021t2p4v3 – profil accommodation**

[lien vers l'extrait vidéo 1](#)


[lien vers l'extrait vidéo 2](#)





[lien vers l'extrait vidéo 3](#)




[lien vers l'extrait vidéo 4](#)





id sol : 13680 13681 13682 13683 13684 13685 13686 13683 13684 13687 13688 13689


	Description	Interprétation
	<p><b>2021t2s4v3_h9tb335h_CM1.mp4</b></p> <p>0'00 (Q) S – <i>Euh.. les dominos avec les soleils, est-ce qu'il faut les ramasser aussi ?</i></p> <p>(ExMT) E – <i>Ben, il faut que tu lises..</i></p> <p>S – <i>Ah oui.. mince</i></p> <p>(8) la souris parcourt l'énoncé</p> <p>→ 552</p> <p>(2) accroche un bloc répéter dans l'éditeur</p> <p>(1p) pointe les dominos un à un</p> <p>(4) renseigne le compteur de boucle à 15</p> <p>(7pg) pointe le premier domino, parcourt doucement les deux dominos suivants, puis plus vite le reste de la ligne</p> <p>(9) place un bloc avancer dans le bloc répéter, l'enlève et le met sur le côté de l'éditeur, détache le bloc répéter pour placer le bloc avancer avant</p> <p>→ 1506</p> <p>(7pg) pointe les 3 premiers dominos</p> <p>(3) place un bloc avancer dans le bloc répéter</p> <p>(7pg) pointe le premier domino, puis les deux dominos bleus suivants (groupés)</p> <p>(3) ajoute les blocs avancer, ramasser le domino, avancer, ramasser le domino dans le bloc répéter</p> <p>(5a) soumission 1 → tous les dominos sont ramassés sauf le premier. Le message « Le robot sort de la grille ! » est affiché.</p>	<p>Le sujet dénombre les cases, de manière automatisée, avant d'avoir considéré le motif → dénombrer les cases fait l'objet d'un schème en lui-même. C'est le schème de création d'unité où l'unité est automatiquement la case.</p> <p>On a là un changement de classe de situation : les éléments à compter ne sont plus en rapport avec le nombre de cases.</p> <p>C'est la remise en cause de ce schème qui pose problème au sujet pour ce problème.</p> <p>Hypothèse : commencer par du déplacement sur grille crée cet obstacle vs commencer par Motif art où dès le départ il n'y a pas correspondance.</p> <p><b>Prend comme repère le premier domino pour l'état initial à l'entrée de la boucle → ajoute un bloc répéter avant la boucle pour que le robot se trouve sur ce domino. C'est l'élément qui est pris comme point de repère et non la position du robot, celle-ci étant ajustée après.</b></p> <p><b>Le premier domino est pris comme point de repère mais n'est pas considéré ensuite dans le codage du motif, c'est le ramassage des deux dominos bleus groupés qui est codé.</b></p> <p>Le motif saillant visuellement est le groupe de deux dominos bleus.</p>

	<p>1'44 (12p) pointe le domino non ramassé (11v) lit en chuchotant le message de feedback (12p) pointe à nouveau le domino non ramassé (Q) S – <b>Il faut qu'il ramasse celui-là aussi ?</b> (ExMT) E – <b>À ton avis ?</b> S – <b>Oui</b> (6) ajoute un bloc avancer après la boucle (5a) soumission 2 → même feedback que la soumission 1</p>	<p>Le message de feedback ne mentionne pas le domino non ramassé ce qui provoque probablement la question du sujet.  Sens de l'ajout du bloc avancer ? Je ne comprends pas car cela ne semble pas en rapport avec le domino restant à ramasser et le sujet a bien remarqué que le robot sort déjà de la grille</p>
	<p>(ExPP) 2'08 E – <b>Tu n'oublies pas que tu as le mode pas à pas aussi.. Tu sais ce que c'est le mode pas à pas ?</b> (Q) S – <b>Euh non.. j'ai oublié</b> (ExGD) E – <b>À côté du bouton play.. à gauche.. à droite du bouton play.. en bas.. voilà.. juste à droite.. non de l'autre côté.. oui</b> (5b) clique deux fois sur le bouton pas à pas, le robot avance sur le premier domino et la boucle est foncée. S – <b>Ah mince</b> (9) détache le programme au niveau du bloc répéter, le met sur le côté, accroche un bloc ramasser le domino après le premier bloc avancer et raccroche le programme en-dessous. (5a) Soumission 3 → tous les dominos bleus sont ramassés. Le message « Le robot sort de la grille ! » est à nouveau affiché.</p>	<p>[demander de changer le visuel du bouton, de mettre le nouveau visuel pour que ce soit plus repérable]  Le sujet repère l'endroit de la première erreur dans le programme et fait un débogage local en intercalant le bloc manquant.</p>
	<p>2'52 S – <b>hop</b> (X6) supprime le dernier bloc avancer, celui qui est hors du bloc répéter (5a) soumission 4 → même feedback</p>	<p>Le sujet prend la manière la plus évidente de raccourcir son programme. Il n'y a pas de procédure de vérification, d'exécution mentale. Le sujet délègue l'exécution à la machine et réagit en fonction du feedback.  Genèse instrumentale – partage des rôles entre l'humain et la machine  2 types de comportement : simulation mentale de l'exécution vs déléguer l'exécution à la machine et réagir en fonction du feedback.  Différer le moment de l'exécution après le clic sur le bouton play pour favoriser la représentation mentale de l'exécution ?</p>
	<p>(ExMT) 3'05 S – <b>Alors.. pourquoi il dépasse ?</b> (ExPP) E – <b>Alors, si tu prends le mode pas à pas ?</b> (5bv) Clique trois fois sur le mode pas à pas S – <b>Avance, ramasse</b> S'arrête lors de l'apparition de la bulle « répétition 1 » pointe la bulle « répétition 1 » (4) change la valeur du compteur de 15 à 14</p>	<p><b>Lorsque le programme est erroné, un mode d'exécution plus lent, déclenché automatiquement, proche du mode pas à pas serait le bienvenu. Ce mode s'adresserait qu'aux novices, il faudrait que ce soit un paramétrage du task.js.</b>  <b>En particulier, ce mode ferait défiler les valeurs du compteur de boucle comme dans le mode pas à pas, et inviterait à relancer l'exécution en mode pas à pas à la fin.</b>  <b>Comment est compris le feedback « répétition 1 » ? la lecture de l'info-bulle est-elle en rapport avec le changement de valeur du compteur ? Le sujet a-t-</b></p>

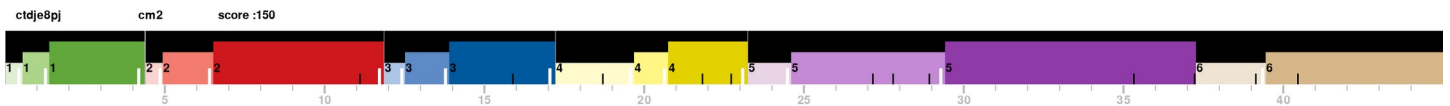
	(5a) soumission 5 → même feedback	<b>il compris qu'il y avait une répétition en trop, dans le sens du langage courant du terme répétition ?</b>
	<p>3'38 (4) change la valeur du compteur de 14 à 10</p> <p>(5a) soumission 6 → même feedback</p>	<p>Est-ce que 10 correspond au nombre de dominos bleus à l'écran ?</p>
	<p>3'57 (Q) S – <b>Mais je ne comprends pas pourquoi il avance</b></p> <p>(ExPP) E – <b>Eh bien essaie avec le mode pas à pas</b></p> <p>(5b) clique deux fois sur le bouton pas à pas, marque un temps d'arrêt</p> <p>clique une fois sur le bouton pas à pas, chuchote</p> <p><b>S – répétition 1</b></p> <p>(4) change la valeur du compteur de 10 à 1</p> <p>(5a) soumission 7 → le robot ramasse les 3 premiers dominos bleus, s'arrête sur la case du 3ème domino bleu. Le message « Le robot n'a pas ramassé les dominos demandés » est affiché.</p>	<p>Le sujet ne comprend pas le feedback « répétition 1 », il tâtonne afin de s'en construire une interprétation.</p>
	<p>4'24 (5bv) clique sur le bouton pas à pas jusqu'à l'apparition du feedback « répétition1 »</p> <p><b>S – Ramasser le domino.. hein..</b></p> <p>continue l'exécution en pas à pas jusqu'à son terme en chuchotant le nom des blocs du corps de la boucle</p> <p>(4) remet la valeur 15 dans le compteur de boucle</p> <p>(5b) exécute en mode pas à pas, s'arrête à l'apparition de la bulle « répétition 1 »</p> <p>(Q) S – <b>Pourquoi il y a marqué répétition une fois ici ?</b></p> <p>(ExGD) E – <b>C'est parce que c'est la première fois qu'il passe dans la boucle.. il compte le nombre de fois qu'il passe dans la boucle.. donc là il te prévient qu'il va passer une première fois dans la boucle..</b></p> <p>(Q) S – <b>Dans la boucle.. c'est quoi ?</b></p> <p>(ExRF) E – <b>Dans.. À l'intérieur du bloc répéter.. donc il va faire une première fois ce que tu as mis à l'intérieur du bloc répéter.. donc continue à cliquer sur le mode pas à pas, tu vas voir</b></p> <p>(5bv) clique sur le bouton pas à pas</p> <p><b>S – Première fois</b></p> <p>(ExRF) E – <b>Vas-y encore tu es toujours dans la première..</b></p> <p>(5bv) clique</p>	<p>Le sujet interprète le 1 de « répétition 1 » comme un cardinal, alors qu'à cet endroit le 1 a une valeur ordinaire</p> <p>→ changer « répétition 1 » en « 1ère exécution des blocs qui sont dans le bloc répéter »</p> <p>le 1 correspond à la valeur de la variable du compteur de boucle, qui contrôle le nombre d'exécution. Dans cet environnement, cette variable est cachée à l'utilisateur, qui n'a pas de représentation du contrôle de la boucle. C'est une limite de la transposition didactique visant à introduire la structure itérative sans mobiliser de variable.</p> <p>Pour le sujet, les « fois » de répéter x fois correspondent aux blocs individuels et non à la séquence de</p>

	<p>S – <b>Deuxième</b></p> <p>(ExRF) E – <b>Tu n’as pas tout fait.. encore</b></p> <p>(5b) exécute en mode pas à pas</p> <p>(ExDE) E – <b>Voilà, là il te dit.. je commence la deuxième fois</b></p> <p>S – <b>ok.. troisième fois.. quatrième.. cinquième..</b></p> <p>(5b) exécute en mode pas à pas jusqu’à son terme : soumission 8</p>	<p>blocs.</p> <p>Le sujet n’a pas compris que le corps de la boucle compte pour une répétition → schème de création mentale d’unité partiellement construit ou alors indépendant du schème d’identification de motif</p> <p>TODO : étudier les fonctions d’étayage de l’expérimentatrice sur ce problème.</p>
	<p>S - <b>donc en fait c’est pas 15..</b></p> <p>(Cl-)ouvre le clavier numérique et le referme</p> <p>S - <b>si ça doit répéter 15 fois ça ( 7ev) parcourt la séquence de haut en bas), alors c’est 14</b></p> <p>(4) change la valeur du compteur de 15 à 14</p> <p>(5b) fait une exécution complète en mode pas à pas : soumission 9</p>	
	<p>(X3) détache les deux derniers blocs de la séquence à l’intérieur du bloc répéter</p> <p>(ExFP) E – <b>Non, tout ça c’est bon</b></p> <p>(3) replace les blocs à leur endroit initial</p> <p>(ExRF) E – <b>Tout ça c’est bon mais il faut que tu trouves combien de fois</b></p> <p>S – <b>Euh..</b></p> <p>(1p) pointe un à un les dominos bleus</p> <p>(4) change la valeur du compteur à 5</p> <p>(5b) fait une exécution complète en mode pas à pas : soumission 10</p>	<p>L’évolution consiste à dénombrer les dominos bleus, ie à prendre les éléments saillants comme unité en faisant abstraction des autres, et donc du pas du robot.</p> <p>Lorsque le sujet indique 5 dans le compteur de boucle, quelle est la prise d’information ? Il vient de dénombrer les dominos bleus. Se réfère-t-il au feedback de l’info-bulle du compteur ?</p>
	<p>7’45 S – <b>Argh.. Alors..</b></p> <p>(4) change la valeur du compteur de 5 à 11</p> <p>(5b) fait une exécution en mode pas à pas, arrête lorsque le robot est sur la dernière case en ayant ramassé le domino.</p> <p>(Q) S – <b>Je ne comprends pas pourquoi il atterrit sur avancer</b></p> <p>(ExDE, ExMT) E – <b>Eh bien, il finit son programme.. mais là tu avais mis 5 fois la fois précédente.. et avec 5 fois il sortait déjà de la grille.. et alors est-ce que avec 11 fois c’est possible ?</b></p> <p>S – <b>Oui..</b></p> <p>(4) change la valeur du compter de 5 à 4</p> <p>(5b) fait une exécution complète en pas à pas. Le robot a ramassé tous les dominos sauf le dernier. Il se trouve sur la case précédant le dernier domino orange – soumission 11</p>	



	<p>(Q) S – <i>Si 5 fois ça dépassait, normalement 4 fois ça devrait marcher..</i></p> <p>(12p) pointe le dernier domino bleu, non ramassé</p> <p>(ExDE, ExMT) E – <i>Ben là.. 4 fois il arrive là.. donc qu'est-ce qu'il reste à faire maintenant ?</i></p> <p>S – <i>Ben..</i></p> <p>(6) place un bloc avancer sous la boucle</p> <p>(7pg) pointe le dernier domino orange</p> <p>(6) ajoute un bloc avancer et un bloc ramasser le domino en fin de programme</p> <p>(5a) soumission 13 valide</p>	
--	---	--

## Alix, 2022 tour 2



### Alix, 2022t2p1v3 – profil expert

[lien vers l'extrait vidéo](#)

### Alix, 2022t2p4v1- profil accommodation

[lien vers l'extrait vidéo](#)

ctdje8pj cm2 score global : 150 temps moyen par bloc : 9.1  
 4 - Chauffer le château [2022-2 white] basic durée : 2'23 ind. rapidité normalisée : 0.75

0 - 17'14    1 - 18'44    2 - 19'37  
                   1'30                   0'53

Programme du robot

avancer

ramasser le bois

répéter 12 fois

avancer

construire une plateforme devant

Programme du robot

avancer

ramasser le bois

répéter 12 fois

construire une plateforme devant

avancer

avancer

déposer le bois

id sol :           2212                   225

Alix lit la consigne à voix haute. Le ton employé lors de la lecture de la deuxième phrase indique un défaut de compréhension.

Temps de latence de 5s

Alix – *Ok.. euh donc là on va avancer une fois.. (ajAV) pourquoi les.. non c'est bon..* (pointe la case devant le bois) *donc.. mm.. ramasser le bois* (ajRAM, pointe le bois) *ramasser le bois.. après on va mettre*



ajREP, repAV

Alix – **Combien de fois ?**

pointe les cases en dénombrant tout haut, recompte une deuxième fois tout haut avec la souris immobile

Alix – **Douze fois..** (comp 12) **douze fois.. avancer construire une plateforme dev..** (repCST) **Marielle ?**

E – **Oui**

Alix – **Quand ils disent construire une plateforme devant est-ce que je dois mettre avancer ?**

E – **Ah**

Alix – **Et donc ça avance**

Alix formule l'hypothèse que l'action d'avancer est prise en charge par le bloc construire une plateforme devant, comme pour le bloc pousser la caisse. Nous avons deux blocs dont la règle de fonctionnement est différente. dans un cas, le déplacement du robot est pris en charge par le bloc d'action, alors que dans l'autre cas, le déplacement n'est pas géré par le bloc d'action. Le fait que ce n'est pas la même règle selon le contexte constitue probablement une difficulté pour les utilisateurs.

E – **Eh bien écoute tu as le mode pas à pas pour te rendre compte de ce que le robot fait quand tu mets un bloc**

Alix – **Donc je vais essayer** (exécute en mode pas à pas jusqu'au feedback)

L'exécution correspond clairement à une stratégie pour construire une représentation de l'exécution associée au bloc construire une plateforme devant.

Alix – **Le robot va se jeter dans le vide.. ah donc non..** (enlève le bloc avancer du corps de la boucle et commence une exécution en mode pas à pas jusqu'à ce que le robot ait construit la première plateforme, exécution non enregistrée) **ok donc non..** (replace le bloc avancer après le bloc CST et exécute à nouveau en mode pas à pas, non enregistré) **donc là..** (stoppe après 3 plateformes construites) **ok donc là c'est bon.. donc là avancer 12 fois et après avancer.. déposer le bois.. on va essayer**

validation

**Alix, 2022t2p5v2 – profil accommodation**

[lien vers l'extrait vidéo](#)

ctdje8pj cm2

score global : 150

temps moyen par bloc : 9.1

5 - Distribuer le bois [2022-2 white] easy

durée : 4'44

ind. rapidité normalisée : 1.84

0 - 24'36

1 - 27'11  
2'35

2 - 27'49  
0'38

3 - 28'57  
1'08

4 - 29'20  
0'23

id sol :

9888

9889

5823

236

lit l'énoncé à voix haute

Alix – **Donc.. donc là ça va être quasiment la même chose**

Cette remarque indique qu'Alix classe cette situation dans la même catégorie que la précédente.

aj REP

suit le parcours du robot avec le pointeur en marquant et énonçant un nombre à la fin de chaque ligne, donc jusque 6

Alix – **Six fois.. on va essayer** (comp 6)

Alix – **Ok.. donc là ça va être avancer** (repAV, survole bièvement la grille) **avancer** (repAV, survole bièvement la grille) **tourner à droite** (repTD) **avancer** (repAV) **et après tourner à gauche** (repTG) **alors là on va faire mode pas à pas**

exécution du programme d'id.235 (programme non valide le plus courant, qui est un programma partiel) en mode pas à pas (non enregistré)

s'arrête lors de la dernière itération lorsque TD est surligné

Alix – **Tourner à droite** (pp pp) **tourner à gauche**

**Le bloc répéter est surligné. C'est ce qui induit probablement Alix en erreur, elle pense que le programme n'est pas fini, qu'il y a encore une itération à venir. Dans ce cas de figure, ce qui est prévu pour être une aide induit l'utilisateur en erreur.**

Alix – **Euh non, on va essayer cinq fois** (comp 5) **donc là arrivé ici ce sera cinq fois (pointe la case au-dessus du tas de bois).**

**On voit ici clairement une erreur de représentation de la position du robot après exécution de la boucle. L'erreur est cohérente avec le pointage lors du dénombrement.**

Alix édite la suite de son programme en prenant cette case comme référence. Elle fait une erreur de sens de rotation du robot, qu'elle corrige spontanément par rapport à ce point de repère. Elle ne considère pas du tout le bloc de rotation pourtant adéquatement placé en fin de boucle.

Alix – **Donc là j’vais mettre tourner à gauche** (aj TG) **avancer** (aj AV, simule le déplacement sur la grille) **ramasser le bois** (ajRAM) **donc tourner donc après ce sera.. j’ai mis tourner à gauche.. donc euh.. non c’est tourner à droite** (remplace TG par TD) **tourner à droite avancer ramasser le bois.. tourner à gauche** (aj TG) **avancer** (ajAV) **et après donner le bois au castor** (ajDEP)

exécution du programme d’id.9888

lit le feedback à voix haute

Alix – **Ah donc non c’est six fois** (comp 6) **six fois**

décroche la séquence qui se trouve après la boucle

Alix – **Et ça je vais mon..**

exécution du programme d’id.9889

**Le fait qu’Alix détache les blocs après la boucle montre que cette exécution est une procédure de vérification du positionnement du robot**

lit le feedback à voix haute

Alix – **Donc..**

ferme le feedback

reprend l’exécution en mode pas à pas

Comme lors de la première fois, elle s’arrête lorsque le bloc répéter est surligné, ce qui laisse supposer que le programme n’est pas fini d’exécuter, le feedback n’étant pas retourné. Cela provoque une réaction verbale d’Alix

Alix – **Euh mais donc là c’est.. ça a été répété que cinq fois ou six fois ?**

E – **Ben six fois**

Alix – **Donc six fois.. donc après tourner à gauche.. on doit mettre ramasser** (ajRAM, pop-up trop de blocs) **ah mais donc j’ai plus de blocs.. non mais donc ça** (supprime la séquence restée dans l’éditeur) **poubelle.. ramasser le bois avancer** (prend avancer mais ne relâche pas le bloc tout à fait assez loin et il disparaît)

Alix ne s’en rend pas compte tout de suite, ce qui provoque l’erreur du programme

Alix – **Après.. donner le bois au castor** (ajDEP)

exécution du programme d’id.5823

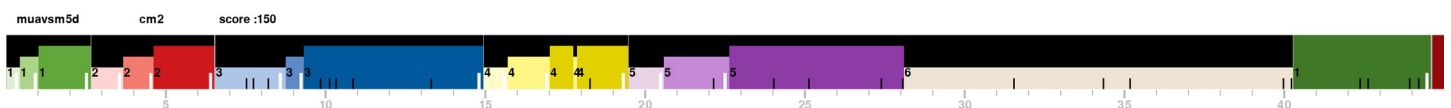
lit le feedback à voix haute

détache DEP, aj AV, remet DEP à la suite

exécution du programme d’id.236 → validation

Dans cette confrontation, les schèmes sont mobilisés mais ils sont perturbés par le fonctionnement de l’EIAH.

## Alix, 2022 tour 3



## Alix, 2022t3p3v2 – profil expert

[lien vers l'extrait vidéo](#)

## **Alix, 2022t3p4v1 – profil expert**

[lien vers l'extrait vidéo](#)

## **Alix, 2022t3p4v3 – profil accommodation**

[lien vers l'extrait vidéo](#)

muavsm5d cm2

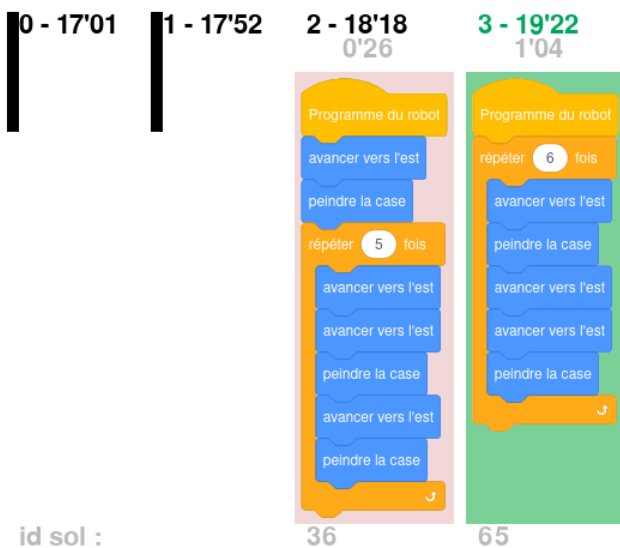
score global : 150

temps moyen par bloc : 7.4

4 - Peindre le dessin [2022-3 white]

medium durée : 2'14

ind. rapidité normalisée : 1.26



Alix – *Heu.. mm.. là c'est les mêmes* (pointe les deux premiers groupes de deux blocs) *heu donc j'fais.. est* (ajEST) ça (pointe le robot, le premier point) *donc là ce sera avancer vers l'est* (ajREP) *et là ce sera pareil..*

pointe les groupes de deux blocs en chuchotant jusque cinq

Alix – *Cinq fois* (comp 5) *cinq.. fois.. avancer* (repEST) *avancer* (repEST) *peindre une case* (repPC) *avancer* (repEST) *il me reste deux blocs* (chuchoté) *heu.. mm..*

→ On sent que Alix se rend compte du problème à ce moment, mais elle va quand même au bout de sa première idée.

retourne sur la version deux étoiles

Alix – *J'vais juste voir un truc*

survole le corps de la boucle de haut en bas

Alix – *Avancer vers l'est.. avancer vers l'est.. peindre une case.. avancer vers l'est.. peindre une case..* (recommence le survol en énonçant le blocs à voix haute)

→ prend comme référence sa solution à la version précédente

retourne sur la version trois étoiles

Alix – *Une case* (chuchoté) *peindre une case* (repPC)

pointe le deuxième bloc du dernier groupe de deux, puis la case vide suivante → représentation de la situation à l'issue de l'exécution de la boucle, représentation mentale de l'exécution, anticipe mentalement l'exécution

Alix – **heu non.. je me suis trompée..**

pointe le 1<sup>er</sup> bloc du 1<sup>er</sup> groupe, le 1<sup>er</sup> bloc séparé visuellement, intercale un bloc PC avant la boucle

→ la représentation mentale de l'exécution amène Alix à se rendre compte qu'elle a oublié une action avant la boucle.

exécution du programme d'id. 36

Alix – **Ah non** (pendant l'exécution)

→ le dernier point n'est pas peint, feedback : Votre robot n'a pas peint les bonnes cases.

Alix – **Ah oui non.. ah là je sais p..** (pas inaudible) **ah mince.. heu.. qu'est-ce que je peux faire ?** (chuchoté)

temps de latence : 7s

Alix – **Ah je crois que j'ai trouvé..** (commence chuchoté et finit tout haut)

modulation du volume, se parle à elle-même

ferme le feedback, remet en situation initiale

Alix – **Ouais.. j'veis encore tout jeter** (vide l'éditeur)

pointe le premier point, puis furtivement la limite entre les deux cases du premier groupe de points, ce qui révèle la nouvelle idée. Dans ce cas, le pointage est plus précis que pour le groupe de deux points.

Alix – **Donc là ce sera..**

ajREP

Alix – **Combien de fois ?**

Compte tout haut jusque 6 en ralentissant juste après le deuxième point de chaque motif

→ signe qu'elle a repéré la limite entre deux motifs.

comp 6

Alix - **Et donc là ce sera.. six fois** (repEST) **avancer vers l'est peindre une case** (repPC, survole furtivement la grille) **avancer** (repEST) **avancer** (repEST) et **peindre une..** (repPC)

exécute le programme d'id.65 → validation

### **Alix, 2022t3p5v1 – profil expert**

[lien vers l'extrait vidéo](#)

Noté pendant la session – verbalisé : « ce sera comme ça » place le bloc répéter d'emblée ; expert

—

lit l'énoncé à voix haute, pointeur qui suit approximativement le texte

survole la grille

Alix – **Ah oui on l'avait déjà fait je crois**

simule le parcours sur la grille en énonçant les actions à voix haute

Alix – *On retombe toujours au même endroit*

→ explicitation de la jonction

Alix – *Ce sera comme ça* (ajREP)

simule le parcours du robot en énonçant un nombre lorsque le pointeur passe sur un objet

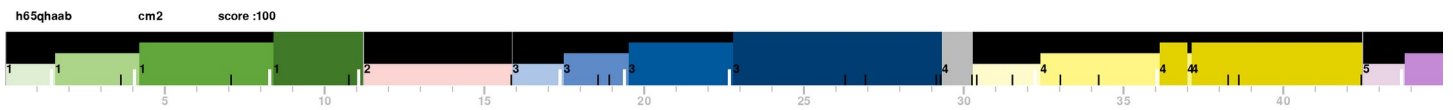
Alix – *Cinq petites fois* (comp 5)

survole la grille devant le robot

édite le motif en énonçant les blocs à voix haute

validation avec le programme d'id.86

## Alix, 2022 expérimentation complémentaire



### Alix, 2022t4p1v2 - profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

h65qhaab cm2

score global : 100

temps moyen par bloc : 9.7

1 - Planter des fleurs [uLille v2] easy

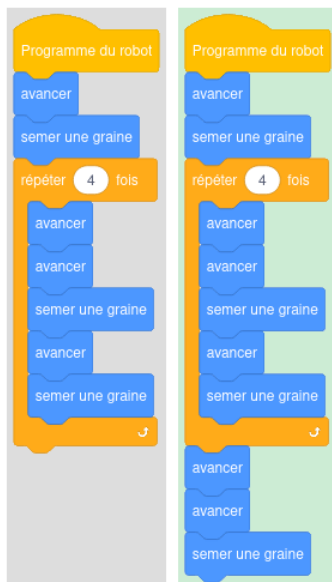
durée : 2'32

ind. rapidité normalisée : 0.96

0 - 1'34

1 - 3'39  
2'05

2 - 4'06  
0'27



id sol : 13631

3398

Sujet 5 - verbalisé : s'interroge sur le bloc de répéter jusqu'à, chercher à construire un schème d'usage. première et dernière zone de terre traitées hors de la boucle

scroll pour voir les blocs dans le bas de la réserve

Alix – *Ah j’comprends pas trop le bloc répéter jusqu’à*

E – *Oui*

Alix – *Donc je sais pas.. heu..*

E – *T’es pas obligée de l’utiliser.. il est là..*

Alix – *Ah oui c’est vrai.. parce que on n’a pas une limite de blocs..*

E – *T’as pas de limite de blocs.. tu vois t’as rien en haut à droite.. donc tu utilises c’que tu veux*

Alix – *Oui mais.. oui mais peut-être que ce sera plus.. enfin plus.. pas plus facile mais plus utile de l’utiliser ou.. enfin voilà.. donc.. (ajAV) mm.. (ajSG) semer une graine (chuchoté) il est là (pointe le premier tas de terre) avance (chuchoté, pointe la case vide suivante, pointe les deux zones de terre suivantes, la case vide suivante)*

E - *Tu peux continuer à penser tout haut*

ajREP

E – *C’est bien*

ouvre le clavier numérique

Alix – *Donc là.. donc.. on note ça (pointe le 2ème groupe de tas de terre) et on doit répéter (pointe le tas de terre et compte à voix basse jusque quatre, pointe le dernier tas de terre) on va essayer quatre fois (comp 4) on va voir si ça marche pas en fait.. donc on est (pointe le premier tas de terre) et là on fait avancer (repAV) avancer (repAV, pointe le dernier tas de terre, le premier, le deuxième) avancer avancer semer une graine (repSG) avancer (repAV) semer une graine (repSG) donc on est là (pointe le 3ème tas de terre) et voilà.. donc on va essayer..*

exécution du programme d’id.13631 → le robot n’a pas semé sur le dernier tas de terre, feedback : Il reste de la terre vide de fleur

Alix – *Ok*

ferme le feedback

Alix – *On avance (ajAV) on avance (ajAV) et on sème (ajSG)*

exécution du programme d’id.3398 → validation

**Alix, 2022t4p1v4 – profil accommodation**

[lien vers l’extrait vidéo](#)

h65qhaab cm2

score global : 100

temps moyen par bloc : 9.7

1 - Planter des fleurs [uLille v2] hard

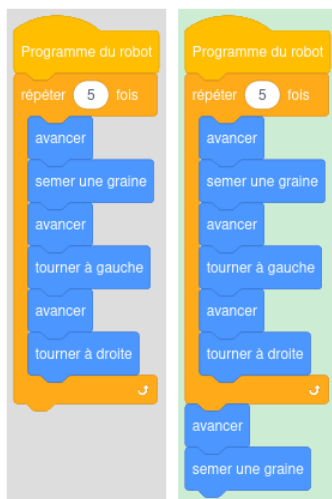
durée : 2'43

ind. rapidité normalisée : 0.87

0 - 8'24

1 - 10'47  
2'23

2 - 11'07  
0'20



verbalisation

procédure de vérification de la sortie de boucle

intéressant à transcrire

voir si on retrouve le pointage sur la **visualisation des mouvements de souris**

—

Alix – *En fait directement la version..*

E – *Oui.. mais tu peux toujours.. ça c'est le traitement.. enfin c'est ce qui est prévu.. parcours prévu.. maintenant tu peux toujours faire retour et..*

Alix \_ *Oui*

E – *Si tu crois que c'est trop difficile.. et puis en faire un autre.. c'est toujours*

Alix – *Oui mais je crois que ça on l'a déjà fait donc..* (pointe les deux zones de terre du bas) *je vais essayer et puis après..*

pointe le bloc répéter dans la réserve

Alix – *Heu.. on va faire ça déjà..* (ajREP) *voilà donc là ce sera* (compte tout haut jusque 5 en parcourant le trajet planifié pour le robot et en ralentissant sur les cases vides avant les zones de terre.)

comp 5

**Alix prend comme repère la case vide avant la prochaine zone de terre, ce qui a pour conséquence qu'elle ne compte que jusque cinq et non six, car son point de repère est manquant pour la dernière itération.**

Alix – *Cinq.. et donc là on va faire.. hop mm.. avancer* (repAV, mouvement du pointeur entre le robot et la première zone de terre) *semer une graine* (repSG, pointage de la première zone de terre, mouvement bref vers la case vide à sa gauche) *avancer* (repAV) *tourner à droite* (repTD) *avancer* (repAV, temp s d'attente) *heu qu'est-ce que je fais moi.. non c'est tourner à gauche..* (décroche les deux derniers blocs) *j'veins mettre.. donc là tourner à gauche* (repTG, repAV resté dans l'éditeur) *avancer.. avancer tourner à gauche avancer tourner à droite* (le pointeur est immobile pendant la simulation de l'exécution, survole la réserve) *donc là tourner à gauche* (pointe la case vide à gauche du premier tas de terre) *avancer* (dé-



place le pointeur sur la case vide en-dessous) **donc tourner à droite..** (repTD) **et c'est bon .. y'en avait déjà un** (supprime le bloc TD laissé dans la réserve) **mm..** (vurvole la grille) **et donc on est bon.. ça ça va faire six fois** (parcourt la diagonale de tas de terre rapidement)

→ **comparer les deux mouvements de souris : celui du début avec les temps d'arrêt et celui de la fin englobant. Dans un cas, nous avons une simulation mentale de l'exécution, dans l'autre cas, le motif est compté globalement. Il faudrait trouver une occurrence où c'est plus net pour le deuxième cas, avec une trajectoire rectiligne alors que le parcours est en diagonale.**

Alix – **Et ça va arriver..** (pointe la case vide à droite de la 5ème zone de terre, fait le parcours du 5ème motif, pointe la case avant le 6ème motif) **là.. on va voir**

exécution du programme d'id.13634 (le pointeur reste immobile sur le bouton pendant l'exécution)



Alix – **Oui.. c'est bien ça.. donc là il me manque juste avancer** (déplace le pointeur de l'emplacement du robot à l'issue de l'exécution jusqu'à la zone de terre suivante) **semmer une graine**

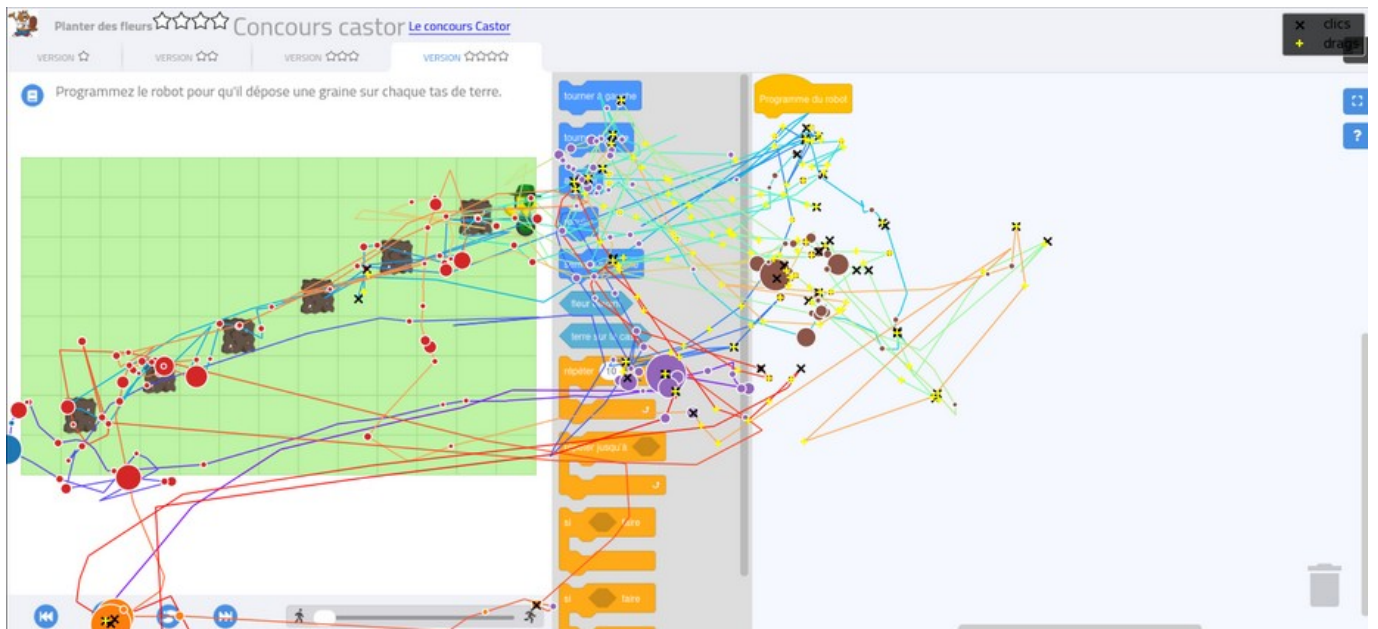
la verbalisation nous montre que l'exécution est une procédure de vérification

Alix – **Avancer** (repAV) **semmer une graine** (repSG) **et je crois que c'est bon**

exécution du programme d'id.13635 → validation

Alix – **Voilà**

E - **Oui**



**Alix, 2022t4p3v1 – profil indéfini**

[lien vers l'extrait vidéo](#)

# Ugo, CM1 en 2022

## Ugo, 2022 tour 1



### Ugo, 2022t1p1v1 – profil indéfini

[lien vers l'extrait vidéo](#)

### Ugo, 2022t1p2v1 – profil accommodation

[lien vers l'extrait vidéo](#)

d9h533hx cm1

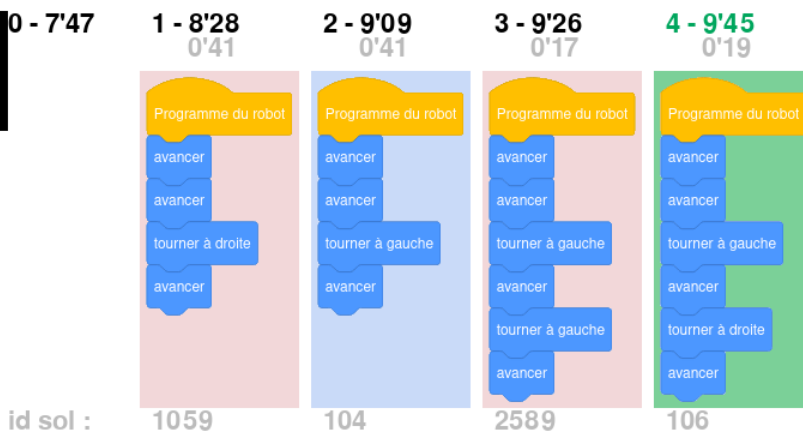
score global : 110

temps moyen par bloc : 12.6

2 - Collecter les pierres [2022-1 white] basic

durée : 1'58

ind. rapidité normalisée : 0.56



### Ugo, 2022t1p4v2 – profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

d9h533hx cm1

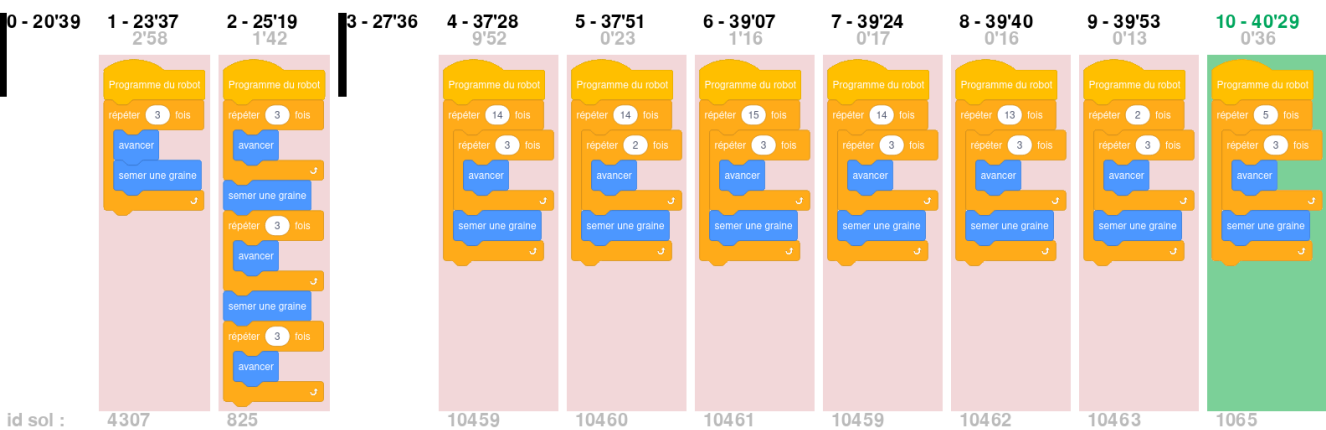
score global : 110

temps moyen par bloc : 12.6

4 - Planter des fleurs [2022-1 white] easy

durée : 17'33

ind. rapidité normalisée : 10.94





#### 2022t1s4v2\_d9h533hx\_CM1.mp4

0'00 parcourt doucement la grille en s'arrêtant sur les zones de terre

0'09 S – **Quinze**

0'10 comp 15 dans la réserve

0'16 S – **Mm.. non**

0'19 pr-rel AV

0'20 S – **Euh.. mince**

0'21 aj AV,AV

0'30 pr-rel AV

0'31 E – **Qu'est-ce que tu as compté pour mettre quinze dans le bloc répéter ?**

0'34 aj AV

0'39 S - **Euf.. mm.. je me suis trompé**

0'41 ouvre comp

0'44 S – **J'vais remettre dix**

0'49 comp 10

0'50 E – **Mais qu'est-ce que tu avais compté pour mettre quinze ?**

0'51 S – **Tous les carrés dans la largeur.. mais au bout d'un moment je me suis demandé comment je vais faire pour planter une graine**

1'00 E – **Ah oui**

1'01 S – **Il va avancer.. ah oui.. maintenant..**

1'06 aj SG (verb)

1'10 comp 3

1'19 pt AV

1'21 aj REP

1'23 lat 7s

1'30 rep AV

1'35 aj SG

1'38 S – **J'vais le mettre de côté** (détache REP,SG)

1'39 sup début du prog

1'44 aj REP

1'47 rep1 AV

1'54 aj SG



1'57 accroche en-dessous REP,SG resté dans l'éditeur

1'58 survole rep2 (chuchotement inaudible)

2'02 aj Rep

2'05 rep AV, 0bloc

2'09 S – **Ah mince** (pt message 0 bloc)

	<p>2'11 E – <b>Alors qu'est-ce qui se passe ?</b></p> <p>2'13 S – <b>Zéro bloc restant autorisé sur huit</b> (pt)</p> <p>2'17 E – <b>Alors ça veut dire quoi ?</b></p> <p>2'20 S – <b>Que je ne peux plus en poser des blocs</b></p> <p>2'22 E – <b>Voilà c'est ça.. que tu as seulement huit blocs pour construire ton programme.</b></p> <p>2'28 vide l'éditeur</p>
	<p>2'30 S – <b>J'vais..</b></p> <p>2'33 fait des allers-retours entre les blocs AV ET SG de la réserve</p> <p>2'37 tente de placer un bloc AV dans le REP de la réserve → le bloc disparaît lorsqu'il est lâché</p> <p>2'40 S – <b>Et mince</b></p> <p>2'41 aj REP (comp déjà à 3)</p> <p>2'43 aj AV,SG</p> <p>2'47 S – <b>Je vais voir si..</b></p> <p>2'49 pp 1 fois</p> <p>2'53 soum 1 → « Votre robot a semé hors des zones de terre. »</p> <p>2'55 E – <b>Sur le mode pas à pas.. c'était une bonne idée mais il faut cliquer plusieurs fois dessus..</b></p>
	<p>3'00 ferme feedback</p> <p>3'01 E - <b>..parce qu'il te montre d'abord les blocs qu'il exécute</b></p> <p>3'03 pp première itération</p> <p>3'06 S – <b>Ah d'accord.. ok c'est par là</b></p> <p>3'12 déplace SG après REP</p> <p>3'20 pt robot, T1, T2, mvt, T1</p> <p>3'25 aj REP</p> <p>3'32 rep2 AV</p> <p>3'36 pt T2</p> <p>3'39 aj REP (difficultés de manip)</p> <p>3'51 rep3 AV</p> <p>3'59 S – <b>Bon..</b></p> <p>4'02 aj SG, 0bloc</p> <p>4'09 pp jusqu'à robot en T3</p> <p>4'18 S - <b>Oups</b></p>



4'20 déplace SG2 avant REP3

4'24 soum 2 pp → « Il reste de la terre vide de fleur ! »

4'39 S – **Ah..**

4'41 E – **On voit que c'est un début de programme qui serait bon..**

4'44 sup REP3(AV)

4'45 E - .. **s'il y avait plus de blocs.. mais là il faut trouver une autre manière**

4'47 vide l'éditeur

4'50 **hmm** (mvt).. **comment faire.. hmm**

5'07 E – **Alors on te dit dans l'énoncé.. si vous êtes bloqué..**

5'14 pt énoncé

5'18 S – **Retournez dans la version 1 et cliquez sur plus de détails**

5'22 E – **Oui**

5'23 S – **Mais comment on fait pour retourner à la version 1 ?**

5'27 E - **Tu cliques sur la ..**

5'29 S – **Ah là..** (pt 1 étoile)

5'30 E – **Voilà**

5'31 Ouvre version 1 étoile

5'32 Ouvre tutoriel

5'33 parcourt tutoriel avec la souris

5'39 E – **Alors tu avais réussi.. mais est-ce que tu as bien vu.. la stratégie ?**

5'45 mvt, lat 5s

5'54 E – **Et si tu descends un petit peu.. ah oui.. non on te dit pas grand-chose de plus.. mais c'est quoi la stratégie ? Est-ce que tu peux expliquer ?**

6'01 S – **Que le bloc répéter.. autant de fois que tu veux.. te permet d'économiser des blocs ?.. mais c'est justement ce que je faisais.. mais comme il n'y en a pas à chaque.. y'a pas un bloc de terre à chaque.. les uns à côté des autres..**

→ **envisage toujours la case comme unité, a conscience de ne plus être dans la même classe de situation, de la rupture dans le traitement de la situation**

6'23 E – **Mm.. donc ça ça marchait quand il y avait un bloc de terre les uns à côté des autres.. qu'est-ce que tu mettais dans le bloc répéter ?**

6'42 S – **Répéter.. euh.. répéter.. je ne sais pas combien de fois.. le.. répéter.. semer une graine.. répéter semer une graine et avancer**

6'48 ferme tutoriel

6'49 E – **Voilà.. mais quand les blocs de terre ne sont plus les uns à côté des autres..**

6'58 retourne sur la version 2 étoiles

7'00 E – **Qu'est-ce qui change ?**

7'02 S – **Ben on peut.. ça nous permet quand même d'économiser mais moins..**

7'08 E – **Mm.. mais peut-être il faudrait trouver une autre manière d'économiser.**

7'13 lat 11s

7'24 S – **Mm.. en fait je suis bloqué**

7'29 E – **Oui.. je vois.. et c'est pour ça que j'essaie de discuter avec toi.. je vois que tu es bloqué.. et alors une autre manière d'économiser.. alors je vais t'aider un petit peu.. qu'est-ce qui se répète sur le dessin ?.. sur la grille du robot ?**

7'47 S – **Et..**

7'50 pt robot, T1

7'52 E – **Qu'est-ce que tu vois qui est pareil ?**

7'55 lat 9s

8'04 E – **Ah.. tu m'entends ?.. je n'ai plus l'impression que la connexion..**

8'08 lat 30s

8'38 déconnexion du sujet

[TO DO : morceau à couper]

11'44 reconnexion du sujet

11'45 aj REP

11'48 rep AV

11'49 E – **Bon tu m'entends [prénom] ?**

11'51 S – **Oui oui**

11'53 E – **D'accord.. donc tu as essayé des choses pendant la coupure ?**

11'57 aj SG

11'58 S – **Euh ben.. mon père il est arrivé quasiment tout de suite**

12'02 E – **D'accord**

[échange avec le père]

12'12 aj REP

12'13 **Et ça ça marche pas** (pr-rel AV, sup rep2)

12'15 E – **Et ça ça marche pas.. et donc j'étais en train de te demander.. mais je ne sais pas si tu as entendu parce que ça a coupé juste à ce moment là.. je te demandais de bien regarder sur le dessin.. sur la grille.. et de me dire ce qui est pareil sur la grille**

12'27 S- **Ben les blocs de terre** (pt).. **ils sont réguliers ..**

12'32 E – **Voilà.. exactement.. tu as remarqué exactement ce qu'il fallait.. les blocs de terre ils sont réguliers.. donc ça c'est ce qui est répété sur la grille.. donc qu'est-ce qu'on va.. qu'est-ce que le robot il va faire pareil à chaque fois ?.. pour ramasser un bloc de terre**

12'51 S – **Non planter une graine !**

12'43 E – **Oui.. pardon.. planter une graine.. excuse-moi.. qu'est-ce que le robot il va faire pareil pour semer une graine ?**

12'59 S – **Euh.. ben le même parcours**

13'05 E – **Oui.. et c'est quoi ce parcours avec des blocs ?**

13'09 S – **Euh.. y'en a deux entre..** (parcourt entre robot et T1) **entre chaque bloc de terre** (parcourt entre T1 et T2)

13'18 E – **Mm.. alors pour aller ramasser la première.. pour aller semer la première graine.. qu'est-ce qu'il faut faire ?**

13'27 S – ***Euuf.. la première.. ben il va avancer de trois blocs et il va s'arrêter pour semer une graine.. ensuite il va faire pareil***

13'42 E – ***Voilà.. donc quand tu dis il va faire pareil..***

13'44 S – ***je sais !***

13'46 E – ***Ah***

13'49 S – ***Euh*** (ouvre comp dans réserve).. ***hm je sais pas si ça va marcher..*** (comp 15)

13'56 E – ***Ah oui mais tu peux essayer***

13'59 pr REP (15) et le déplace au-dessus du prog

14'02 S- ***Euh.. je vais les mettre tous dans.. mais non je pense pas que ça va..***

14'07 E – ***Si si.. ça marche***

14'08 S – ***Non mais*** (continue à déplacer REP 15 au-dessus du prog → ombres à différents endroits dont englobe tout)

14'09 S – ***Ça..*** (entoure SG)

14'10 E – ***Si.. reprends***

14'15 S – ***Attends..*** (difficultés de manip)

14'20 E – ***Je crois que tu as compris mais..*** (continue à manipuler les blocs répéter)

Finale on a REP 15 (REP 3 (AV,SG))

14'29 S – ***Je sais pas si ça va marcher***

14'31 E – ***Alors.. t'as vu tu peux les mettre l'un à l'intérieur de l'autre***

14'32 pp

14'37 E – ***C'est ce que tu voulais voir.. maintenant je suis pas sûre que tu aies replacé tes blocs comme tu voulais***

→ le manque de dextérité entrave le passage de la boucle à laboucle imbriquée

14'44 pp robot arrivé sur T3 avec toutes les cases avec une fleur

14'45 S – ***Je voulais les placer comme ça mais ça les met tous.. ça en met sur tout son parcours***

14'53 E – ***Alors..***

14'55 S – ***Je sais..***



14'56 ouvre comp

15'00 E – ***Je te propose de prendre le mode pas à pas pour ça***

15'02 S – ***C'est quoi le mode pas à pas ?***

15'03 comp 14

15'04 E – ***C'est le bouton avec les deux petits pas***

15'07 S – ***Oui c'est justement ce que je faisais***

15'10 S – ***Regarde*** (pp)

15'12 E – ***Mm***




15'14 S – ***Ah.. mais ça en met partout***

15'17 E – ***Et ça en met partout.. pourquoi ça en met partout ?***

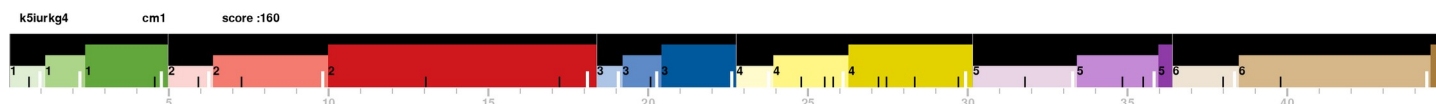
15'20 S – ***Parce que c'est le*** (pt SG1).. ***parce que.. ah je sais..***

	<p>15'26 déplace SG après REP1</p> <p>15'30 S – <b>Hein..</b> (pr-rel prog)</p> <p>15'33 détache rep2(AV)</p> <p>15'36 sup rep2(AV)</p> <p>15'38 rep AV</p> <p>15'40 mvt</p> <p>15'50 S - <b>Alors.. mais là il va planter qu'une graine</b></p> <p>15'53 E – <b>Oui.. mais tu y étais presque je crois</b></p> <p>15'59 pr-rel REP</p>
	<p>16'01 E – <b>Pour planter une graine.. tu as dit qu'il fallait qu'il fasse..</b></p> <p>16'03 comp 3 dans la réserve</p> <p>16'06 <b>Euh.. semer une graine</b> (déplace REP 3 au-dessus du prog).. <b>qu'il doit..</b> (entoure AV avec REP 3)</p> <p>16'14 E – <b>Oui</b></p> <p>16'16 S – <b>Je sais..</b> (déplace SG dans rep1 après rep imbriqué).. <b>je sais pas si ça va marcher.. on va voir</b></p> <p>16'26 pp jusqu'à ce que les cinq zones de terre aient une fleur</p> <p>16'44 bouton play → soum 3 → « Le robot sort de la grille »</p>
	<p>16'50 E – <b>Alors qu'est-ce qui se passe cette fois-ci ?</b></p> <p>16'52 survole prog, pt comp rep imbriqué</p> <p>16'54 comp 2 pour rep imbriqué</p> <p>17'02 S – <b>On va voir</b></p> <p>17'03 soum 4 → S – Quoi ?.. Non (deux premières fleurs hors des zones de terre).. raté → « Le robot sort de la grille »</p>
	<p>17'13 S – <b>C'est trois ici</b> (remet comp 3 pour rep imbriqué)</p> <p>17'22 S – <b>Je pense avec douze</b> (comp 12 pour rep1).. <b>ça..</b></p> <p>17'25 pp</p> <p>17'26 E – <b>Alors qu'est-ce qui te fait mettre douze ?</b></p> <p>17'30 S – <b>Bah.. je sais pas</b></p> <p>17'31 E – <b>Ah.. comment tu as réfléchi pour mettre douze ?</b></p> <p>17'34 S – <b>Ben.. je sais pas.. j'me suis dit quand.. non.. attends.. non.. attends.. un deux trois</b> (continue à compter de plus en plus bas puis à nouveau tout haut, jusque 15)</p> <p>17'48 E – <b>Quinze ?.. C'est quoi ça quinze ?</b></p> <p>17'51 S – <b>Attends.. ben quinze c'est le nombre jusqu'au dernier pot de terre</b></p> <p>17'58 E – <b>Mm</b></p> <p>18'03 comp 15</p> <p>18'10 S – <b>Quinze c'est jusque là</b> (pt).. <b>jusqu'au</b></p> <p>18'13 E – <b>D'accord.. c'est le nombre de cases</b></p>



	<p>18'18 S – <i>Jusqu'au dernier pot de terre</i></p> <p>18'19 soum 5 → « Le robot sort de la grille »</p>
	<p>18'22 E – <i>C'est le nombre de cases.. oui jusqu'au dernier pot de terre si tu veux</i></p> <p>18'26 S - <i>Et là il sort de la grille</i></p> <p>18'27 E – <i>Et là il sort de la grille</i></p> <p>18'30 S – <i>Et avec 14 c'est pareil je crois</i> (comp 14 pour rep 1)</p> <p>18'38 E – <i>Au cas où tu aurais mal compté..</i></p> <p>18'38 soum 6 → « Le robot sort de la grille »</p>
	<p>18'40 E - <i>..le nombre de cases.. c'est ça ?</i></p> <p>18'41 S – <i>Oui</i></p> <p>18'46 S – <i>Et treize je sais pas si ça marche</i></p> <p>18'48 comp 13 pour rep1</p> <p>18'53 soum 7 → « Le robot sort de la grille »</p> <p>18'54 E – <i>Donc là.. ce que tu vois.. c'est que le robot fait ce que tu veux mais après il ne s'arrête pas.. c'est ça ?</i></p> <p>18'59 S – <i>Oui.. c'est ça</i></p>
	<p>19'02 comp 2, mvt</p> <p>19'03 S – <i>Euh..</i> (comp 0)</p> <p>19'04 E – <i>Tiens essaie deux pour voir ce qu'il fait quand tu as mis deux</i></p> <p>19'07 comp 2</p> <p>19'09 soum 8 → « Il reste de la terre vide de fleur ! »</p>
	<p>19'14 S – <i>Tu vois ?..</i></p> <p>19'17 E – <i>Alors ça veut dire quoi ?</i></p> <p>19'20 S – <i>En fait je voulais mettre douze.. sauf que j'ai d'abord mis le deux</i> (comp 12)</p> <p>19'23 E – <i>Oui mais regarde bien ce qu'il a fait quand tu as mis le deux.. c'était intéressant</i></p> <p>19'27 efface les chiffres dans comp</p> <p>19'30 S – <i>Un.. deux</i> (pt T1 , T2).. <i>trois.. quatre.. cinq !</i> (pt T3 à T5) (comp 5)</p> <p>19'36 E – <i>Alors comment tu trouves cinq ?</i></p> <p>19'39 S – <i>Ben il a planté sur deux pots de terre</i> (pt T1 et T2) <i>seulement quand j'ai mis deux</i></p> <p>19'42 soum 9 valide S- <i>Un.. deux.. trois.. quatre.. cinq..</i> (au fur et à mesure de l'exécution) <i>ouais !</i></p>

## Ugo, 2022 tour 2



## Ugo, 2022t2p4v3 – profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

k5iurkg4 cm1

score global : 160

temps moyen par bloc : 7.8

4 - Chauffer le château [2022-2 white] medium durée : 3'43

ind. rapidité normalisée : 1.6

0 - 26'16

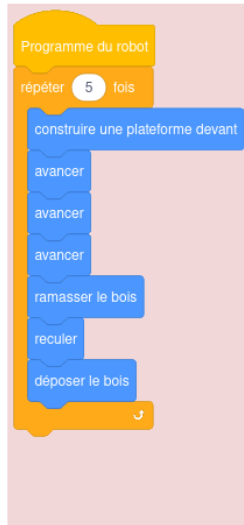
1 - 27'14  
0'58

2 - 27'29  
0'15

3 - 28'22  
0'53

4 - 29'44  
1'22

5 - 29'59  
0'15



id sol :

228

229

230

1622

231

pr-rel REP

Ugo – *Un.. deux.. trois..* (pointe les cases devant le robot : case vide, cheminée, bois) *quatre..* (pointe le tas de bois suivant) *heu un.. deux.. trois.. quatre.. cinq..* (pointe les cases vides)

→ début du schème « compter les plus petites unités visibles » donc les cases, se ravise et compte les cases vides, soit la première case de chaque motif.

comp 5 dans la réserve, ajREP

survole furtivement la grille

Ugo – *Cinq..*

repCST, repAV, repRAM

pointe le premier tas de bois, la cheminée, le robot

Ugo – *Bah.. là y'aura* (pointe le tas de bois, la cheminée, revient vers la réserve, retourne furtivement sur la grille) *mais c'est là..*

intercale un deuxième AV après le premier, repREC, repDEP

→ **Nous observons une prise d'information sur la grille, mais celle-ci est approximative et aboutit à une erreur**

exécution du programme d'id.228 → le robot s'arrête sur la première cheminée, feedback : il n'y a rien à ramasser ! → 1448

laisse la situation à l'issue de l'exécution, pointe la case derrière le robot, la case du robot et le tas de bois

Prend un bloc avancer et l'amène au-dessus de la grille, et énonce chaque action en plaçant le bloc au-dessus de la case

**Ugo dénombre les déplacements en se représentant le robot sur la case du départ, et non par complétion.**

Il intercale ce troisième bloc AV après les deux autres dans le programme.

exécution du programme d'id. 229 → (erreur la plus fréquente pour ce puzzle) → le robot a déposé le premier tas de bois dans la cheminée, feedback : il y a déjà une plateforme ici → 1809

Ugo – *Gnein..*

E – *Le mode pas à pas.. ça doit être un réflexe.. dès que tu fais une erreur..* → 2154

ferme le feedback

Ugo – *Alors.. (pp) là je ramasse le bois.. je recule.. j'ai compris mon erreur*

E – *Alors c'est quoi ton erreur*

Ugo – *Ben en fait.. là ça va tout de suite reprendre..*

E – *Mm*

Ugo – *Mais si..* ( survole le programme de bas en haut) *mais si je mets avancer d'abord.. ça va faire.. un drôle de truc..* → 3165

→ correction de la jonction en plaçant le bloc manquant en début de boucle plutôt qu'en fin de boucle

exécution du programme d'id.230 → le robot reste sur la case de départ, feedback : le robot va se jeter dans le vide

Ugo – *fff..* (prend le bloc AV ce qui décroche toute la séquence, le remet en place) *ah oui c'est vrai..* (à nouveau)

décroche et remet DEP

Ugo – *Gneu.. ben..* (prend le troisième AV et emmène à nouveau les blocs accrochés au-dessous)

E – *Pour enlever.. ça va toujours prendre ce qui est en dessous.. si tu veux*

prend à nouveau toute la séquence → 3809

E – *Si tu veux laisser..* (supprime la séquence) *ah bon.. ben*

Ugo – *Ah là.. construire une plateforme devant* (repCST, pointe la case vide devant le robot) *avancer* (repAV, pointe la cheminée puis la case vide devant le robot, puis à nouveau la cheminée) *trois fois..* (repAV, repAV) *une fois là..* (pointe le tas de bois) *ramasser le bois..* (repRAM) reculer.. (repREC) *déposer le bois..* (repDEP) *là..* (pointe la cheminée) *un, deux..* (pointe le tas de bois puis la deuxième case vide, repAV, repAV)

→ **considère le remplacement du robot mais le replace sur la case vide, celle où il doit construire une plateforme : conflit avec la règle habituelle « le robot agit sur la case où il se trouve ».**

règle le curseur pour exécuter plus rapidement

Exécution du programme d'id.1622 → le robot a déposé le premier tas de bois et avancé ensuite d'une case, feedback : le robot va se jeter dans le vide !

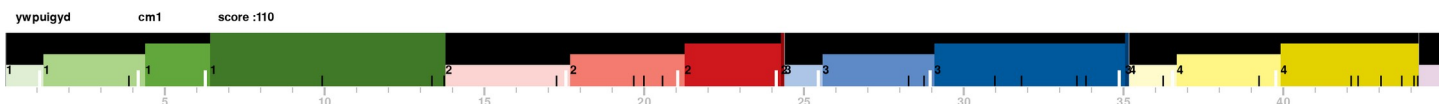
supprime le dernier AV

Exécution du programme d'id.231

Ugo – *Oui !* (lorsque le robot a déposé le troisième tas de bois)

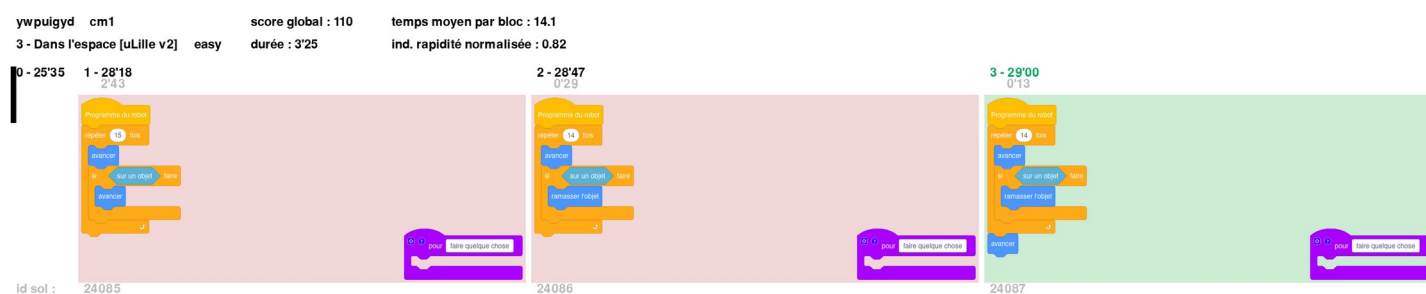
**Ugo, 2022 expérimentation complémentaire**

---



## Ugo, 2022t4p3v2 – profil accommodation

[lien vers l'extrait vidéo](#)



Ugo – **Répéter une deux trois quatre cinq fois** (pointe les groupes de deux objets)

→ schème de dénombrement

ouvre capteurs, ouvre fonctions

Ugo – **Hein c'est quoi ça ?..** (pr-rel un bloc de fonction, sélectionne faire quelque chose) **zut.. euh tu peux m'expliquer ce que c'est..**

E – **Alors c'est un bloc pour créer.. tes propres blocs.. quand.. si tu mets ce bloc-là dans l'éditeur.. vas-y tu le mets.. tu le prends tu le mets sur le côté..** (Ugo amène un bloc dans l'éditeur) **voilà.. plus loin encore.. plus loin.. sans prendre le programme du robot.. parce qu'il ne s'accroche pas celui-là.. tu le mets plus loin sur.. voilà.. oui par exemple.. et tu vois tu as eu un autre bloc qui est apparu que celui-là** (amène un bloc faire quelque chose dans l'éditeur) **que celui-là tu peux utiliser dans le programme du robot.**

amène un autre bloc de fonction dans l'éditeur

E – **Non euh.. enlève celui que tu viens de mettre.. voilà**

supprime le dernier bloc de fonction mis

E – **Donc tu as créé un nouveau bloc et mm.. ce nouveau bloc il va faire les actions que tu vas mettre.. à l'intérieur de celui que tu as placé dans l'éditeur**

ouvre maths

Ugo – **Maths.. c'est quoi ?**

E – **Maths mm.. il s'est mis automatiquement mais c'est.. tu ne t'en occupes pas parce que en fait tu remplis directement**

ouvre contrôles

Ugo – **Donc.. faire.. j'ai pas trop compris** (ouvre fonctions) **pour faire quelque chose**

E – **Alors il faut que tu lui dises quel est ce quelque chose.. il faut que tu remplisses en fait.. que tu remplisse le bloc que tu as mis dans l'éditeur**

ouvre contrôles, ouvre fonctions

E – **Mais si tu ne le connais pas tu n'es pas obligé de l'utiliser..**

Ugo - **Ouais**

E - *C'est comme tu veux*

Ugo – *Alors..* (ouvre capteurs) *ouh non zut* (ouvre contrôles) *répéter dix fois* (ajREP) *non répéter* (compte tout haut jusque quinze en pointant les cases) *fois..* (ouvre actions, survole les blocs) *non quinze* (comp) *quinze fois.. avancer*

→ **Ugo passe spontanément à une imbrication d'une structure conditionnelle dans une boucle. Cela pose la question de la restriction des blocs : aide ou entrave à la progression ?**

ouvre contrôles, amène un bloc SI dans l'éditeur, ouvre actions

repAV

ouvre capteurs

place un bloc sur un objet dans l'emplacement de la condition du bloc SI

place le bloc SI dans REP après avancer

ouvre actions

siAV

Ugo – *Avancer*

Exécution du programme d'id.24085 → le robot avance sans rien ramasser, feedback : le robot sort de la grille

Ugo – *Zut.. zut..*

comp 14

→ schème : il y en a trop alors je mets un de moins

Ugo – *Euh mais..* (survole le programme) *zut..* (supprime le bloc AV dans SI) *pas avancer..* (siRAM)

Exécution du programme d'id.24086 → le robot a ramassé tous les objets, s'arrête une case avant la fusée, feedback : votre robot doit rejoindre la fusée

ajAV

→ **schème de complétion : d'un côté permet la décomposition du problème (PI) quand elle est anticipée, d'un autre côté engendre des décompositions inadaptée comme ici, produit des solutions non optimales, alors que le problème pourrait être traité dans sa globalité de manière économique.**

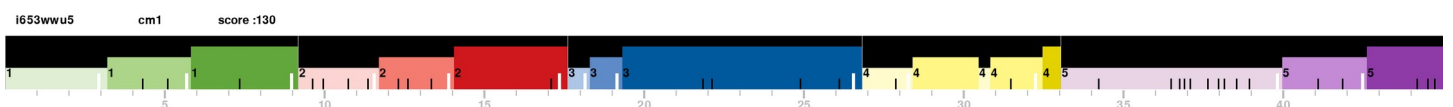
déplace le curseur de vitesse à plus rapide

exécution du programme d'id.24087 → validation

Ugo – *Ouais..*

## Robin, CM1 en 2022

### Robin, 2022 tour 1



### Robin, 2022t1p1v1 – profil indéfini

[lien vers l'extrait vidéo](#)

**Robin, 2022t1p4v2 – profil accommodation**

[lien vers l'extrait vidéo](#)

i653wwu5 cm1

score global : 130

temps moyen par bloc : 8.9

4 - Planter des fleurs [2022-1 white] easy


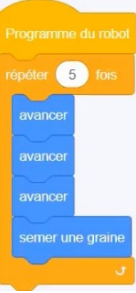
durée : 3'32

ind. rapidité normalisée : 2.4

0 - 28'24 | 1 - 30'50 | 2 - 31'30  
0'40 | 3 - 32'18  
0'48

id sol : 4307 140

	Description
	<p>2022t1s4v2_i653wwu5_CM1.mp4</p> <p>0'00 mvt</p> <p>0'03 aj REP</p> <p>0'04 rep AV</p> <p>0'07 comp 3</p> <p>0'12 pt AV</p> <p>0'14 aj SG</p> <p>0'16 aj REP</p> <p>0'23 rep2 AV</p> <p>0'28 aj SG</p> <p>0'33 aj REP</p> <p>0'36 rep AV, 0 bloc</p> <p>0'42 aj SG, pop-up 0 bloc</p> <p>0'49 ferme pop-up</p> <p>0'51 E – <i>Alors qu'est-ce qu'il se passe là ?</i></p> <p>0'53 mvt</p> <p>0'54 S – <i>Les blocs autorisés.. un bloc en trop utilisé pour 8 autorisés.. j'ai pas trop compris</i></p> <p>1'04 E – <i>Alors.. tu as le droit d'utiliser 8 blocs dans ton programme..</i></p> <p>1'07 comp3 pour rep2</p> <p>1'11 E - <i>..pour construire ton programme.. et seulement huit blocs</i></p> <p>1'14 mvt</p>


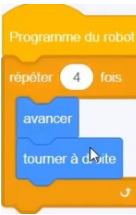
	<p>1'21 compte jusque 9 (chuchote), pas de mvt de souris</p> <p>1'28 S – <b>Bon..</b></p> <p>1'29 E – <b>Alors il y a quelque chose que tu n'as pas vu.. est-ce que tu peux relire l'énoncé ?</b></p> <p>1'30 déplace le prog dans l'éditeur (avec le bloc prRob)</p> <p>1'37 S – <b>Vous êtes bloqué en particulier.. si vous n'avez pas assez de blocs pour finir votre programme.. retournez dans la version une étoile et cliquez sur le bouton plus de détails pour obtenir de l'aide.</b></p> <p>1'52 pt les onglets de version</p> <p>1'56 E – <b>Oui.. donc la version une étoile.. là où tu as une petite étoile jaune (pt).. la version que tu as déjà réussie</b></p>
	<p>2'02 ouvre la version 1 étoile</p> <p>2'03 ouvre plus de détails (chuchoté)</p> <p>2'05 mvt</p> <p>2'07 S – <b>Alors.. on remarque que le même groupe de blocs est utilisé plusieurs fois (mvt).. utilisez le bloc vous permet d'utiliser des blocs</b> (suit le texte).. <b>Aaah.. on peut mettre.. aah d'accord</b> (mvt)</p> <p>2'23 retourne sur la version 2 étoiles</p> <p>2'23 E – <b>D'accord quoi ?.. on peut mettre quoi ?</b></p> <p>2'27 détache SG1 et ce qui est en-dessous</p> <p>2'28 détache SG3 et essaie de le mettre dans la poubelle</p> <p>2'39 E – <b>Donc pour mettre à la poubelle.. tu peux mettre directement au-dessus de la zone grise</b></p> <p>2'46 continue à essayer de mettre dans la poubelle</p> <p>2'47 E – <b>Tu peux faire comme ça aussi.. mais dans la zone grise..</b></p> <p>2'53 sup le morceau de prog détaché au-dessus de la zone grise</p> <p>2'55 E – <b>Voilà</b></p> <p>2'56 rep SG</p> <p>3'01 S – <b>Et voilà</b></p> <p>3'02 soum 1 → « Votre robot a semé hors des zones de terre</p>
	<p>3'05 S – <b>Ah</b></p> <p>3'07 ferme feedback</p> <p>3'09 pt SG1</p> <p>3'10 mvt</p> <p>3'13 S – <b>Qu'est-ce qui va pas ?</b></p> <p>3'17 E – <b>Alors peut-être tu peux..</b></p> <p>3'18 S – <b>Ah oui je</b> (sup SG1).. <b>je crois que je sais</b></p> <p>3'22 rep AV,AV,SG</p> <p>3'35 pt zones de terre (compte en chuchotant)</p> <p>3'40 comp 5</p>

3'47 soum 2 valide






3'55 S – *Yes !*


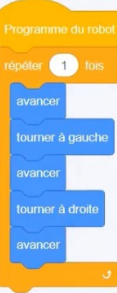

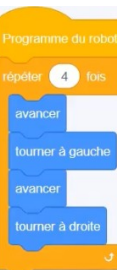
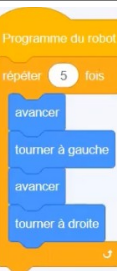
## **Robin, 2022t1p5v1 – profil accommodation**

[lien vers l'extrait vidéo](#)

	Description	Interprétation
	<p><b>2022t1s5v1_i653wwu5_CM1.mp4</b></p> <p>0'00 lat 3s</p> <p>0'03 pr-rel AV</p> <p>0'05 aj REP</p> <p>0'06 S – <i>Alors c'est..</i></p> <p>0'10 rep AV</p> <p>0'15 pt TG,</p> <p>0'18 rep TD chuchote « droite »</p> <p>0'24 pt AV, TD, TG</p> <p>0'28 rep AV (chuchoté), TG (chuchoté)</p> <p>0'36 pt 3,1 (chuchote 1), 3 (chuchote 2), parcourt 5,7,9 (chuchote 3,4), pt 9</p> <p>0'48 mvt</p> <p>0'53 pt TG</p> <p>0'59 pt comp, compte en chuchotant jusque 4</p> <p>1'03 comp 4</p> <p>1'08 soum 1 → « Le robot essaie de se déplacer sur un obstacle ! »</p>	
	<p>1'10 lat 5s</p> <p>1'15 S – hmm</p> <p>1'17 pt TD1</p> <p>1'20 ferme feedback</p> <p>1'22 pp AV,TD1</p> <p>1'26 S – Non..</p> <p>1'28 pt AV2</p> <p>1'31 sup AV2,TG1</p> <p>1'35 mvt</p> <p>1'38 pp itération 1</p>	

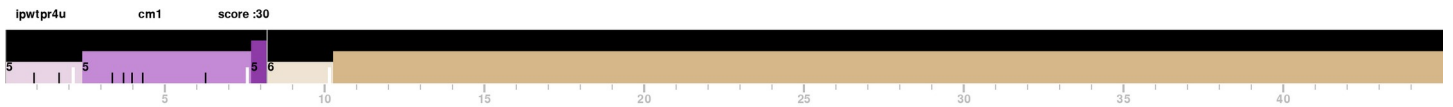


	<p>1'46 sup TD1</p> <p>1'47 S – <b>Trouvé l'erreur</b></p> <p>1'51 perd le bloc prRob</p> <p>1'52 E – <b>Ah là t'as perdu ton bloc programme.. sûrement la barre grise qui est tout à droite.. tu vois la barre.. remonte la.. non dans l'autre sens.. ah faut déjà descendre</b> (monte et descend la barre mais le bloc n'apparaît pas dans l'éditeur).. <b>alors où est-ce que tu.. ah bah voilà</b> (a cliqué sur le fond de l'éditeur et fait réapparaître le programme) .. <b>ça y est on l'a retrouvé</b></p> <p>2'15 déplace le prRob et le programme qui est en-dessous</p> <p>2'18 S – <b>Ah oui.. hein.. pourquoi..</b></p> <p>2'30 E – <b>Tu as toujours ta souris.. lâche ta souris..</b></p> <p>2'38 S – <b>Quoi ?</b></p> <p>2'40 E – <b>Là.. tu as toujours le doigt sur la souris ?.. il faudrait que tu le lâches.. voilà</b></p> <p>2'45 recentre prRob</p> <p>2'53 vide l'éditeur</p> <p>2'56 S – <b>Voilà</b> (mvt dans l'éditeur vide)</p> <p>3'03 S – <b>Alors.. Programme du robot..</b> (mvt)</p> <p>3'08 aj REP</p> <p>3'12 rep AV,TG,AV (verb)</p> <p>3'24 soum 2 → « Le robot essaie de se déplacer sur un obstacle ! »</p>	
	<p>3'26 lat 5s</p> <p>3'31 sup AV2</p> <p>3'38 soum 3 → « Le robot essaie de se déplacer sur un obstacle ! »</p>	
	<p>3'41 sup TG</p> <p>3'49 soum 4 → « Le robot essaie de se déplacer sur un obstacle ! »</p>	
	<p>3'51 comp 0</p> <p>4'01 soum 5 → « Votre robot n'a pas tout ramassé »</p>	
	<p>4'05 ferme feedback</p> <p>4'07 S- <b>Mm</b> (mvt)</p> <p>4'14 rep TD (verb)</p> <p>4'16 survole furtivement la grille</p>	

	<p>4'17 sup TD1</p> <p>4'18 S – <b>Ah non tourner à gauche</b></p> <p>4'21 rep TG</p> <p>4'27 comp 1</p> <p>4'33 soum 6 → « Votre robot n'a pas tout ramassé »</p>	
	<p>4'37 S – <b>Voilà</b></p> <p>4'39 rep AV,TD (verb)</p> <p>4'49 S – <b>Je me demande qu'est-ce que ça fait si je fais ça</b></p> <p>4'52 soum 7 → « Votre robot n'a pas tout ramassé »</p>	
	<p>4'55 S – <b>Oui !.. j'ai compris</b></p> <p>4'57 rep AV</p> <p>5'04 soum 8 → « Votre robot n'a pas tout ramassé »</p>	<b>allonge le motif</b>
	<p>5'07 Oui !</p> <p>5'11 compte jusque 3 en chuchotant, souris immobile</p> <p>5'16 comp 4</p> <p>5'22 soum 9 → « Le robot essaie de se déplacer sur un obstacle ! »</p>	
	<p>5'33 ferme feedback</p> <p>5'36 relance l'exécution soum 10 identique</p> <p>5'40 S – <b>Quel avancer est de trop ?</b> (détache AV3) <b>celui-là peut-être ?</b> (sup AV3)</p> <p>5'47 soum 11 → « Votre robot n'a pas tout ramassé »</p>	
	<p>5'56 comp 3, 34, 0, 4</p> <p>6'11 soum 12 identique</p>	
	<p>6'13 E – <b>Comment tu fais pour trouver 4 ?</b></p> <p>6'16 S- <b>Mm</b></p> <p>6'20 E – <b>Qu'est-ce que tu as compté ?</b></p> <p>6'21 mvt lents</p> <p>6'27 S – <b>Hein ?.. bon je vais essayer cinq voir si ça marche</b></p>	

6'33 comp 5	
6'38 S – <i>Si je fais cinq..</i>	
6'39 soum 13 valide	
6'47 S - <i>Yes !</i>	

## Robin, 2022 tour 2



(complément suite à un souci technique)

### Robin, 2022t2p5v2 – profil accommodation

[lien vers l'extrait vidéo](#)

ipwtpr4u cm1 score global : 30 temps moyen par bloc : 10.5  
 5 - Distribuer le bois [2022-2 white] easy durée : 5'13 ind. rapidité normalisée : 1.71

0 - 2'25	1 - 3'23 0'58	2 - 3'44 0'21	3 - 4'00 0'16	4 - 4'20 0'20	5 - 6'18 1'58	6 - 7'38 1'20
id sol :	3158	4495	461	1232	14614	236

Robin – *Ouh.. encore un truc avec répéter 10 fois.. alors..*

pointe les arbres en comptant tout haut jusque 6

**Ce qui est singulier ici, c'est que ce sont des éléments de décor qui sont pris comme points de repère pour le dénombrement. Nous pouvons nous poser la question de l'appartenance de ces arbres, disposés régulièrement, au motif visuel. En d'autres termes, le motif visuel peut-il contenir des éléments de décor ?**

pointe l'avant-dernier arbre

Robin – *Non cinq.. cinq cinq cinq cinq..*

Pourquoi Robin exclut-il le dernier arbre ?

Robin - **Alors c'est.. avancer** (ajAV) **avancer** (ajAV) **là c'est déjà des tests.. tourner à droite** (ajTD)

Nous pouvons nous demander ce que Robin appelle « test »

entoure la séquence avec REP

Robin – **J'vais déjà voir ça**

Exécution du programme d'id.3158

Robin – **Hein.. ah oui j'ai compris.. c'est.. ensuite c'est tourner à gauche**

Exécution du programme d'id.4495

Robin – **Ah oui entre les deux y'a un avancer** (intercale AV) **voilà**

Exécution du programme d'id.461

Pendant l'exécution, Robin s'apprête à changer la valeur du compteur du bloc répéter

Dénombrer tout haut les 5 premiers arbres.

Robin – **Cinq fois** (comp 5) **ok**

Exécution du programme d'id.1232

Robin – **D'accord** (pendant l'exécution) **un deux** (pointe les deux cases devant le robot) **deux avancer** (aj AV aj AV) **un deux.. quatre blocs.. un deux tourner à droite** (ajTD) **avancer** (ajAV) **ramasser le bois** (ajRAM) **avancer** (ajAV) **donner le bois au..** (pr DEP) **argh** (feedback trop de blocs) **comment est-ce qu'on peut faire** (fenêtre pop-up, que Robin ferme) **d'accord.. oh..** (sup DEP)

En éditant son programme, Robin énonce combien de blocs il peut encore utiliser

temps de latence : 15s

Robin : **j'ai une question**

E – **Je t'écoute**

Robin – **Je crois que j'ai une petite idée** (survole la grille)

Exécution du programme en pas à pas, s'arrête à la fin de la boucle (non enregistré)

Robin – **Et là si je remets une fois** (comp 6) **je me demande ce que ça fait.. comme ça..**

Exécution du programme d'id.14614

Robin – **J'sais pas.. on verra bien** (pendant l'exécution) **d'accord..**

Robin a la bonne démarche, mais il ne compense pas l'ajout d'une itération par la suppression des blocs équivalents dans la suite de son programme. La tentative d'aide de E ne fait pas sens.

E – **Ben oui mais tu as laissé tous tes blocs en-dessous**

Robin – **Ouais.. ah mais oui.. en fait j'ai pas compris**

E – **Qu'est-ce que tu n'as pas compris ?**

Robin – **Je me demande comment est-ce que je vais faire**

E – **Prends le mode pas à pas**

Robin – **Ah oui**

E – **T'as la bonne idée mais seulement t'as laissé tous tes blocs en-dessous alors.. en-dessous du bloc répéter**

exécution du programme jusqu'à ce que le robot ait exécuté la boucle

E – *Où est-ce qu'est le robot lorsqu'il a fini tous les tours de bloc répéter.. regarde bien*

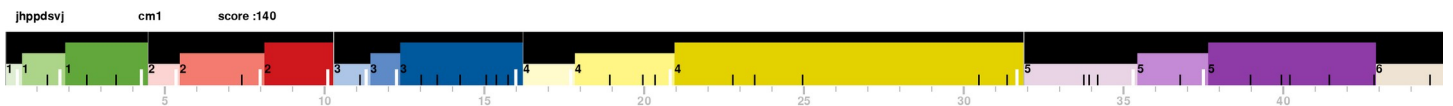
Robin – *Ah d'accord..* (supprime la séquence en-dessous du bloc répéter) *j'ai compris.. il est là donc ramasser le bois* (ajRAM) *avancer* (ajAV) *donner le bois au castor* (ajDEP)

Robin – *Si j'ai bien compris ça marche*

Exécution du programme d'id.236

Les instructions placées après la boucle ont gêné Robin pour comprendre l'exécution du programme. L'aide de E consiste à lui indiquer le moment où il doit arrêter l'exécution.

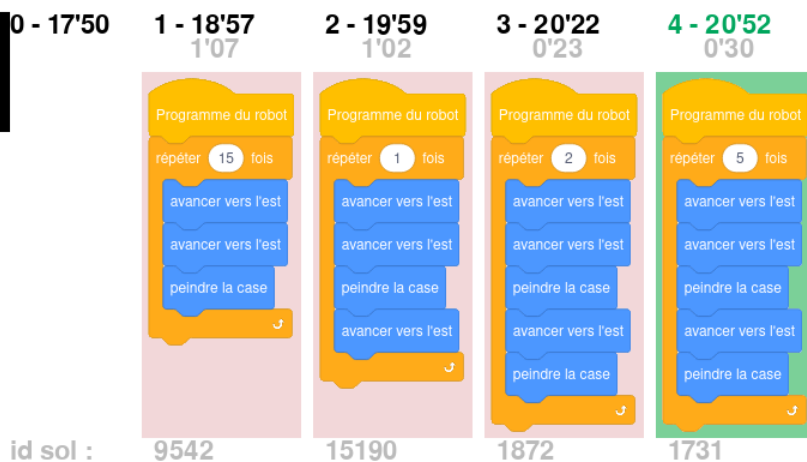
### Robin, 2022 tour 3



### Robin, 2022t3p4v2 – profil accommodation

[lien vers l'extrait vidéo](#)

jhppdsvj cm1 score global : 140 temps moyen par bloc : 11.4  
4 - Peindre le dessin [2022-3 white] easy durée : 3'02 ind. rapidité normalisée : 1.0



pointe le 1<sup>er</sup> point

Robin – *Un*

pointe le robot

Robin – *Avancer vers..* (prend un bloc EST et le laisse de côté sur l'éditeur) *alors..* (ajREP) *si j'en déduis c'est..* (survole le premier groupe de deux points) *avancer vers l'est.. avancer vers l'est* (met le bloc EST resté dans l'éditeur dans la boucle) *avancer vers l'est* (pointe le premier point, puis le 2ème) *avancer vers l'est* (repEST) *peindre la case* (ajPC, survole la grille, pointe chaque case en dénombrant à voix haute jusque 15)

→ schème de dénombrement

comp 15

→ traite une case à la fois comme dans la classe de situation précédente, sans tenir compte que toutes les cases ne doivent pas être traitées de la même façon (ouverture vers instruction conditionnelle imbriquée dans une boucle si le bloc était disponible), la constitution du motif et le dénombrement sont cohérents

exécution du programme d'id.9542

Robin – **Oh tiens..** (le robot peint une case sur deux)

feedback : le robot sort de la grille

pr un bloc PC, survole le programme avec le relâche

Robin – **Attends.. je vais juste voir ce qui va pas** (commence une exécution en mode pas à pas) **alors là il peint la case** (le robot est sur le premier point et le bloc peindre la case est surligné, pp) **et là ça revient au début.. avancer vers l'est.. alors ça voudrait dire que..** (pointe le bloc EST de la réserve, prend un bloc PC)

E – **Alors ça voudrait dire que quoi ?**

Robin – **On doit mettre un.. avancer vers l'est** (repEST).. **là j'veis voir ce que ça fait** (comp 1)

exécution du programme d'id.15190 → le robot a peint le premier point et se trouve sur le 2ème, feedback : votre robot n'a pas peint les bonnes cases

Le feedback est ambigu dans ce cas de figure

→ **Nous avons là un passage par l'unité spontané très intéressant.**

survole le programme

repPC

comp 2

exécution du programme d'id.1872 → le robot a peint les deux premiers groupes de points, feedback : votre robot n'a pas peint les bonnes cases

Le fait de mettre le compteur de boucle à 2 permet de se concentrer sur la jonction entre les deux premières itérations. La stratégie de Robin pour trouver le motif est efficace.

Robin – **Yes.. alors combien il faut le répéter ?**

dénombre tout haut les groupes de deux points

Robin – **Cinq fois** (comp 5) **et non quinze**

exécution du programme d'id.1731 et validation

E – **Alors qu'est-ce que tu avais compté pour quinze et qu'est-ce que tu comptes pour cinq ?**

Robin – **Bah à quinze j'avais compté les carreaux.. et maintenant je m'en.. je me suis rendu compte que non**

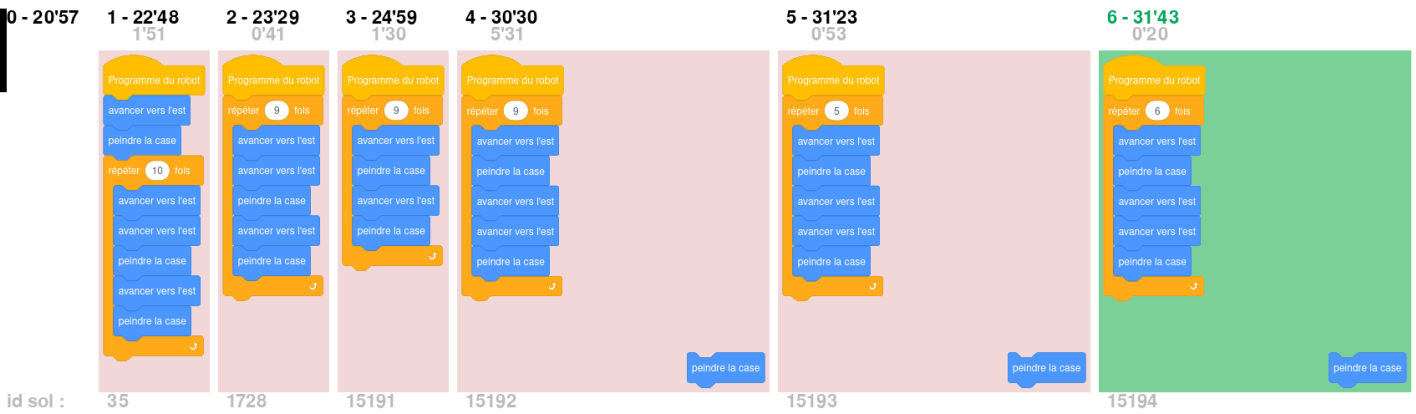
E - **Mm**

### **Robin, 2022t3p4v3 – profil accommodation**

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

[lien vers l'extrait vidéo 3](#)



ajEST, ajPC

Robin – ***Maintenant c'est..*** (pointe le premier point) ***alors voilà..*** (amène un bloc REP dans l'éditeur) ***elle sert à quoi cette petite flèche ?*** (pointe la flèche de signalétique en bas à droite du bloc répéter)

E – ***Elle sert à dire que ça va recommencer au début quand c'est arrivé à la fin.. de toutes les instructions***

Robin – ***Ah.. d'accord***

→ 860

pointe le deuxième point

Robin – ***Alors.. c'est avancer vers l'est*** (repEST) ***on va arriver là*** (pointe la case vide après le premier point) ***avancer vers l'est*** (repEST) ***peindre la case*** (repPC) ***on en est à là*** (pointe le deuxième point) ***avancer vers l'est*** (repEST) ***peindre la case*** (repPC)

→ inférence lors de la constitution du motif : simulation de l'exécution révélée par des allers-retours entre la grille et le programme en cours d'édition

accroche la boucle à la suite du début de programme → 2029

Robin – ***Le problème c'est qu'on a huit blocs autorisés.. comment est-ce qu'on peut faire ?***

E – ***Alors c'est ça qu'il faut chercher.. comment tu peux faire ?***

exécution du programme d'id.35 → tous les points sont peints, feedback : le robot sort de la grille !

→ 2697

survole le programme, pointe le bloc EST avant la boucle, décroche le programme

Robin – ***Je me demande si.. y'a besoin d'ça*** (met les deux instructions qui étaient avant la boucle à la poubelle)

comp 9

Robin – ***J'mets une fois de moins***

→ illustration du schème « Il y en a trop, je mets une fois de moins »

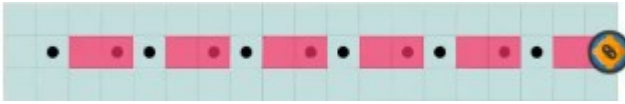
accroche le programme

Robin – ***Qu'est-ce que ça fait ?***

Exécution du programme d'id.1728

Robin – ***Il sort toujours de la.. ah ben.. et ben d'accord il est totalement décalé..***

→ **idée de décalage prémisse à la fenêtre glissante**



→ 3689

pointe EST dans la réserve, puis PC

Robin – ***Alors si je résume c'est avancer vers l'est.. peindre la case.. peindre la case.. non..*** (souffle)

pointe le premier point

Robin – ***Ou il faudrait que.. ce soit peindre la case.. peindre la case.. peindre la case*** (pointe les trois premiers points)

→ Omet les déplacements dans sa description

Robin – ***Faudrait tout peindre***

E – ***Oui.. et il faut peindre que les points noirs***

Robin – ***Oui.. il faut peindre que les points noirs ? aïe..***

temps de latence : 10s

décroche le programme, décroche la séquence, supprime le premier bloc, remet le reste dans la boucle et raccroche le programme.

exécution du programme d'id.15191 → le robot peint toute la ligne , feedback : Votre robot n'a pas peint les bonnes cases.

Robin – ***Ah oui***

temps de latence : 20s

→ 6439

Robin – ***Non.. je suis pas sûr mais je pense que j'ai trouvé***

pointe le premier point

survole le programme

ajoute un bloc EST en début de boucle

Robin – ***Alors.. là*** (parcourt le corps de la boucle de haut en bas) ***il faut un avancer vers l'est.. mais que à la fin.. mais comment est-ce que je peux faire ?.. que à la fin et au début..***

E – ***Qu'est-ce que tu veux dire par que à la fin et au début ?***

Robin – ***Ben là y'a un point noir.. un seulement..***

E – ***Oui***

Robin – ***Et là y'en a un seul.. seulement***

E – ***Oui***

Robin - ***Le problème..***



E – *C'est quoi le problème ?*

Robin – *C'est que je sais pas comment on peut faire pour que ce soit à la fin et au début*

E – *Mm.. d'accord ce que tu veux dire c'est que .. tu sais peindre*

Robin – *Ce point-là et ce point-là* (pointe le premier et le dernier point)

E – *Donc ceux-là ils te posent problème ?*

Robin – *Oui* → 8469

E – *Mm*

→ **verbalisation de la stratégie qui consiste à traiter le premier et le dernier élément hors de la boucle**

ajoute un bloc PC après le premier EST

E – *Et pourquoi ils te posent problème ?*

Robin – *Si j'fais ça..* (lance une exécution) *il se passe ça.* (interrompt l'exécution avant la fin)

remet en situation initiale

Robin -*Voilà..*

décroche le bloc PC précédemment ajouté

Robin – *Lui je sais pas où le mettre* (survole la séquence avec le bloc PC, les autres blocs s'écartant chacun leur tour)

→ lui correspond au bloc qui code le dépôt de peinture sur la première case. Robin sent la nécessité de placer cette instruction dans la boucle, ce que révèle le survol du corps de la boucle muni de ce bloc, mais il ne sait pas délimiter le motif pour arriver à cela.

E – *Alors c'est peut-être le mode pas à pas qui va t'aider*

Robin – *Ouais*

laisse le bloc PC dans l'éditeur

lance une exécution en mode pas à pas

exécute les deux premiers blocs EST de la première itération

Robin – *Ah..*

temps de latence : 15s

pp trois itérations

Robin – *Y'a un problème.. tout est décalé..*

→ 10848

E – *Alors si tu recommences avec le mode pas à pas et que tu regardes bien où ça commence à être décalé..*

continue la même exécution en pp quelques clics

remet en position initiale

Robin – *Avancer vers l'est* (pp) *ok..* (pointe le bloc PC qui est resté dans l'éditeur) *et là il faut peindre la case..*

→ Robin énonce correctement les deux premières instructions mais il intercale le bloc PC après les deux premiers EST

Robin – *Je vais essayer* (pp) *il va la peindre.. hein* (après le deuxième EST) *ah oui..* (déplace le bloc PC après le premier EST → 12368) *si j'le mets là.. tout ça j'le mets là* (raccroche la fin de la séquence dans la boucle)

Nouveau début d'exécution en pp

Robin – *D'accord* (quand le premier point est peint) *il a peint..* (pp, enlève le deuxième bloc PC)

souci de manipulation, change l'ordre d'autres blocs sans faire exprès

E – *Ah tu l'as pas remis comme il faut*

Robin – *Oui je pense aussi* (pp) *là ah.. ah oui.. en effet* (rectifie)

exécution du programme d'id.15192 → le robot peint tous les points noirs, feedback : le robot sort de la grille

→ **Plusieurs tentatives avec le mode pas à pas sont nécessaires, il semble que Robin trouve un peu par hasard, car à aucun moment il ne montre ou mentionne la limite entre les motifs, ce qui est confirmé par la difficulté à les dénombrer. Il pointe la deuxième case de terre dans le groupe de deux, puis il décale au cours du dénombrement, pour aboutir à cinq sur la dernière zone de terre. Il semble que le premier et le dernier motif comportent trois zones de terre, sans que cette incohérence avec le nombre de blocs « peindre la case » du motif algorithmique ne soit explicitée. Ce pointage révèle que Robin n'a pas construit l'invariance dans l'exécution du corps de la boucle : dans la représentation que révèle ses gestes de pointage, ce n'est pas toujours le même motif algorithmique qui est exécuté, celui-ci n'est pas répété à l'identique. Lorsque l'expérimentatrice l'interpelle, il met en cause la bonne mise en œuvre du schème de dénombrement, tout en indiquant qu'il compte « les répétitions » et procède par ajustement. Dans cette confrontation, la stratégie est approximative, le motif visuel n'est pas identifié, la correspondance avec le motif algorithmique n'est que partiellement construite, la résolution aboutit positivement grâce à une part d'essai-erreur.**

survole la grille

compte tout bas

Robin – *Oh oui j dois le répéter..* (compte tout haut jusque cinq en pointant les groupes de deux points) *on va essayer quatre*

E – *Alors tu as compté cinq et tu essaies quatre.. tu peux m'expliquer ?*

Robin – *Non je me suis dit que j'avais peut-être mal compté*

E – *Alors qu'est-ce que tu as compté ?*

Robin – *Ben les.. j'veais essayer avec cinq*

comp 5

exécution du programme d'id.15193

E – *Qu'est-ce que tu avais voulu compter ?*

Robin – *La répétition*

→ le robot n'a pas peint le dernier motif

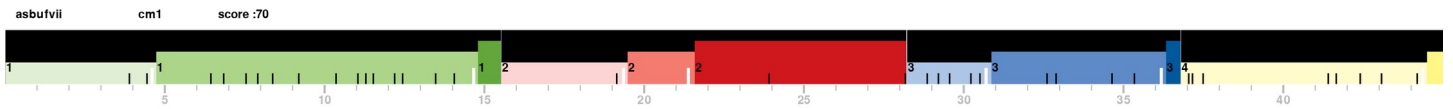
Robin – *Ah cinq.. il faudrait six*

comp 6

exécution du programme d'id.15193 → validation

Robin – *Alors.. ouais !*

## Robin, 2022 expérimentation complémentaire



### Robin, 2022t4p3v2 – profil accommodation

[lien vers l'extrait vidéo](#)

asbufvii cm1

score global : 70

temps moyen par bloc : 19.1

3 - Dans l'espace [uLille v2] easy

durée : 5'22

ind. rapidité normalisée : 1.11

0 - 30'52

1 - 32'38  
1'46

2 - 32'55  
0'17

3 - 34'40  
1'45

4 - 35'22  
0'42

5 - 36'14  
0'52

Programme du robot

répéter 10 fois

avancer

ramasser l'objet

avancer

ramasser l'objet

avancer

ramasser l'objet

avancer

avancer

Programme du robot

répéter 10 fois

avancer

avancer

ramasser l'objet

avancer

ramasser l'objet

avancer

ramasser l'objet

avancer

avancer

Programme du robot

avancer

répéter 10 fois

ramasser l'objet

avancer

ramasser l'objet

avancer

avancer

Programme du robot

avancer

répéter 5 fois

ramasser l'objet

avancer

ramasser l'objet

avancer

avancer

avancer

Programme du robot

avancer

répéter 5 fois

ramasser l'objet

avancer

ramasser l'objet

avancer

avancer

reculer

id sol :

8353

8354

8355

8356

3403

Robin – *Capteurs* (ouvre capteurs) *fonctions* (ouvre, amène un bloc fonction dans l'éditeur, clique sur le point d'interrogation qui ouvre la possibilité d'écrire un commentaire) *décris cette fonction..* (sélectionne faire quelque chose)

E – *Alors tu peux.. tu peux choisir un nom pour une fonction.. et une fonction en fait ça te crée un nouveau bloc.. je ne sais pas si tu en a besoin mais en tout cas ça fait ça..*

Robin – *Ah ok..* (supprime le bloc de fonction, ouvre maths) *ok maintenant.. maths.. hein..* (amène un bloc zéro dans l'éditeur)

E – *Maths tu n'en aura pas besoin.. il se met automatiquement mais.. ça c'est sûr*

supprime le bloc zéro, ouvre capteurs, ouvre actions, ouvre contrôles, pr-rel REP

→ explore le menu dont des blocs qu'il ne connaît pas, compromis entre la mise à disposition d'outils, et le fait que des blocs inutiles pour le puzzle agissent comme distracteurs

Robin – *Ok* (chuchoté) *avancer ramasser un panneau avancer ramasser un panneau avanc.. avancer ramasser un panneau.. avancer avancer.. ok*

ajREP, ouvre actions

Robin – *Avancer* (repAV) *ramasser l'objet* (repRAM) *avancer* (repAV) *ramasser l'objet* (repRAM) *et maintenant avancer* (repAV) *avancer* (repAV) *ok*

→ **Robin a bien vu que le motif doit comporter le traitement du groupe de deux objets, mais il se trompe dans le remplacement du robot**

exécution du programme d'id.8353 → il n'y a rien à ramasser



Robin – **Hein..**

ajoute AV en début de boucle, pr-rel la séquence

→ **repart de la position du robot lorsque l'exécution s'interrompt, sauf que l'exécution n'est pas complète, elle est suspendue lors de la première erreur. Ce n'est pas homogène, car pour le dépôt l'exécution va à son terme. Le sujet construit des schèmes qui sont dépendant de l'EIAH.**

exécution du programme d'id.8354 → il n'y a rien à ramasser



Robin – **Ok.. c'est pas là qu'il y a le problème..** (supprime le bloc AV mis précédemment)

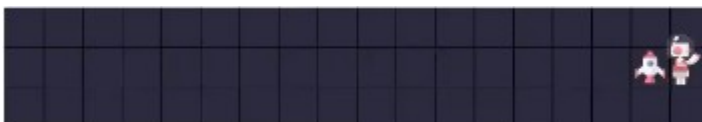
déplace le programme dans l'éditeur de manière désordonnée, puis la barre de scroll

→ probablement signe de blocage

Robin - **Mettons ça là** (replaces le programme à sa position habituelle, après l'avoir déplacé encore quelques secondes) **ok du coup là.. pas à pas..** (pp) **là c'est ramasser l'objet** (pp) **avancer** (pp) **avancer** (pp) **ah attends j'ai une idée** (pr-rel le dernier AV) **si j'mets comme ça** (met le premier AV avant la boucle) **j'essaie**

exécution du programme d'id.8355

Robin – **On dirait que j'ai réussi**



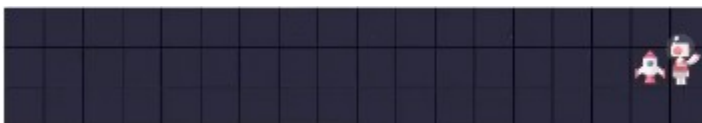
→ il n'y a rien à ramasser !

Robin – **Oups**

pointe 5 sur le clavier virtuel, comp 6, temps de latence, clic sur 5 → 65, supprime les deux chiffres

Robin – **Soixante-cinq..** (clicque sur 2, puis sur 5 → 25, supprime les deux chiffres, clicque sur 5 et re-ferme) **j'essaie**

exécution du programme d'id.8356 → votre robot doit rejoindre la fusée



Robin – **Alors comment faire ?**

temps de latence : 10s

Robin – **J'aimerais découvrir les autres versions**

E – Alors non elles sont bloquées.. tout le temps que tu n’as pas réussi celle-là.. donc là regarde tu y es presque.. qu’est-ce qu’il doit encore faire le robot ?

Temps de latence avec mouvement de souris : 10s

E – Pour rejoindre la fusée..

Robin – Reculer (ajREC)

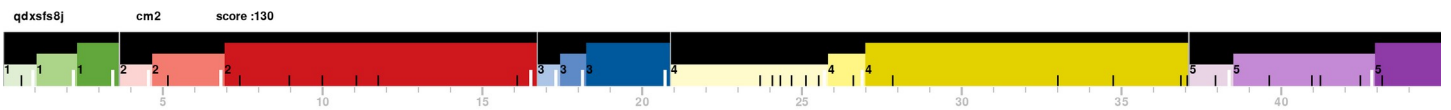
exécution du programme d’id.3403 → validation

Robin - Ouais

→ Les différentes manifestations de l’activité du sujet répondent à la question : comment appréhender l’activité du sujet sur l’interface, en synthèse, après avoir étudié le rôle de la souris.

## Sven, CM2 en 2021

### Sven, 2021 tour 2



### Sven, 2021t2p4v3 – profil échec

[lien vers l’extrait vidéo](#)

qdxsfs8j cm2

score global : 130

temps moyen par bloc : 8.1

4 - Jouer aux dominos [white 2021]

medium durée : 10'06

ind. rapidité normalisée : 8.35

0 - 26'59

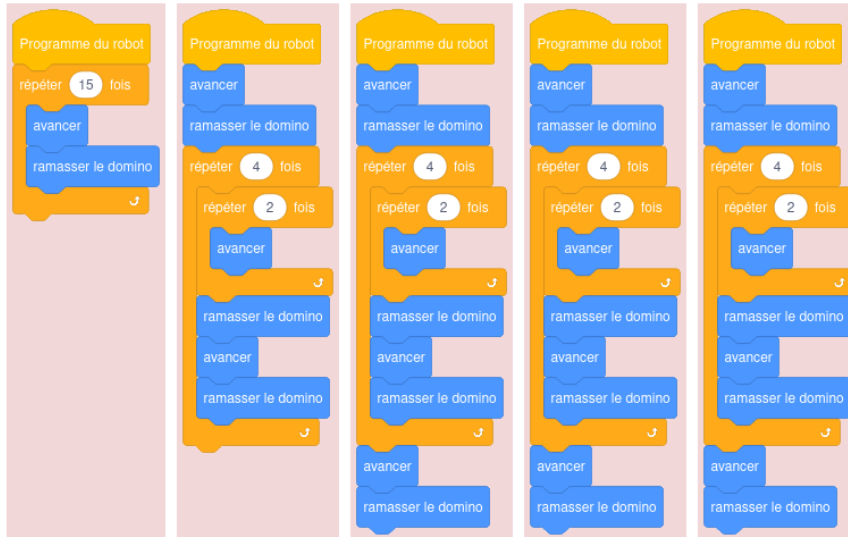
1 - 27'52  
0'53

2 - 33'02  
5'10

3 - 34'46  
1'44

4 - 36'53  
2'07

5 - 37'05  
0'12



id sol :

18694

18695

18696

18696

18696

### Sven, 2021t2p5v1 – profil ajustement

[lien vers l’extrait vidéo](#)

qdxsfs8j cm2

score global : 130

temps moyen par bloc : 8.1

5 - Planter des fleurs [white 2021] basic

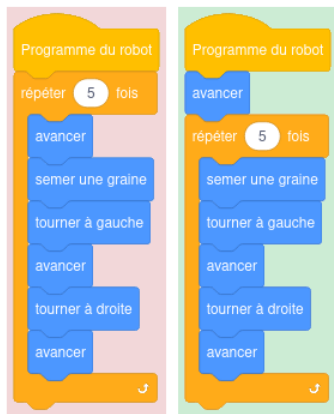
durée : 1'17

ind. rapidité normalisée : 0.19

0 - 37'08

1 - 37'58  
0'50

2 - 38'25  
0'27



id sol :

18697

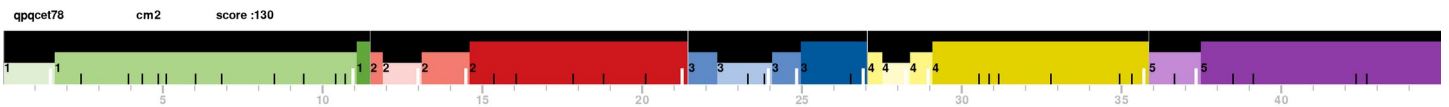
18698

pas de verbalisation, une transcription n'apporterait rien en plus de la suite des programmes soumis, car il n'y a pas d'éléments particuliers du point de vue du processus.

**un bloc avancer de trop dans le motif, corrige en plaçant le premier AV avant la boucle**

## Lou, CM2 en 2021, 6<sup>ème</sup> en 2022

### Lou, 2021 tour 1 passage 1



### Lou, 2021t1p1v1 -profil indéfini

[lien vers l'extrait vidéo](#)

### Lou, 2021t1p1v2 - profil accommodation

[lien vers l'extrait vidéo](#)

qpqcet78 cm2

score global : 130

temps moyen par bloc : 8.5

1 - Planter des fleurs [white 2021] easy

durée : 9'23

ind. rapidité normalisée : 8.46

0 - 1'37

1 - 2'28  
0'51

2 - 3'57  
1'29

3 - 4'23  
0'26

4 - 4'53  
0'30

5 - 5'08  
0'15

6 - 6'03  
0'55

7 - 6'52  
0'49

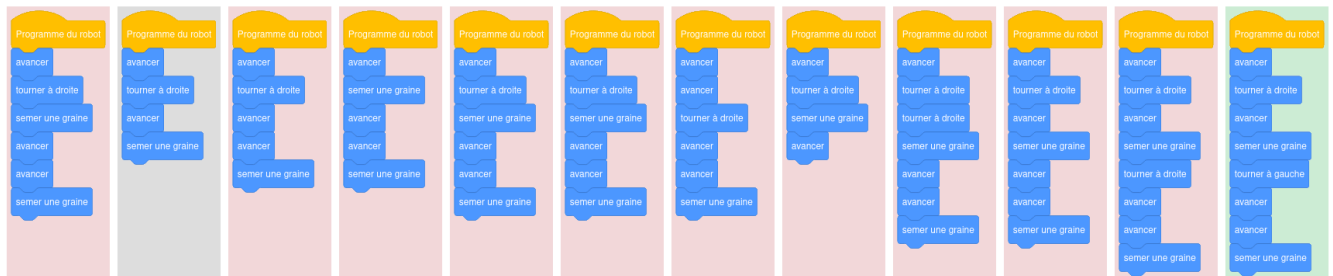
8 - 8'31  
1'39

9 - 9'26  
0'55

10 - 10'26  
1'00

11 - 10'44  
0'18

12 - 11'00  
0'16



id sol :

19074

6441

19075

19076

19074

19074

19077

5276

19078

19079

19080

19081

### Lou, 2021t1p4v1 - profil expert

[lien vers l'extrait vidéo](#)

0 - 27'31

1 - 28'18  
0'47



id sol : 4277

**Tour 1 passage 1 version basic :**  
**2021t1s4v1\_qpqcet78\_CM2\_identification\_motif.mp4**

Le sujet lit la consigne à mi-voix en accompagnant sa lecture avec un mouvement de souris sur le texte.

Il accroche les blocs en séquence en se représentant l'exécution du point de vue du robot :

*Lou – J'avance.. je sème une graine..*

Après 4 blocs il arrête et contrôle par un mouvement de pointage des deux premières zones de terre avec la souris.

*Lou – Alors attends, avancer.. semer une graine..* (pointe la première puis la deuxième case de terre)

ajAV ajSG, pointe les cases de terre

ajAV, ajSG

Après 6 blocs, un nouveau contrôle en parcourant les trois premières zones de terre avec la souris, puis le sujet place les deux derniers blocs ;

*Lou – Attends, je vais juste faire un truc pour voir.. est-ce que ça marche ?*

Il place un bloc répéter dans l'éditeur et met un bloc avancer et un bloc semer une graine dedans.

*Lou – Attends j'vais juste faire un truc pour voir est-ce que ça marche..* (ajREP, repAV) *est-ce que ça marche ?* (repSG) *ok bon c'est bon ça marche.. c'est bon.. ça y est j'ai compris..* (supprime le bloc répéter et ce qu'il contient) *j'pouvais faire autre chose.. mais bon ça ça marche..* (lance l'exécution, valide) *c'est bon*

passage à la version 2 étoiles

*Lou – C'est bon j'ai compris donc.. alors on met répéter 10 fois j'crois là..* (ajREP, pointe les cases de terre en dénombrant – chuchoté) *répéter douze fois.. alors un.. deux..* (comp 12) *douze fois.. avancer semer une graine.. avancer..* (repAV) *semer une*

La prise de conscience du motif se fait progressivement et de manière concomitante à l'extension du schème d'usage du bloc de répéter : possibilité de mettre plusieurs blocs dans le bloc répéter

Il y a contrôle du nombre de motifs déjà placés par pointage des éléments de la grille : un élément de la grille correspond à deux instructions ? Autant de blocs semer une graine que d'éléments sur la grille ?

Cette confrontation illustre bien le fait que le passage de une à plusieurs instructions dans le corps de la boucle relève au moins en partie d'une règle d'utilisation du bloc de boucle : est-il possible de placer plusieurs instructions à l'intérieur de ce bloc ? Nous sommes sur la construction d'un schème d'usage plutôt que d'un schème d'action instrumentée

### Lou, 2021t1p4v2 – profil indéfini

[lien vers l'extrait vidéo](#)

ppqcet78 cm2

score global : 130

temps moyen par bloc : 8.5

4 - Planter des fleurs [white 2021] easy

durée : 1'04

ind. rapidité normalisée : 0.51

0 - 27'04

1 - 28'23

2 - 29'00  
0'37



id sol :

883

Si on ne considère que le deuxième passage, après avoir résolu la version 1 étoile avec une séquence et tester le bloc répéter lors de cette résolution, il s'agit d'un **PROFIL EXPERT**.

### Lou 2021t1p4v3 – profil accommodation

[lien vers l'extrait vidéo 1](#)

[lien vers l'extrait vidéo 2](#)

[lien vers l'extrait vidéo 3](#)

ppqcet78 cm2

score global : 130

temps moyen par bloc : 8.5

4 - Planter des fleurs [white 2021]

medium durée : 6'40

ind. rapidité normalisée : 4.88

0 - 29'05

1 - 30'34  
1'29

2 - 30'53  
0'19

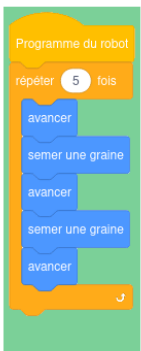
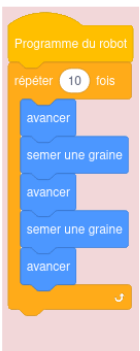
3 - 31'11  
0'18

4 - 32'49  
1'38

5 - 34'58  
2'09

6 - 35'21  
0'23

7 - 35'45  
0'24



id sol :

15450

19086

884

19087

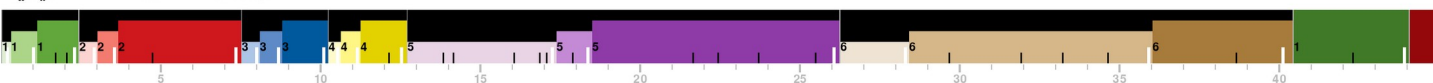
19088

15452

885

### Lou, 2021 tour 1 passage 2

8jjixqpe cm2 score :190



### Lou, 2021t1p4v3 – profil accommodation

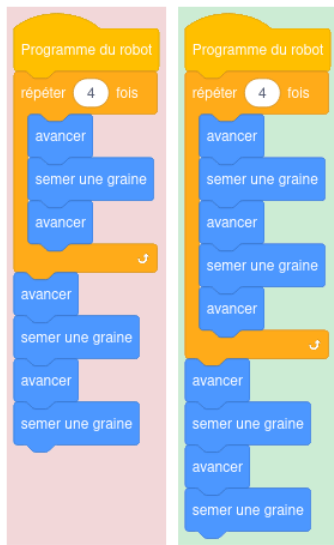
[lien vers l'extrait vidéo](#)



0 - 11'15

1 - 12'10  
0'55

2 - 12'35  
0'25



id sol :

6490

6491

Description	Interprétation
<p>0'00 (2) accroche un bloc répéter dans l'éditeur</p> <p>(3) place un bloc avancer et un bloc semer une graine dans le bloc répéter.</p> <p>(5d) parcourt les deux premières cases de terre et s'arrête sur la case suivante</p> <p>S – <b>Semer une graine, avancer</b></p> <p>(7pg) pointe la deuxième case de terre</p> <p>(3) ajoute un bloc avancer dans le bloc répéter</p> <p>(1p) pointe la première case vide de terre, puis les 4 cases vides entre les groupes de deux cases de terre.</p> <p>(4) met la valeur du compteur à 4</p> <p>(7pg) pointe la dernière case vide intercalée entre deux zones de terre</p> <p>(6) place les blocs avancer, semer une graine, avancer, semer une graine après le bloc répéter</p> <p>(5a) soumission 1 → message « Votre robot a semé hors des zones de terre »</p>	<p>dénombrer des cases vides pour renseigner le compteur de boucle, mais ne prend pas en compte la dernière. Est-ce parce qu'elle est visuellement différente car non entourée par deux zones de terre</p> <p>La dernière case n'est pas vue comme faisant partie d'un motif, donc le dernier groupe de 2 cases de terre est traité à part.</p> <p>Pourquoi, alors que le motif est bien délimité, Lou ne met-elle qu'un bloc semer une graine dans le corps de la boucle ?</p>
<p>0'55 S – <b>Crotte.. oh non, j'me suis ratée</b></p> <p>(7pg) pointe les deux premières cases de terre</p> <p>(3) ajoute un bloc semer une graine et un bloc avancer dans le bloc répéter.</p> <p>(5a) Soumission 2 valide</p>	<p>phase d'analyse de l'exécution sur la grille</p>

0 - 17'23    1 - 17'55    2 - 18'25  
0'32    0'30

id sol : 6496    3655

verbalisation

motif trouvé du premier coup

valeur par défaut dans le compteur

pointage pour le dénombrement des motifs net

—

Lou – *Oh non j’vais pas y arriver.. avancer.. avancer.. ça* (ajREP) avancer (repAV) avancer (repAV)

survole brièvement la grille

Lou – *Tourner à gauche* (repTG) *avancer* (repAV) *tourner à droite* (repTD)

exécution du programme d’id.6496

Lou – *Ah non c’est pas répéter dix fois*

→ Le robot sort de la grille

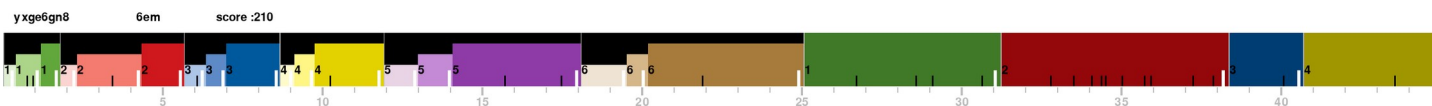
Lou – *C’est pas répéter dix fois.. alors combien est-ce que ça fait ?*

ferme le feedback, remet dans l’état initial

**Lou parcourt le trajet du robot, elle marque un temps d’arrêt sur les cases vides avant les groupes de deux pierres, qui correspondent à l’emplacement du robot au début de chaque itération. Simultanément à ce temps d’arrêt qui correspond à un pointage, Lou numérote verbalement de un à six, et termine en totalisant en ajoutant « fois » au nombre attribué. Nous voyons parfaitement dans cette confrontation quel point de repère le sujet prend pour compter les motifs.**

comp 6

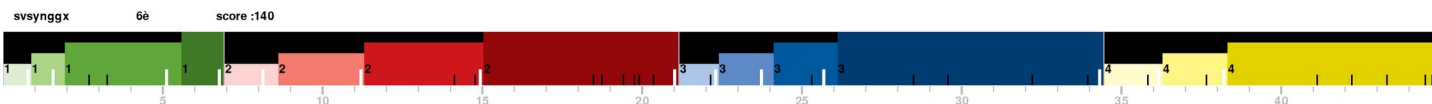
exécution du programme d’id.3655 → validation



## Lou, 2022t1p4v1 – profil expert

[lien vers l'extrait vidéo](#)

## Lou, 2022 expérimentation complémentaire



## Lou, 2022t4p3v1 – profil ajustement

[lien vers l'extrait vidéo](#)

svsynggx 6em

score global : 140

temps moyen par bloc : 8.9

3 - Dans l'espace [uLille v2] basic

durée : 1'11

ind. rapidité normalisée : 0.33

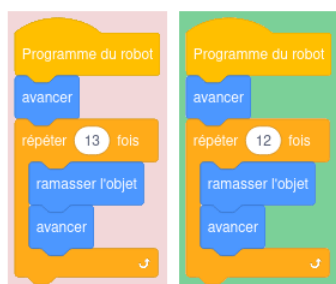
0 - 21'10

1 - 22'10

2 - 22'21

1'00

0'11



id sol :

20455

13018

[explicitation à propos du puzzle précédent]

survole la grille et la réserve de blocs

ajREP

pointe les objets et aussi la fusée

ajoute AV avant le bloc répéter

comp 13, repRAM, repAV

exécution du programme d'id.20455

rectifie le compteur à 12

validation

[explicitation – capturer à nouveau l'extrait vidéo car actuellement se trouve sur la version suivante]

**Lou – Ah bah c'était.. il fallait que j'lui dise d'abord avancer parce que vu qu'il y avait la fusée au bout et ben.. on pouvait pas ramasser d'objet donc c'était avancer.. et comme ça on est déjà sur la case et après.. fallait avancer.. et j'ai compté la.. la fusée comme une.. comme la case où il y avait un objet dessus et j'trouve il fallait ramasser sur cette case.. et faire que douze fois.. et pas treize**

## Lou, 2022t4p3v2 – profil expert

svsynggx 6em

score global : 140

temps moyen par bloc : 8.9

3 - Dans l'espace [uLille v2] easy

durée : 1'23

ind. rapidité normalisée : 0.17

0 - 22'24 1 - 23'47  
1'23



id sol : 3545

explore le menu, ouvre fonctions, puis maths, puis capteurs, puis contrôles, puis actions

survole brièvement la grille

ouvre contrôles, ajREP

parcourt la ligne d'objet en ralentissant sur chaque groupe de deux, chuchote quatre en cours de parcours.

→ Parfois le pointage est bien net, parfois il ne s'agit que d'une variation de vitesse dans le parcours.

comp 5

ouvre actions

repAV, repRAM, repAV, repRAM

ferme une fenêtre pop-up de notification

survole la grille et pointe les deux premiers objets

→ la simulation mentale de l'exécution provoque la correction du motif algorithmique

rep(AV)

survole brièvement la fin de la ligne d'objets

exécution du programme d'id.3545

Lou – *Normalement là ça marche*

→ validation

E – *Comment tu sais que ça marche ?*

Lou – *Ben parce que c'était.. parce que c'était.. y'avait cinq groupes* (fait des mouvements circulaires avec la souris à l'emplacement du premier groupe de deux objets) *de deux.. de deux objets*

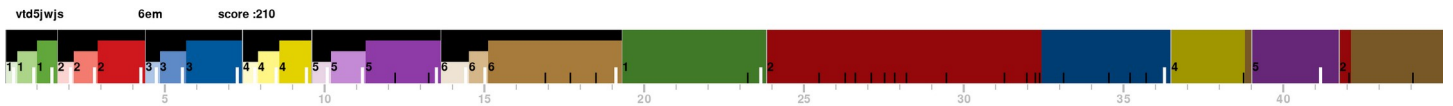
E – *Mm*

Lou – *Et du coup si on faisait cinq fois avancer ramasser l'objet avancer ramasser l'objet avancer.. ça tombait pile poil sur la.. sur la fusée.*

→ Lou verbalise la représentation de la position du robot après exécution de toutes les itérations.

## Louna, CM2 en 2021, 6<sup>ème</sup> en 2022

### Louna, 2022 tour 1



### Louna, 2022t1p3v3 – profil expert

[lien vers l'extrait vidéo](#)

### Louna, 2022t1p5v2 – profil expert

[lien vers l'extrait vidéo](#)

	Description	Interprétation
	<p>2022t1s5v2_vtd5jwjs_6e.mp4</p> <p>0'00 lat 5s</p> <p>0'05 ajREP</p> <p>0'08 survole le trajet du robot par à coups qui correspondent aux marches</p> <p>0'12 comp 4</p> <p>0'15 lat 6s</p> <p>0'21 repAV, rep TD</p> <p>0'33 pt 1.2, 1.3, 1.4</p> <p>0'35 repREP</p> <p>0'39 comp 3 dans REP imbriqué</p> <p>0'43 aj AV dans REP imbriqué</p> <p>0'46 pt 1.2, 1.3, 1.4</p> <p>0'49 pt TD</p> <p>0'51 aj TG après REP imbriqué</p> <p>0'55 soum 1 valide</p>	

### Louna, 2022t1p5v3 – profil ajustement

[lien vers l'extrait vidéo](#)

vtd5jwjs 6em

score global : 210

temps moyen par bloc : 6.6

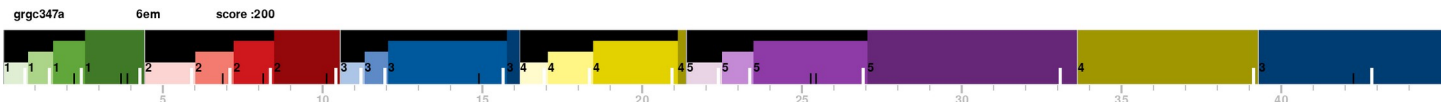
5 - Collecter les pierres [2022-1 white] medium durée : 2'16

ind. rapidité normalisée : 0.08

0 - 11'17    1 - 12'14 (0'57)    2 - 13'17 (1'03)    3 - 13'33 (0'16)

id sol : 969    3263    3264

### Louna, 2022 expérimentation complémentaire



### Louna, 2022t4p3v1 - profil expert

[lien vers l'extrait vidéo](#)

### Alexis, CM2 en 2021, 6ème en 2022

#### Alexis, 2021 tour 1



### Alexis, 2021t1p1v3 - profil ajustement

[lien vers l'extrait vidéo](#)

6pxvf7ce cm2

score global : 160

temps moyen par bloc : 10.1

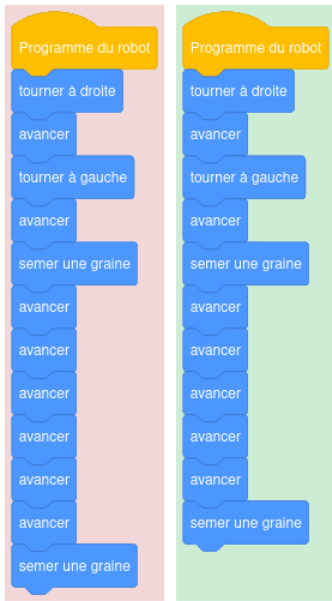
1 - Planter des fleurs [white 2021] medium durée : 1'10

ind. rapidité normalisée : -0.31

0 - 1'57

1 - 2'57  
1'00

2 - 3'07  
0'10



id sol :

4503

4504

**Alexis, 2021t1p3v1 – profil expert**

[lien vers l'extrait vidéo](#)

**Alexis, 2021t1p4v3 - profil accommodation**

[lien vers l'extrait vidéo](#)

6pxvf7ce cm2

score global : 160

temps moyen par bloc : 10.1

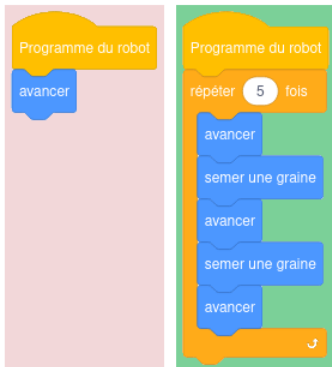
4 - Planter des fleurs [white 2021] medium durée : 8'21

ind. rapidité normalisée : 5.2

0 - 10'54

1 - 13'05  
2'11

2 - 19'15  
6'10


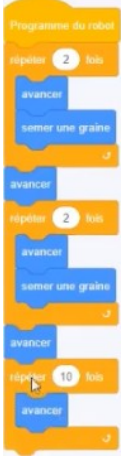



id sol :


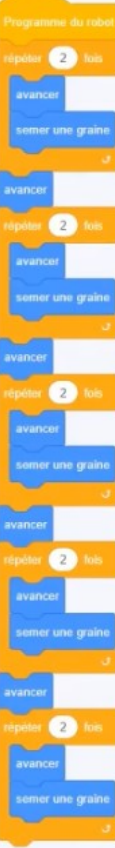
3646

885


	Description	Interprétation
--	-------------	----------------

	<p><b>2021t1s4v3_6pxvf7ce_CM2.mp4</b></p> <p>0'00 (9) accroche dans l'éditeur la séquence de blocs avancer, semer une graine, avancer, semer une graine, avancer, avancer, avancer, semer une graine, avancer, semer une graine</p> <p>(Obloc) s'arrête lorsque le message d'alerte sur la contrainte en blocs disponibles clignote en rouge</p> <p>(X) vide l'éditeur</p>	
	<p>0'19 (2) accroche un bloc répéter dans l'éditeur</p> <p>(3) place dedans les blocs avancer et semer une graine</p> <p>(4) met la valeur du compteur à 2</p> <p>(6) place un bloc avancer en-dessous du bloc répéter</p> <p>(2s) ajoute un bloc répéter à la fin du programme</p> <p>(4s) met la valeur du compteur de ce bloc à 2</p> <p>(3s) place dedans les blocs avancer et semer une graine</p> <p>(7g) parcourt la ligne de la grille avec les zones de terre</p> <p>(2s) ajoute un bloc répéter à la fin du programme</p> <p>(6) intercale un bloc avancer avant ce dernier bloc répéter</p> <p>(3s) place un bloc avancer à l'intérieur du dernier bloc répéter</p> <p>(Obloc) le message d'alerte indique qu'il n'y a plus de bloc disponible</p> <p>(X) vide l'éditeur</p>	<p>Le sujet est toujours sur un traitement séquentiel au niveau global, mais il traite les deux cases avec zone de terre ede manière itérative</p>
	<p>1'18 (lat) temps de latence : 10 secondes</p> <p>(2, X2) prend un bloc répéter sans le déposer – temps de latence : 6 secondes, le relâche au-dessus de la réserve</p> <p>(lat) temps de latence : 8 secondes</p> <p>(1pv) pointe les cases une à une en comptant (à peine audible) jusque la première case du 4ème groupe de deux zones de terre, s'arrête sur cette case</p> <p>(lat) temps de latence : 5 secondes</p> <p>(mvt) mouvements amples et désordonnés de la souris</p> <p>(9) accroche un bloc avancer dans l'éditeur</p> <p>(5a) soumission 1 → le robot se trouve sur la première zone de terre. Le message « Il reste de la terre vide de fleur ! » est affiché.</p>	



	<p>2'11 (X) vide l'éditeur</p> <p>(2) accroche un bloc répéter dans l'éditeur</p> <p>(3) place dedans les blocs avancer et semer une graine</p> <p>(2s) accroche un autre bloc répéter en-dessous</p> <p>(3) place dedans les blocs avancer et semer une graine</p> <p>(2s) place en-dessous 3 autres blocs répéter</p> <p>(X2s) supprime ces 3 blocs répéter</p> <p>(X) vide l'éditeur</p>	
	<p>2'44 (lat) temps de latence : 14 secondes</p> <p>(7pg) pointe la première case vide entre deux zones de terre</p> <p>(lat) temps de latence : 6 secondes</p> <p>(ExMT) E – <b>Qu'est-ce qui te gêne pour celui-là ?</b></p> <p>S – <b>Ben.. j'ai 10 blocs qui sont autorisés.. et là j'ai plus que 10 blocs</b> (pointe la dernière zone de terre)</p> <p>E – <b>Ok.. pour les manières que tu vois de résoudre le problème..</b></p> <p>S – <b>Je sais pas..</b></p> <p>E – <b>Tu aurais besoin de plus de 10 blocs</b></p> <p>S – <b>Ben oui.. parce que</b> (à peine audible)</p> <p>(lat) temps de latence : 7 secondes</p> <p>(ExMT) E – <b>Alors quelle manière tu voyais de résoudre le problème si tu avais plus de 10 blocs ?</b></p> <p>S – <b>avancer..</b></p> <p>(2, X2) accroche un bloc répéter dans l'éditeur puis le supprime</p> <p>(5d) S – <b>semer deux fois..</b> (pointe les zones de terre) <b>avancer, et resemer deux fois</b> (pointe les deux zones de terre suivante et continue en parcourant la fin de la ligne)</p> <p>E – <b>mm</b></p> <p>S – <b>Pour chaque tas de terre.. eh bien..</b></p> <p>(7pvg) pointe les deux premières zones de terre</p> <p>(ExMT) E – <b>D'accord.. alors tu vois bien.. quand tu dis chaque tas de terre.. c'est chaque tas ou chaque groupe de tas ?</b></p> <p>(1pv) S – <b>Chaque.. ah oui.. chaque groupe de tas..</b> (pointe le premier groupe puis parcourt la ligne) <b>ah oui y'en a cinq</b></p> <p>compte les zones de terre à mi-voix</p> <p>(2) accroche un bloc répéter dans l'éditeur</p> <p>(3) place dedans les blocs avancer et semer une graine</p>	

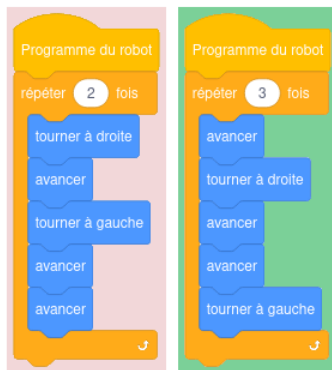
<p>(6) place un bloc avancer en-dessous du bloc répéter</p> <p>(2s) ajoute un bloc répéter à la fin du programme</p> <p>(3s) place dedans les blocs avancer et semer une graine</p> <p>(6) ajoute un bloc avancer en-dessous</p> <p>(2s) ajoute un bloc répéter à la fin du programme</p> <p>(3s) place un bloc avancer dedans</p> <p>(0bloc) le message indiquant qu'il n'y a plus de bloc disponible clignote</p> <p>(X3) supprime le dernier bloc avancer placé</p> <p>(ExRF) E – <b><i>Continue.. c'est pas grave.. tu vas pouvoir continuer à mettre tes blocs.. va jusqu'au bout de ta démarche</i></b></p> <p>(3s) place un bloc avancer et un bloc semer une graine dans le dernier bloc répéter</p> <p>(0bloc) le pop-up indiquant que le nombre de blocs autorisés est dépassé apparaît,</p> <p>(ExRF) E – <b><i>Voilà.. tu lui fais d'accord..</i></b></p> <p>referme la fenêtre pop-up</p> <p>(7pe) pointe le premier bloc répéter</p> <p>(ExRF) E – <b><i>Et tu continues quand même</i></b></p> <p>(7e) parcourt son programme</p> <p>(ExFP) E – <b><i>Par contre c'est peut-être pas répéter 10 fois</i></b></p> <p>S – <b><i>J'veis les changer</i></b></p> <p>E – <b><i>Oui ok</i></b></p> <p>(4) met la valeur du compteur des 3 blocs répéter à 2</p> <p>(6) ajoute un bloc avancer à la fin du programme</p> <p>(2s) ajoute un bloc répéter à la fin du programme</p> <p>(3s) place dedans les blocs avancer et semer une graine</p> <p>(7pg) pointe le groupe de deux zones de terre sur la grille</p> <p>(7pe) pointe les blocs répéter dans le programme</p> <p>(6) ajoute un bloc avancer à la fin du programme</p> <p>(2s) ajoute un bloc répéter à la fin du programme</p> <p>(3s) place dedans les blocs avancer et semer une graine</p> <p>(4s) place la valeur du compteur des deux derniers blocs répéter à 2.</p> <p>survole la zone de contrôle de l'exécution (les boutons ne sont pas visibles, remplacés par le message « 9 blocs en trop utilisés pour 10 autorisés »)</p> <p>(ExMT) E – <b><i>Voilà maintenant regarde bien ce que tu as fait</i></b></p> <p>(7e) parcourt son programme</p>	<p>Procédure de vérification entre le nombre de groupes de deux zones de terre et le nombre de blocs répéter</p>
---	--

	<p>(ExMT) E – <b>Ton but c’est quoi ?</b></p> <p>S – <b>C’est de lui faire planter une fleur sur chaque tas de terre</b></p> <p>E – <b>Oui.. donc ça c’est ce que tu dois faire faire au robot.. mais tu m’as dit tout à l’heure que.. ce qui t’embêtait c’est que tu n’avais que 10 blocs</b></p> <p>S – <b>Euh oui</b></p> <p>E – <b>Donc ton but c’est d’économiser des blocs</b></p> <p>S – <b>euh oui</b></p> <p>E – <b>Oui.. et qu’est-ce qui permet d’économiser des blocs ?</b></p> <p>(7pe) S – <b>Ben les boucles de répétition</b> (pointe le premier bloc répéter de son programme)</p> <p>(ExMT) E – <b>Voilà.. les boucles de répétition.. exactement.. est-ce que.. dans ce que tu as fait.. y’a pas des choses qui se répètent ?</b></p> <p>S – <b>Euh si.. avancer.. c’est que une fois entre chaque.. ah si.. si je peux..</b></p> <p>(X2s) 7’30 décroche le sous-programme à partir du 2ème bloc répéter</p>	
	<p>7’30 (X6, 3) déplace le bloc avancer qui est sous le premier bloc répéter dans ce bloc répéter</p> <p>(7pg) pointe les deux premières zones de terre et la case vide suivante</p> <p>(4) met la valeur du compteur de boucle à 5</p> <p>supprime le sous-programme décroché et resté en bas de l’éditeur</p> <p>(7pg) pointe les deux premières zones de terre</p> <p>(7pe) pointe le dernier bloc avancer à l’intérieur du bloc répéter</p> <p>(3) ajoute les blocs semer une graine et avancer à la séquence dans le bloc répéter</p> <p>(1pv) pointe les groupes de deux zones de terre en les comptant à mi-voix</p> <p>pointe les cases vides entre deux groupes en les comptant à mi-voix, s’arrête plusieurs secondes sur la 4ème, puis pointe la dernière case de la ligne après le dernier groupe</p> <p>(7pe) pointe la valeur dans le compteur du bloc répéter</p> <p>(5a) soumission 2 valide</p>	<p>procédure de contrôle</p>

**Alexis, 2021t1p5v1 – profil accommodation**

[lien vers l’extrait vidéo](#)

0 - 19'27

1 - 27'19  
7'522 - 29'23  
2'04

id sol :

4506

888

intéressant à transcrire :

verbalisation

début en séquence

la limite de blocs empêche de mettre en œuvre des procédures de vérification

la représentation mentale de l'exécution est en place, mais défaillante à certains moments, ce qui empêche d'identifier le motif à partir du motif algorithmique.

étayage pour comprendre le tutoriel

—

le pointeur suit l'énoncé lors de la lecture

Alexis – **Donc.. avancer** (ajAV)

pr-rel TG, ajTD, ajAV, ajTG, ajAV

survole brièvement la grille

ajTD, ajAV

temps d'arrêt, pointe TG dans le programme, pointe les deux premiers diamants sur la grille

Alexis – **Avancer avancer** (chuchoté)

ce qui est chuchoté ne correspond pas aux blocs placés dans le programme

pointe le robot puis les deux premiers diamants, s'arrête sur le deuxième, pointe le deuxième avancer du programme

à reprendre à un niveau plus général : la représentation mentale de l'exécution consiste dans notre contexte à faire correspondre les actions verbalisées en même temps que les cases pointées avec la souris avec les blocs placés dans le programme. Alexis n'y parvient pas correctement

aj TG

pointe les deux premiers diamants, s'arrête à nouveau sur le deuxième

ces temps d'arrêt plus marqués pourraient indiquer qu'Alexis sent qu'il y a un souci à cet endroit, sans être en mesure de le corriger

entoure la séquence à l'exception des deux premiers blocs avec un bloc REP → opo-up trop de blocs

parcourt le corps de la boucle

ne laisse qu'un bloc AV dans la boucle, place le reste en-dessous

comp 2

l'utilisation de la boucle montre qu'Alexis avait bien repéré le problème, mais l'utilisation de la boucle pour le nombre de déplacement ne permet pas de satisfaire la contrainte de blocs.

parcourt le trajet du robot jusqu'à la troisième pierre, pointe les deux premières pierres puis la case vide en-dessous, la troisième pierre, la quatrième

entoure les trois derniers blocs de la séquence avec REP, se reprend et entoure seulement le bloc AV au-dessus

comp 2

parcourt le trajet du robot jusqu'à la cinquième pierre

accroche les trois blocs en-dessous du REP

dans l'état, le programme édité ne correspond pas au parcours pointé sur la grille

pointe le cinquième bloc

pointe le dernier bloc du programme : TG

pointe les cases du trajet jusqu'à la troisième pierre en chuchotant (inaudible)

pointe TD après le deuxième REP

intercale deux blocs AV avant le deuxième REP

temps de latence : 12s

pointe à nouveau les cases du trajet jusqu'à la troisième pierre en chuchotant (inaudible)

supprime le dernier bloc AV ajouté et le remplace par un bloc TD

Alexis reprend un programme en séquence à partir du milieu du programme édité

pointe toutes les cases du trajet jusqu'à la dernière pierre

supprime le bloc TG de la fin du programme

pointe le premier bloc REP

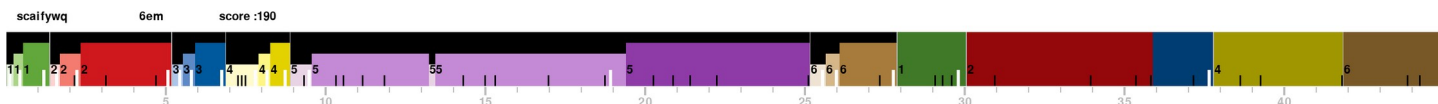
temps de latence : 20s

pointe toutes les cases du trajet du robot jusqu'à la case en-dessous de la dernière pierre, pointe la cinquième pierre

ajTG, ajAV

**Nous faisons l'hypothèse que les erreurs de sens de rotation empêchent Alexis d'identifier le motif algorithmique. Comme Alexis ne peut pas exécuter son programme (blocage due à la limite de blocs), il n'est pas en mesure d'utiliser les fonctionnalités de l'environnement de programmation pour identifier son erreur, en particulier le mode pas à pas.**

**D'un côté le mode pas à pas serait nécessaire pour aider à déboguer, le mode pas à pas est inaccessible au moment où il serait le plus nécessaire. Est-ce qu'il pourrait être envisagé de laisser actif seulement le mode pas à pas lorsque la limite de blocs est dépassée ? Cela irait dans le sens de limiter la stratégie en essai-erreur, tout en laissant possible la manipulation nécessaire à la compréhension, et éviter le blocage.**



### Alexis, 2022t2p4v1 – profil accommodation

[lien vers l'extrait vidéo](#)

scaifywq 6em

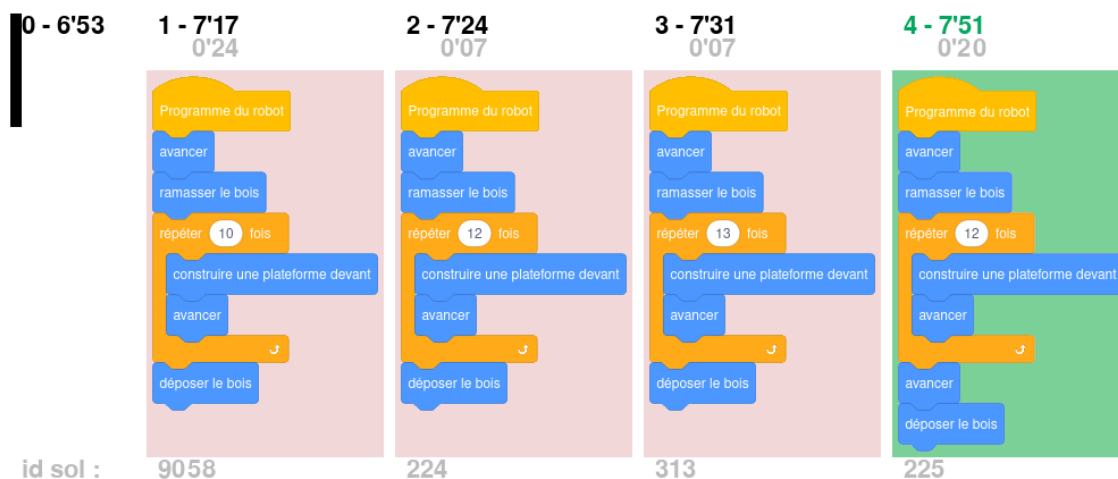
score global : 190

temps moyen par bloc : 3.7

4 - Chauffer le château [2022-2 white] basic

durée : 0'58

ind. rapidité normalisée : 0.74



224 'scaifywq': 'accommodation'}"]],

313 'scaifywq': 'accommodation'}"]],

Alexis manie la souris avec beaucoup de dextérité, son temps moyen pour placer un bloc est l'un des plus courts : 3,7 secondes.

ajAV, ajRB, ajREP, repCST, repAV, ajDB sans pause, puis lance l'exécution (id. 8410).

La soumission suivante (id 222) un **AJUSTEMENT** de l'oubli de renseigner le nombre d'itérations.

Comme Alexis a placé un bloc avancer dans la boucle après le bloc construire un plateforme devant dès le premier essai, nous pouvons supposer que la représentation mentale de l'exécution associée à ce bloc est au moins partiellement construite. Pourtant, il faut ensuite deux soumissions à Alexis pour corriger l'erreur en sortie de boucle.

Pour la soumission d'id 224, le robot dépose le bois une case avant la cheminée. Le schème qu'Alexis a probablement activé alors qu'il n'est pas pertinent ici est « mettre un de plus pour aller une case plus loin ». Ce schème est en lien avec l'unité considérée pour le dénombrement. Quand Alexis ajoute un, que comprend ce un ?

Après la soumission d'id 313, Alexis mobilise le mode pas à pas pour trouver l'erreur, signe que l'erreur ne relève pas d'une erreur de mise en œuvre, mais de la mobilisation d'un schème non pertinent.

Alexis maîtrise déjà la notion de boucle dans des situations beaucoup plus complexes. Ce qui a posé problème à Alexis est une **difficulté dans la représentation de la position du robot à l'issue de l'exécution de la boucle**. La confrontation relève donc bien du profil **ACCOMMODATION**.

### Alexis, 2022t2p5v2 – profil accommodation

[lien vers l'extrait vidéo 1](#)

## [lien vers l'extrait vidéo 2](#)

scaifywq 6em

score global : 190

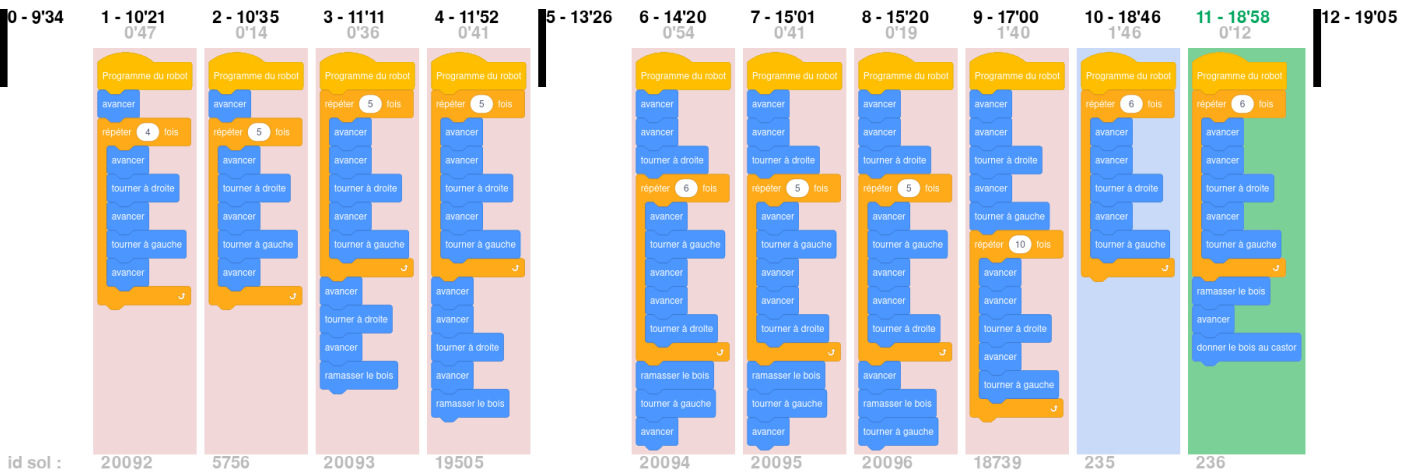
temps moyen par bloc : 3.7

5 - Distribuer le bois [2022-2 white]

easy

durée : 7'50

ind. rapidité normalisée : 10.55



**passage du programme d'id.20093 au programme d'id.19505 (soumission 3 à soumission 4) : inclus le bloc AV qui était avant la boucle dans la boucle, et enlève celui qui est à la fin du corps de la boucle. Cette action montre que le sujet a compris l'aspect cyclique de l'exécution de la boucle, la jonction entre une itération et la suivante, qu'on peut décaler le motif afin d'obtenir un motif optimal.**

**Soumission 4 : le sujet n'est pas en mesure de finir son programme à cause de la limite de blocs. Sur la case du tas de bois, deux actions doivent être exécutées par le robot : ramasser le bois et tourner à gauche. L'ordre où ces instructions sont exécutées n'importe pas si le programme est en séquence. Mais placer d'abord le bloc ramasser le bois empêche Alexis de voir une sixième occurrence du motif algorithmique, qu'il est nécessaire d'intégrer à la boucle.**

vide l'éditeur et régresse vers une séquence, avant de vider à nouveau l'éditeur

retourne consulter le tutoriel de la version 1

stratégie de dérogage lors de la soumission 6 : démarrer l'exécution à vitesse normale, et passer en pas à pas un peu avant l'endroit de l'erreur

**E – Alors là ce serait bien que tu réfléchisse tout haut pour moi**

comp 5

**Alexis – Alors le truc c'est qu'il ne me reste plus que zéro bloc**

E – Mm

**Alexis - Et.. il me manquait juste un bloc pour donner le bois au castor**

soumissions 7 et 8

**E – Et alors qu'est-ce que tu pourrais faire ?**

Alexis – Euh

Temps de latence : 14s

vide l'éditeur

simule le parcours du robot avec le pointeur

soumission 9 : édite un premier motif hors de la boucle, puis le même motif dans la boucle

temps de latence : 30s

E – *Alors.. qu'est-ce que tu te dis là ?*

Alexis – *Ben j'essaie de trouver un moyen qui permet d'avoir assez de.. de blocs pour pouvoir ramasser le bois et pour pouvoir après le donner au castor.*

Le but ici est de modifier le programme pour satisfaire la contrainte de blocs. La contrainte induit en quelque sorte un resserrement du but. Non seulement, il faut que le robot effectue la tâche prescrite, mais il faut aussi « économiser » les blocs.

E – *Mm.. et dans ce que tu as déjà fait ?.. dans le programme que tu as mis là dans l'éditeur ?*

temps de latence : 10s

E – *Donc on est bien d'accord que ton problème c'est qu'il te manque des blocs.. et quelle est ta stratégie pour essayer d'en économiser ?*

Alexis – *Ben de voir ce qui se répète..*

E – *Mm*

Alexis – *Pour éviter que ça se répète*

**Alexis amène le motif placé avant la boucle à côté du corps de celle-ci. Il met en regard deux occurrences du même motif avant de fusionner.**



comp 6

soumission 10, programme d'id.235, qui correspond au programme non valide le plus fréquent.

Il permet à Alexis de repérer la position du robot après exécution de la boucle, et de finir son programme par complétion.

E – *Ok.. donc qu'est-ce qui t'avait gêné ?*

Alexis – *Euh ben qu'j'avais.. j'avais remarqué.. que là y'avait que deux blocs ici (pointe les deux cases devant le robot) et que là y'en avait trois (parcourt la deuxième marche) mais j'm'étais pas dit que j'pouvais faire comme ça.. j'm'étais dit qu'il fallait d'abord que je fasse celui-là (parcourt les deux cases devant le robot) avant de pouvoir faire tous les autres mais..*

E – *Mm.. ok*

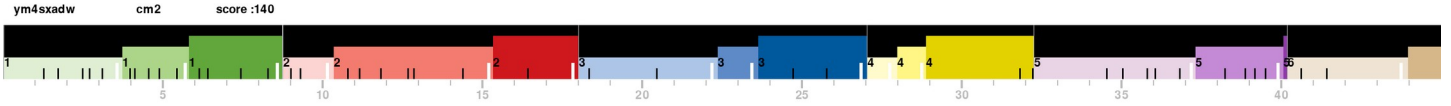
**Le fait que le premier motif n'est visuellement pas identique aux autres à cause de la présence du robot a amené Alexis à le considérer à part. Alexis prend en compte le motif visuel indépendamment de la présence du robot. Il ne translate pas le robot sur la première case de la deuxième marche lorsqu'il simule mentalement l'exécution.**

**La stratégie qui consiste à lancer une exécution pour repérer la position du robot à l'issue du programme déjà édité est très répandue. Les endroits de prise de repère, c'est à dire la décomposition du problème, évoluent avec l'expertise du sujet. Nous pouvons repérer des « endroits types ». Déléguer à la machine de déterminer la position du robot à l'issue de l'exécution d'une boucle est un de ces endroits types.**



# Louane, CM2 en 2022

## Louane, 2022 tour 1



### Louane, 2022t1p1v1 - profil accommodation

[lien vers l'extrait vidéo](#)

ym4sxadw cm2

score global : 140

temps moyen par bloc : 11.1

1 - Chauffer le château [2022-1 white] basic

durée : 3'35

ind. rapidité normalisée : 3.84

0 - 0'02

1 - 1'18

1'16

2 - 1'45

0'27

3 - 2'31

0'46

4 - 2'44

0'13

5 - 3'08

0'24

6 - 3'37

0'29



id sol :

23921

23922

87

947

2213

88

### Louane, 2022t1p1v2 - profil accommodation

[lien vers l'extrait vidéo](#)

ym4sxadw cm2

score global : 140

temps moyen par bloc : 11.1

1 - Chauffer le château [2022-1 white] easy

durée : 2'01

ind. rapidité normalisée : 0.56

0 - 3'44

1 - 4'00

0'16

2 - 4'09

0'09

3 - 4'35

0'26

4 - 4'55

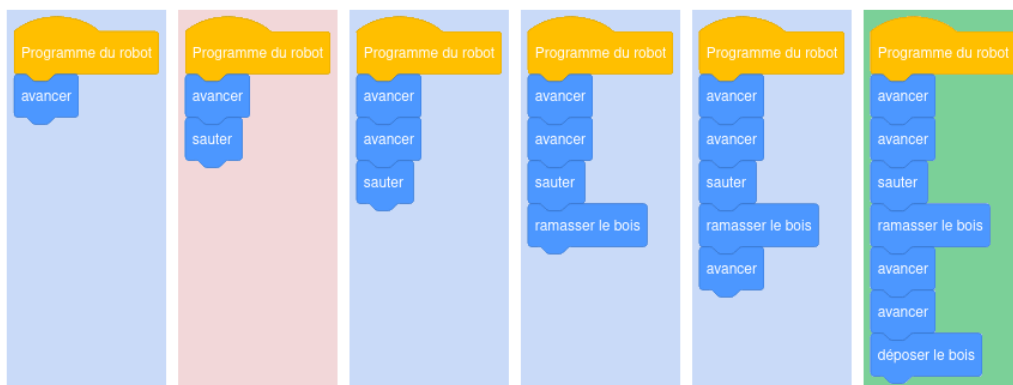
0'20

5 - 5'28

0'33

6 - 5'45

0'17



id sol :

87

1056

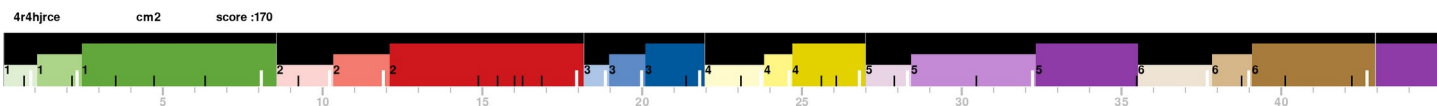
1057

8577

3425

92

## Louane, 2022 tour 2

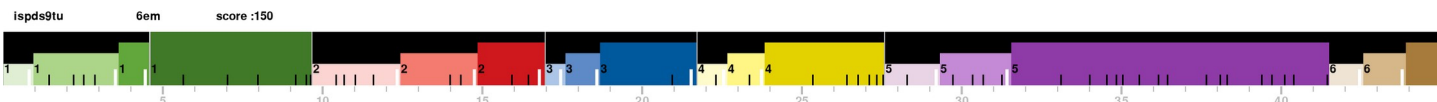


## Louane, 2022t2p4v2 – profil expert

[lien vers l'extrait vidéo](#)

## Colin, CM2 en 2021, 6<sup>ème</sup> en 2022

### Colin, 2022 tour 1



## Colin, 2022t1p1v2 – profil accommodation

[lien vers l'extrait vidéo](#)

ispds9tu 6em

score global : 150

temps moyen par bloc : 8.3

1 - Chauffer le château [2022-1 white] easy

durée : 2'37

ind. rapidité normalisée : 1.7

0 - 0'57

1 - 1'28  
0'31

2 - 2'14  
0'46

3 - 2'33  
0'19

4 - 2'54  
0'21

5 - 3'34  
0'40



id sol :

1056

4618

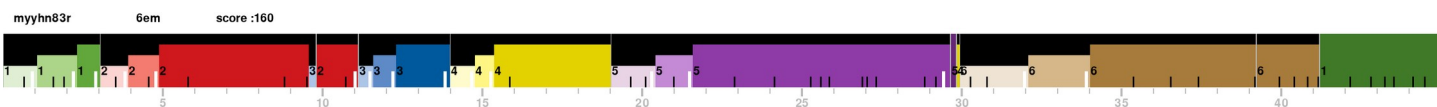
14615

727

92

## Sam, 6<sup>ème</sup> en 2022

### Sam, 2022 tour 1



## Sam, 2022t1p1v1 – profil accommodation

[lien vers l'extrait vidéo](#)

myyhn83r 6em

score global : 160

temps moyen par bloc : 5.4

1 - Chauffer le château [2022-1 white] basic

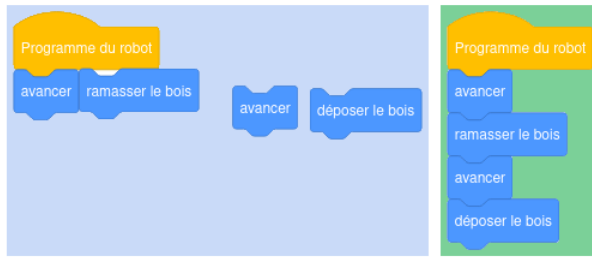
durée : 0'57

ind. rapidité normalisée : 1.64

0 - 0'01

1 - 0'41  
0'40

2 - 0'58  
0'17



id sol : 17196

88

### Sam, 2022t1p4v1 - profil indéfini

[lien vers l'extrait vidéo](#)

myyhn83r 6em

score global : 160

temps moyen par bloc : 5.4

4 - Planter des fleurs [2022-1 white] basic

durée : 0'43

ind. rapidité normalisée : 0.59

0 - 14'00

1 - 14'43  
0'43

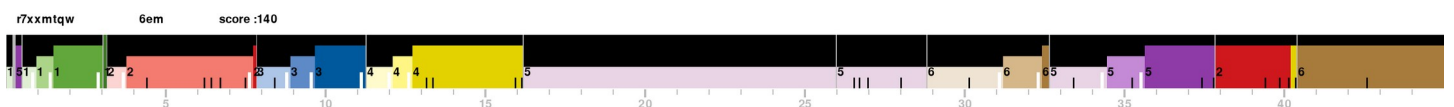


id sol : 760

Première session Algorea

	Description	Interprétation
	<p><b>2022t1s4v1_myyn83r_6e.mp4</b></p> <p>0'00 mvt</p> <p>0'05 aj AV,SG, AV</p> <p>0'15 pr-rel SG</p> <p>0'18 sup AV2</p> <p>0'20 entoure AV,SG avec REP</p> <p>0'29 ouvre comp, lat 7s, comp 8</p> <p>0'38 soum 1 valide</p>	<p>Sam commence en séquence, et se ravise pour utiliser le bloc répéter. Cette hésitation fait basculer le profil de confrontation dans la catégorie indéfini</p>

### Sam, 2022 tour 2



### Sam, 2022t2p4v3 - profil échec

## [lien vers l'extrait vidéo](#)

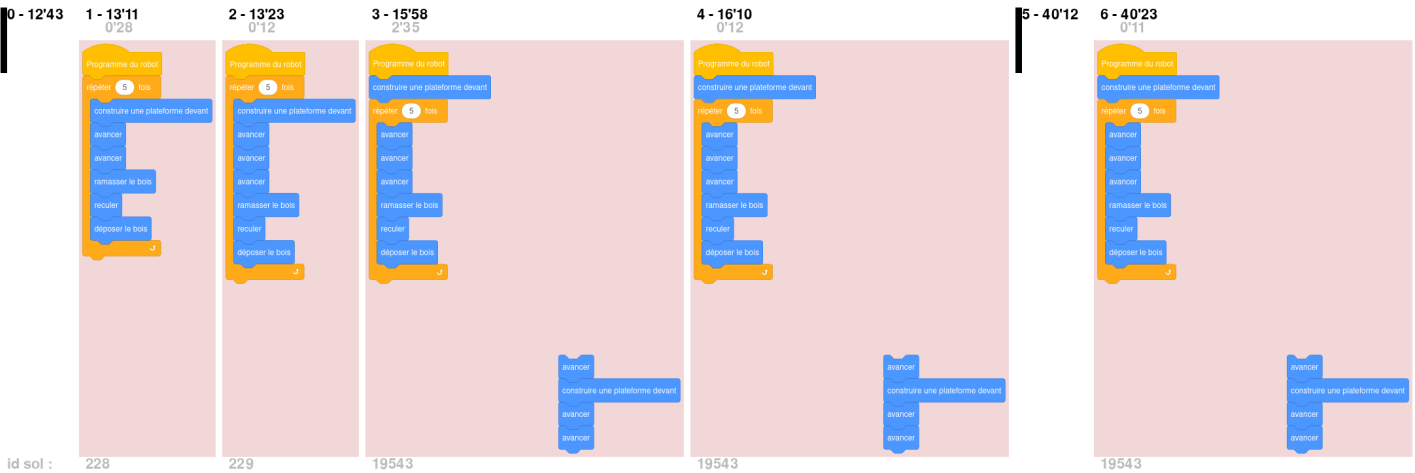
r7xxmtqw 6em

score global : 140

temps moyen par bloc : 5.3

4 - Chauffer le château [2022-2 white] medium durée : 3'38

ind. rapidité normalisée : 2.74



ajCST, ajAV, ajAV, ajRAM, ajREC, ajDEP

pointe la case devant le robot, le premier tas de bois, les cheminées suivantes

entoure la séquence avec REP

comp 5

exécution du programme d'id.228 → le robot s'arrête sur la première cheminée, feedback : il n'y a rien à ramasser !

survole la grille, pointe le deuxième AV du programme, pointe le robot en position à l'issue de l'exécution, pointe l'emplacement initial du robot, ajoute un AV avant les deux autres

exécution du programme d'id. 229 → (erreur la plus fréquente pour ce puzzle) → le robot a déposé le premier tas de bois dans la cheminée, feedback : il y a déjà une plateforme ici

pointe CST dans le programme, décroche la séquence, puis CST qu'il met avant la boucle

→ **débugage local qui ne prend en compte que l'action suivante que le robot doit faire : le bloc CST gêne pour que le robot puisse avancer donc Sam le place avant la boucle**

exécution en pas à pas jusqu'à ce que déposer le bois soit surligné, temps d'arrêt, pp début deuxième itération jusqu'à ce que le robot soit au bord du vide et le deuxième AV surligné, temps d'arrêt

→ **exécution tout le temps que le robot effectue les actions attendues. La stratégie est efficace pour identifier l'erreur mais pas pour la corriger. Il faudrait une phase d'analyse supplémentaire sur la cause de l'erreur, qui prenne en compte que l'erreur est située au sein d'une boucle. La stratégie de débogage avec le mode pas à pas est efficace pour une séquence, mais elle est mise en échec pour certaines erreurs liées à la boucle.**

décroche la séquence, décroche REP, accroche la séquence au début du programme, pr-rel le deuxième bloc AV avec la séquence en-dessous, remplace CST en début de séquence,

comp 4, accroche REP en-dessous de la séquence

repAV, repAV

→ Sam a placé le premier motif algorithmique avant la boucle, afin d'inclure le remplacement du robot en début de boucle. Sa stratégie est mise en échec par le nombre de blocs disponibles.

pointe la première cheminée, le premier tas de bois

intercale CST après le premier AV de la boucle

pointe le premier tas de bois, la deuxième case vide

repAV,

prend un autre AV, message trop de blocs, rel AV

pointe le début du programme, le robot, la première case vide, le robot, la deuxième cheminée

enlève la séquence de REP et la laisse dans l'éditeur, entoure AV AV RAM REC DEP avec REP, met aussi le premier AV dans REP

comp 5

Exécution du programme d'id.19543 → le robot a déposé le premier tas de bois et a avancé d'une case, feedback : le robot va se jeter dans le vide !

quitte le puzzle

debrief

E – ***Ta solution la plus proche c'était ça..*** (édite le programme d'id.229) ***et là quand je lance..*** (exécute le programme) ***voilà.. quelle est l'erreur ?***

Sam – ***Heu..***

remet en situation initiale

Sam – ***J'ai fait un parcours trop court..***

E – ***Oui..***

Sam – ***Répéter cinq fois il faut plus de choses dedans..***

E – ***Oui.. quoi en plus ?*** (temps de silence 10s) ***qu'est-ce qui te fait dire qu'il faut plus de choses dedans ?***

Sam – ***Parce que.. heu.. on peut pas faire une plateforme quand on est sur la cheminée et devant y'a une plateforme.. donc..***

E – ***Donc ?***

Sam – ***Donc il faut mettre un.. avancer.. devant construire une plateforme devant***

E – ***Mm si tu mets un avancer devant..*** (prend un avancer et survole la première case vide avec) ***il va .. la première fois.. il va se jeter dans le vide..***

Sam – ***Oui***

E – ***C'est pas là qu'il faut le mettre..***

Sam (soupir) – ***Déjà pour gagner un bloc.. au lieu de mettre trois avancer on pourrait mettre.. on pourrait mettre avancer trois***

E – ***Oui ça exact.. mais c'est pas ça le problème.. t'auras assez de blocs.. mets le mode pas à pas*** (première itération en pp) ***qu'est-ce que ce serait bien que je fasse là ?***

Sam – ***Bah après on.. bah c'est qu'on peut pas faire c'que j'veux..***

E – ***Oui mais qu'est-ce que je voudrais qu'il fasse le robot ?***

Sam – ***Qu'il avance encore***

E – ***Mm***

Sam – *Et qu'il répète la même démarche..*

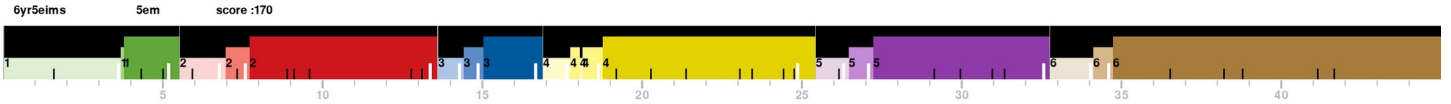
E – *Et donc le avancer j'le mets où ?*

Sam – *Après le déposer le bois*

repAV et exécution du programme valide

## Timéo, 5ème en 2022

### Timéo, 2022 tour 1



### Timéo, 2022t1p1v1 – profil accommodation

[lien vers l'extrait vidéo](#)

6yr5eims 5em

score global : 170

temps moyen par bloc : 4.6

1 - Chauffer le château [2022-1 white]

basic

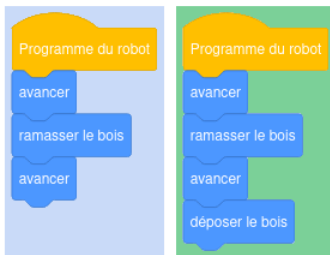
durée : 3'38

ind. rapidité normalisée : 10.85

0 - 0'02

1 - 1'37  
1'35

2 - 3'40  
2'03



id sol :

2214

88

### Timéo, 2022t1p4v3 – profil accommodation

6yr5eims 5em

score global : 170

temps moyen par bloc : 4.6

4 - Planter des fleurs [2022-1 white]

medium durée : 6'09

ind. rapidité normalisée : 7.02

0 - 18'46

1 - 19'13  
0'27

2 - 20'18  
1'05

3 - 21'24  
1'06

4 - 23'05  
1'41

5 - 23'28  
0'23

6 - 24'27  
0'59

7 - 24'47  
0'20

8 - 24'55  
0'08



id sol :

140

4607

3646

4197

4608

4609

1068

1069

Description



### 2022t1s4v3\_6yr5eims\_5e.mp4

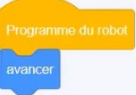

0'00 aj REP  
 0'02 pt 1,2,T1 (chuchoté 1,2,3), T2, T1 à T5  
 0'08 rep AV,AV,AV,SG  
 0'16 comp 5  
 0'19 soum 1 → « Votre robot a semé hors des zones de terre ».



0'26 S – **Ah.. c'est pas la.. j'ai pas vu**  
 0'29 ferme feedback  
 0'30 vide l'éditeur (difficultés de manip)  
 0'37 recentre prRob  
 0'40 S – **Trois..trois..trois..trois** (pt T1 à T5).. (pt 1) **ah oui**  
 0'47 aj AV,AV,SG  
 0'50 pr-rel REP, aj REP  
 0'53 comp 4  
 0'55 rep AV,AV,AV,SG,0bloc  
 1'02 pr-rel SG2  
 1'06 sup AV4,AV5,SG2  
 1'07 aj REP  
 1'09 rep2 AV  
 1'12 sup AV4 de rep2  
 1'13 sup rep2  
 1'13 rep AV,AV,SG, 0bloc  
 1'20 S - (pt 0bloc) **Et là ça veut dire que je ne peux plus rien mettre ?**  
 1'23 E – **Voilà c'est ça.. c'est ça que ça veut dire**  
 1'26 soum 2 → « Votre robot a semé hors des zones de terre ».  
 1'29 S - Non



1'33 S – **D'accord**  
 1'34 ferme feedback  
 1'36 sup REP et ce qu'il contient  
 1'38 pr-rel REP  
 1'41 aj AN après AV1  
 1'43 pt SG  
 1'45 pt Robot, T2  
 1'47 aj REP  
 1'49 parcourt 3 à T2  
 1'52 S - (pr prog) **eah..** (vide l'éditeur)

	<p>1'58 pr-rel REP</p> <p>2'00 pt robot, 1, 2 T1, 3 à T2, robot, parcourt 1 à T1, pt 3 à T2, 6, 7</p> <p>2'10 aj AV,AV,AV,SG</p> <p>2'16 aj REP</p> <p>2'18 S- (rep AV) <b>Mais..</b> (rep AV,AV, 0bloc) <b>hei..mm</b></p> <p>2'26 pt curseur de vitesse</p> <p>2'28 vide l'éditeur</p>
	<p>2'31 E – <b><i>Là je vois bien que tu aurais une solution si tu avais plus de blocs..</i></b></p> <p>2'37 aj AV</p> <p>2'38 E – <b><i>..mais ce qu'on cherche c'est une solution avec seulement 8 blocs</i></b></p> <p>2'39 soum 3 → « Il reste de la terre vide de fleur ! »</p>
	<p>2'41 ferme feedback</p> <p>2'44 pr-rel AV</p> <p>2'46 vide l'éditeur</p> <p>2'49 S – <b><i>Mm</i></b> (pt REP)</p> <p>2'50 parcourt robot à T1</p> <p>2'52 pt REP</p> <p>2'53 aj AV,AV,AV,SG</p> <p>2'59 entoure prog avec REP</p> <p>3'03 comp 5</p> <p>3'11 rep AV après AV1</p> <p>3'12 pt robot, 1, 2, 3, 4, 3, 2, T1</p> <p>3'16 pr-rel AV4,SG1</p> <p>3'19 pt 1, 5</p> <p>3'21 pr-rel contenu du bloc répéter</p> <p>3'24 pt AV2</p> <p>3'27 pr-rel SG1</p> <p>3'28 pr-rel AV3,AV4,SG1 (deux fois)</p> <p>3'31 sup SG1</p> <p>3'32 vide l'éditeur</p>







3'34 S – **Euh..ah..**  
 3'36 pt AV  
 3'38 pr-rel REP  
 3'39 aj AV,AV,AV,SG  
 3'44 pr-rel REP  
 3'46 pt AV3  
 3'47 entoure prog avec REP  
 3'49 comp 3  
 3'51 S – **Ah je sais**  
 3'54 sup AV2,AV3,SG1  
 3'55 aj SG après REP  
 3'56 aj REP  
 3'59 rep2 AV,AV,AV,SG  
 4'16 soum 4 → « Le robot sort de la grille ».



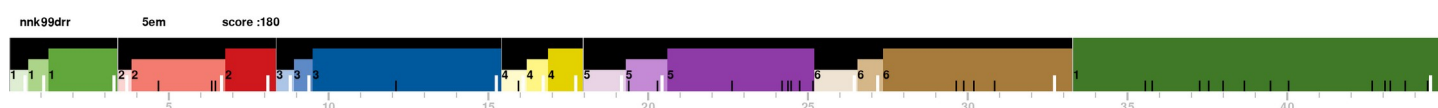
4'19 S – **Euh ouais**  
 4'21 pt les deux premières fleurs (grille après exécution)  
 4'23 ferme feedback  
 4'25 détache SG1 et ce qui est en-dessous et le place dans REP1  
 4'31 reprend REP2 et le met en séquence  
 4'32 parcourt corps de REP2  
 4'33 mvt lents  
 4'37 comp 1 pour REP1  
 4'39 soum 5 → « Le robot sort de la grille »



4'42 S – **Euch (mvt)**  
 4'46 pt fleur en 1, 0  
 4'48 rep1 AV, pop-up 0 bloc de suite refermé  
 4'52 sup rep2 et ce qu'il contient  
 4'55 parcourt 1 à T1  
 4'57 pr AV, l'amène avant AV1 puis le relâche  
 5'00 met SG1 après REP  
 5'01 sup AV2  
 5'02 comp 3  
 5'04 aj REP  
 5'07 pt T1, 3 à T2  
 5'10 comp 4  
 5'13 rep AV

	<p>5'17 entoure AV2 avec REP</p> <p>5'20 rep2 SG</p> <p>5'24 ouvre et referme comp REP imbriqué</p> <p>5'27 comp 6 pour rep2</p> <p>5'30 ouvre comp REP imbriqué, pt T1</p> <p>5'33 S – <b>Euh</b> (comp 4, survole brièvement la grille)</p> <p>5'37 soum 6 S – <b>Oui j'ai réussi</b> (→ « Le robot sort de la grille »).. <b>Mm</b></p>
	<p>5'42 S – <b>Ah pardon il sort de la grille</b></p> <p>5'44 pt comp rep2, comp rep imbriqué</p> <p>5'48 E – <b>Mais y'a de l'idée</b></p> <p>5'50 comp 5 mais ne referme pas comp, pt T1 à T5, ferme comp</p> <p>5'58 soum 7 → « Le robot sort de la grille »</p>
	<p>6'02 comp 4</p> <p>6'05 soum 8 valide</p> <p>6'10 E – <b>Alors comment tu expliques que c'est quatre et pas cinq ?</b></p> <p>6'13 S – <b>Parce qu'en fait y'avait déjà un se.. (pt) le programme.. le bloc du haut qui faisait déjà..</b></p>

## Timéo, 2022 tour 2



### Timéo, 2022t2p5v1 – profil expert

[lien vers l'extrait vidéo](#)

verbalisation

lecture de l'énoncé à voix haute mais en mangeant des syllabes

Timéo – **D'accord.. avanc..** (pr-rel AV)

dénombrer à voix haute jusque 5 en marquant des temps d'arrêt sur la grille, mais les endroits ne correspondent pas systématiquement à des éléments homogènes (castor ou bois). En particulier, Timéo dit « un » au moment où il pointe le robot. Compte-t-il le parcours des motifs ?

ajREP

Timéo – **Cinq fois donc** (comp 5) **cinq fois..** [incompréhensible] **cinq fois..**

survole brièvement la grille

Timéo – **Avancer** (repAV) **tourner à droite** (repTD)

pointe la case avec le tas de bois

Timéo – (repAV) **ramasser le bois** (repRAM)

pointe à nouveau brièvement la case avec le tas de bois

Timéo – **avancer avancer ramasser le bois encore avancer** (repAV) **donner le bois au castor** (DEP) **tourner à gauche** (repTG) **et normalement..** (pointe le robot) **j’crois que c’est bon**

**débuté l’exécution du programme d’id.233 en mode pas à pas jusqu’au milieu de la deuxième itération, en marquant un temps d’arrêt à la fin de la première itération**

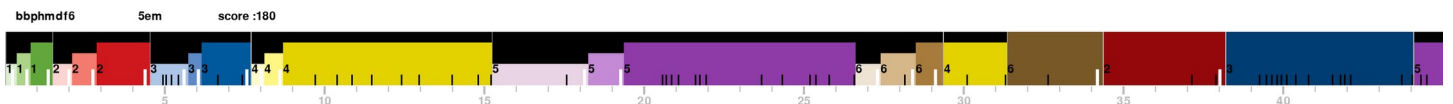
Timéo – **Oui c’est bon**

finit l’exécution à vitesse normale

Timéo – **J’crois que c’est bon.. oui c’est bon**

**Nous pouvons observer la procédure de vérification attachée à la bonne jonction entre les deux premières itérations.** Timéo induit qu’ensuite ce sera le cas pour celles qui suivent, mais sa verbalisation indique qu’il n’en est pas tout à fait sûr.

## Timéo, 2022 tour 3



### Timéo, 2022t3p3v1 – profil accommodation

[lien vers l’extrait vidéo](#)

bbphmdf6 5em

score global : 180

temps moyen par bloc : 6.7

3 - Pousser les caisses [2022-3 white]

basic

durée : 1'06

ind. rapidité normalisée : 1.46

0 - 4'33

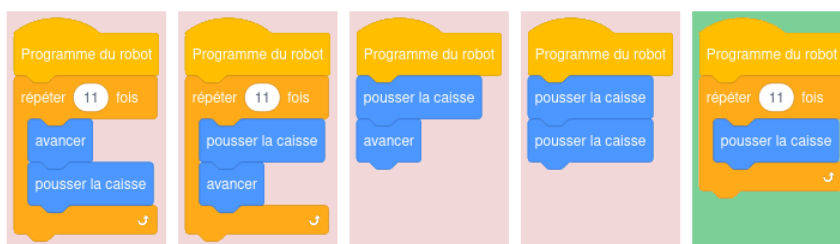
1 - 4'58  
0'25

2 - 5'04  
0'06

3 - 5'14  
0'10

4 - 5'27  
0'13

5 - 5'39  
0'12



id sol :

365

366

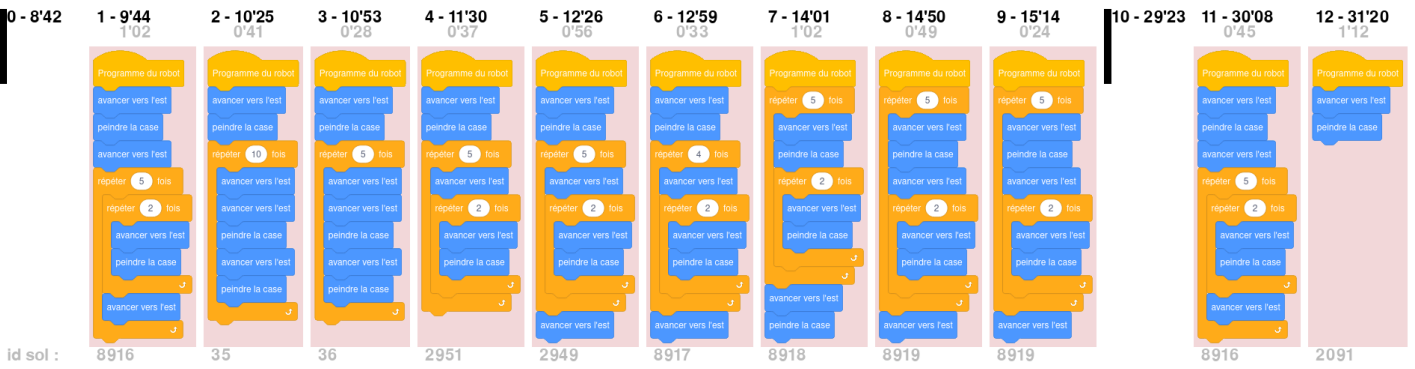
4981

8357

12

### Timéo, 2022t3p4v3 – profil échec

[lien vers l’extrait vidéo](#)



Timéo – **Version trois étoiles.. avancer vers l'est (ajEST) peindre la case (ajPC)**

pointe le premier point noir puis les groupes de deux points

recommence à pointer les groupes de deux points noirs en les numérotant verbalement

ajREP

comp 5

Timéo – **Avancer vers l'est (repEST) peindre la case (repPC) avancer vers l'est (repEST) peindre la case (repPC)**

supprime les deux derniers blocs de la séquence, ajoute un bloc REP imbriqué dans le premier qui entoure les blocs REP et EST déjà placés

comp 2

pointe le robot, le 1<sup>er</sup> point, le robot, le 1<sup>er</sup> point, le 2<sup>ème</sup> point

Timéo – **Ok..**

intercale un bloc EST juste avant la boucle

pointe le 1<sup>er</sup> point, le 2<sup>ème</sup>, le 3<sup>ème</sup>

Timéo – **Est.. puis ensuite avancer vers l'est (repEST dans la boucle externe)**

Exécution du programme d'id.8916 → le robot s'arrête sur la case juste avant le dernier point, feedback : Votre robot n'a pas peint les bonnes cases.

Timéo – **Ok.. donc..**

décroche le programme, décroche les boucles, puis vide l'éditeur

Timéo – **Alors..**

ajEST, ajPC, ajREP

Timéo – Vers l'est (repEST)

repEST, repPC, repEST, repPC

exécution du programme d'id.35 → le robot a peint toutes les cases, feedback : le robot sort de la grille

ferme le feedback, met le curseur de vitesse à plus rapide, relance l'exécution du même programme et interrompt lorsque le robot est sur la dernière case

Timéo – **peindre la case.. ah oui..**

pointe le compteur, pointe le premier point peint, l'emplacement du robot lorsqu'il est en situation initiale

pointe les groupes de deux cases peintes en verbalisant

comp 5

exécution du programme d'id. 36 → le dernier point n'est pas peint, feedback : Votre robot n'a pas peint les bonnes cases.

Timéo – **Mm**

ferme le feedback

pointe le robot en position finale

Timéo – **Comment faire ? mm.. mm.. mm..**

pointe le robot,

Timéo – **Alors.. il reste plus de blocs**

parcourt le corps de la boucle de haut en bas

Timéo – **Et dedans j' pense qu'on pourrait.. répéter quelque chose** (survole le corps de la boucle, supprime les deux derniers blocs, ajoute un bloc REP autour des blocs EST et PC, comp 2) **et je fais ça**

→ stratégie de passage à la boucle imbriquée, de synthèse de l'écriture de ce qui est déjà codé (à partir du motif algorithmique) vs recherche du motif optimal (à partir du motif visuel)

exécution du programme d'id.2951 → le dernier point n'est pas peint, feedback : Votre robot n'a pas peint les bonnes cases.

Timéo – **Et ça marche..**

ferme le feedback, met le curseur à plus rapide, relance une exécution du même programme, puis une nouvelle fois qu'il interrompt avant la fin, utilise le mode pas à pas pour peindre le dernier motif

survole la grille, survole le programme,

Timéo – **Mm.. ah oui il me reste un bloc (ajEST) mm.. heu..**

exécution du programme d'id.2949 → le dernier point n'est pas peint, feedback : Votre robot n'a pas peint les bonnes cases.

ferme le feedback, remet en situation initiale, relance l'exécution et l'interrompt lorsque le robot est sur le dernier groupe de points, continue ensuite en mode pas à pas jusqu'au feedback

Timéo – **On va essayer quatre fois** (comp 4)

exécution du programme d'id.8917 → le robot a peint le premier point et les quatre premiers groupes, feedback : Votre robot n'a pas peint les bonnes cases.

ferme le feedback

Timéo – **C'est bien cinq.. c'est bien cinq** fois (comp 5)

supprime le bloc EST après la boucle, survole la grille

Timéo – **Comment faire ?.. je vais essayer de mettre ça..** (décroche le programme, enlève les instructions avant la boucle) **alors..** (met un bloc EST en début de boucle, décroche le corps de la boucle externe, supprime le bloc EST qu'il vient de mettre)

survole la grille

Timéo – **ça c'est de mettre est.. peindre la case** (intercale un bloc PC en deuxième position dans la boucle) **ça..** (pointe le 1<sup>er</sup> point) **peindre la case.. puis..** (pointe le groupe de deux points suivant)

pointe le 1<sup>er</sup> point, le 2<sup>ème</sup> point, le 3<sup>ème</sup> point

→ Timéo a inclus le traitement du premier point dans la boucle, mais sans ajuster correctement le reste du motif algorithmique.

Timéo – **Ah je pense que..** (ajEST, ajPC)

exécution du programme d'id.8918 → le robot peint toutes les cases, feedback : Votre robot n'a pas peint les bonnes cases.

Timéo – **Ah non.. ah oui non ok**

ferme le feedback, remet en situation initiale

intercale un bloc EST juste avant la boucle → feedback un bloc en trop

Timéo – **Et comme ça.. non un autre truc..**

prend les deux instructions après la boucle, les met dans la boucle, les remet à leur place

pointe le 2<sup>ème</sup> point

survole le programme

Timéo – **Euh..** (pr-rel une partie du programme) **j'vois pas trop..** (mouvements désordonnés de la souris)

décroche le dernier bloc PC

Timéo – **J'vais essayer ça.. est-ce que ça marche ?**

exécution du programme d'id.8918 → le robot peint la première case puis des groupes de trois cases, feedback : Votre robot n'a pas peint les bonnes cases.

Timéo – **Ok.. donc ça va pas**

ferme le feedback, remet en situation initiale, exécute à nouveau en mode pas à pas

Timéo – **Avancer vers l'est** (pp) **peindre la case** (pp) **avancer vers l'est** (pp) **avancer vers l'est peindre la case**

E – **Peut-être passer au suivant et d'y revenir à la fin si..**

Timéo – **D'accord**

E – **il te reste du temps**

quitte le puzzle

Deuxième passage

Timéo – **Ok..**

vide l'éditeur

Timéo – **Avancer** (ajEST, ajPC, ajEST) **avancer vers l'est.. donc ça c'est bon.. faire**

pointe la première case vide, la 2<sup>ème</sup>

Timéo – **Un deux** (pointe le 2<sup>ème</sup> point, le 3<sup>ème</sup>)

ajREP

comp 2

Timéo – **Avancer vers l'est** (repEST) **puis peindre la case** (repPC) **puis avancer vers l'est** (ajEST)

décroche la boucle et l'instruction en aval et les entoure avec un bloc REP

pointe le 1<sup>er</sup> point, puis les groupes de deux points en les dénombrant (chuchoté)

comp 5

accroche à la suite du programme

Timéo – **C'est comme tout à l'heure**

Exécution du programme d'id.8916 → le robot s'arrête sur la case juste avant le dernier point, feedback :  
Votre robot n'a pas peint les bonnes cases.

Timéo – **Ok..** (relance l'exécution et l'interrompt lorsque le robot est sur la dernière case vide)

→ On peut se demander si le schème de débogage par complétion ne constitue pas un obstacle à la modification du motif, car il semble difficile pour le sujet de remettre en question le début du programme déjà édité.

parcourt les points de la grille

supprime la boucle, vide l'éditeur

ajEST, ajPC, aj REP

Timéo – **Quoi faire ?**

→ expression du blocage : verbal

comp 5

repEST, repPC, repREP

décroche la boucle externe, décroche la séquence qui est dedans, remet cette boucle, comp 2 pour la boucle interne, rep2EST, rep2PC

E – **Alors si tu n'arrives pas.. tu peux aussi aller voir les versions quatre de ceux que tu as débloqués**

supprime la boucle

E – **Surtout ceux que tu as trouvés facile**

Timéo – **D'accord**

→ Cette confrontation est caractéristique d'un blocage qui engendre une stratégie en essais-erreurs, la seule qui reste disponible dans ce cas. Le sujet alterne entre une stratégie par essais-erreurs, et une répétition de sa stratégie de départ. Il tourne en boucle sans que sa stratégie évolue vraiment. Il a épuisé tous les schèmes à sa disposition. Il essaie des variantes autour de sa stratégie initiale, en étant parfois convaincu que ça ne marche pas, même avant d'avoir exécuté le programme.

## Timéo, 2022 expérimentation complémentaire



### Timéo, 2022t4p4v1 – profil ajustement

[lien vers l'extrait vidéo](#)

ps84axji 5em

score global : 180

temps moyen par bloc : 6.6

4 - Pousser les caisses [uLille v2] basic

durée : 1'11

ind. rapidité normalisée : 0.34

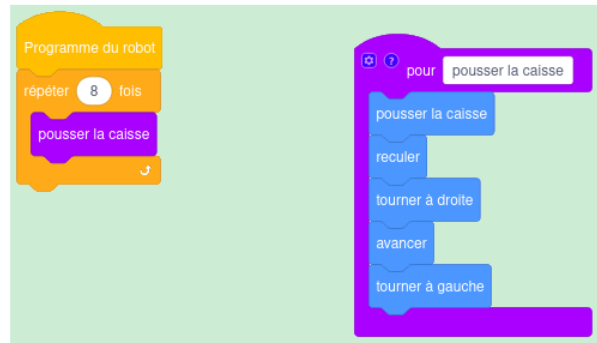
0 - 29'20

1 - 29'51  
0'31

2 - 30'31  
0'40



id sol : 18439



18440

verbalisation → intéressant à transcrire

test du bloc pousser la caisse

### utilisation du bloc de fonction

Timéo – *Ok mm..* (ouvre les différentes catégories du menu)

amène un bloc de fonction dans l'éditeur, édite le nom

Timéo – *Pousser la caisse*

ouvre la catégorie actions

Timéo – *Pousser la caisse* (foncPC) *ah faut que je teste un truc d'abord* (ajPC)

exécution du programme d'id.18439

Timéo – *Ok.. reculer* (foncREC) *tourner à droite* (ajTD) *avancer* (ajAV) *tourner à gauche* (ajTG)

pr-rel PC, ouvre la catégorie fonctions

Timéo – *Et bah on va..* (aj le bloc d'appel à la fonction pousser la caisse, ouvre la catégorie contrôle) *faire* (entoure le bloc avec REP)

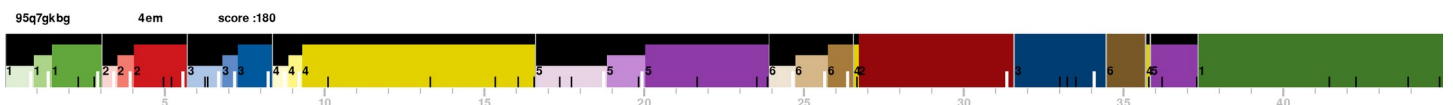
pointe les caisses en dénombrant tout haut jusque huit

Timéo – *Huit fois pousser la caisse* (comp 8)

exécution du programme d'id.18440 → validation

## Mickaël, 4<sup>ème</sup> en 2022

### Mickaël, 2022 tour 3



### Mickaël, 2022t3p5v1 - profil accommodation

[lien vers l'extrait vidéo](#)



95q7gkbg 4em

score global : 180

temps moyen par bloc : 7.4

5 - Dans l'espace [2022-3 white]

basic

durée : 2'10

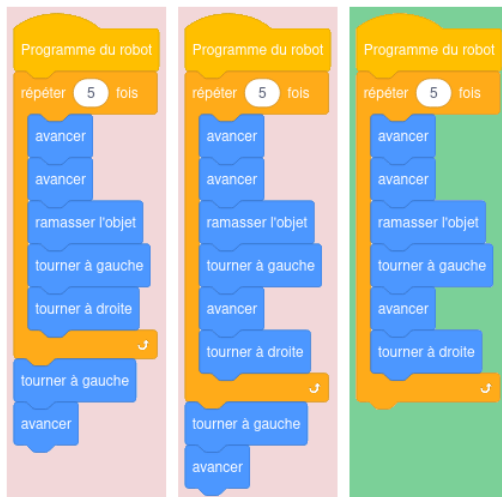
ind. rapidité normalisée : 0.95

0 - 16'37

1 - 17'23  
0'46

2 - 17'45  
0'22

3 - 18'47  
1'02



id sol :

6972

6973

86

Noté pendant la session : manque une instruction de déplacement, erreur en sortie en boucle ; rectifié grâce à une exécution en mode pas à pas

aj REP

rep AV AV RAM TG TD

pointe le bouton d'exécution

aj TG AV

**Les instructions que Mickaël place après la boucle révèle qu'il se représente le robot sur la case du dernier objet et orienté vers la droite après la boucle.**

comp 5

Exécution du programme d'id.6972

parcourt les trois premiers blocs de son programme de haut en bas → simule mentalement l'exécution

remet la grille en situation initiale

lance un début d'exécution en mode pas à pas, jusqu'à ce que le bloc TD soit surligné.

intercale le bloc AV manquant

Exécution du programme d'id.6973

ouvre le clavier virtuel

simule le parcours du robot avec le pointeur, une fois entièrement, puis une fois partiellement

ferme le feedback, remet la grille en situation initiale

début une exécution avec le bouton play, l'interrompt lorsque le robot est sur le dernier objet, et continue en mode pas à pas

supprime les deux blocs après la boucle

validation avec le programme d'id.86

Dans cette confrontation, le mode pas à pas permet de corriger les deux erreurs, l'une après l'autre. Utilisation experte du mode pas à pas qui vient supporter la représentation mentale de l'exécution qui n'est pas efficace ici.

## Mickaël, 2022 expérimentation complémentaire



### Mickaël, 2022t4p1v4 - profil ajustement

[lien vers l'extrait vidéo](#)

8f74cnut 4em score global : 190 temps moyen par bloc : 6.7  
1 - Planter des fleurs [uLille v2] hard durée : 1'15 ind. rapidité normalisée : 0.24

0 - 3'55	1 - 4'49 0'54	2 - 5'03 0'14	3 - 5'10 0'07
id sol :	6346	6347	6348

seul verbatim en tout début d'extrait, en référence à la résolution du puzzle précédent qui a été résolu avec une séquence : « Là je ne vais pas faire ça ce serait bête »

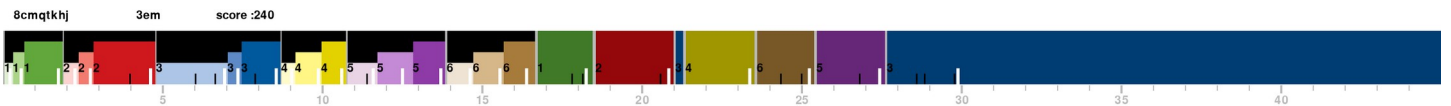
distingue les deux types de situation mais n'explique pas verbalement

**pas de passage sur la grille** → visualisation des mouvements de souris



**Noé, 4<sup>ème</sup> en 2021, 3<sup>ème</sup> en 2022**

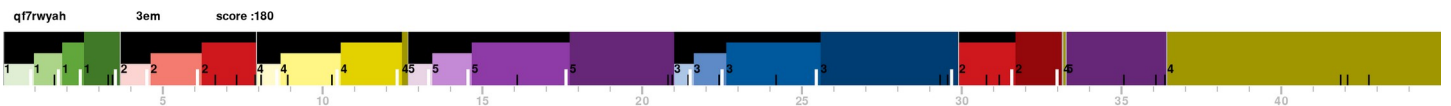
### Noé, 2022 tour 3



### Noé, 2022t3p4v2 – profil expert

[lien vers l'extrait vidéo](#)

### Noé, 2022 expérimentation complémentaire



### Noé, 2022t4p1v2 – profil ajustement

[lien vers l'extrait vidéo](#)

qf7rwyah 3em score global : 180 temps moyen par bloc : 6.4  
 1 - Planter des fleurs [uLille v2] easy durée : 0'49 ind. rapidité normalisée : -0.04

0 - 0'58	1 - 1'38 0'40	2 - 1'47 0'09
id sol :	7626	18812

## Jade, 2021 tour 1



### Jade, 2021t1p5v2 – profil ajustement

[lien vers l'extrait vidéo](#)

pas de verbalisation

conçoit le motif algorithmique et le teste d'abord tout seul avant de l'inclure dans un bloc répéter.

Confrontation assez canonique et intéressante de ce point de vue

Du point de vue du profil, ce n'est pas une erreur donc pas un profil ajustement. Dans quelle mesure est-ce un programme partiel ? au sens d'un premier motif ?

## **Annexe 10 : traitements relatifs aux premiers apprentissages de la boucle**

# traitements\_premiers\_apprentissages\_boucle

July 29, 2024

## 1 Traitements relatifs aux premiers apprentissages de la boucle (échelles des classes)

```
[1]: import sys
      sys.path.append('../')

      from src import *
      import pandas as pd
      import re
      import seaborn as sns

      %load_ext autoreload
      %autoreload 2
```

pygame 2.1.2 (SDL 2.0.16, Python 3.9.15)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

2024-07-29 14:54:17.329914: I tensorflow/core/platform/cpu\_feature\_guard.cc:193]

This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2024-07-29 14:54:17.432609: E tensorflow/stream\_executor/cuda/cuda\_blas.cc:2981]

Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

2024-07-29 14:54:17.910946: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dLError: libnvinfer.so.7: cannot open shared object file: No such file or directory

2024-07-29 14:54:17.911003: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer\_plugin.so.7'; dLError: libnvinfer\_plugin.so.7: cannot open shared object file: No such file or directory

2024-07-29 14:54:17.911009: W

tensorflow/compiler/tf2tensorrt/utils/py\_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed

properly.

```
[2]: pd.options.mode.chained_assignment = None
```

```
[3]: pd.set_option('display.max_rows',200) # augmente la taille maximale d'un
      ↪ dataframe à afficher entièrement
      # si le dataframe dépasse cette taille, il sera condensé, sinon, il sera
      ↪ affiché entièrement
```

### 1.0.1 Initialisation

```
[4]: file_tx_reussite_groupe = "../data/data_these/taux_reussite_groupe.csv"
      file_sol_prot = '../data/data_these/table_sol_canoniques.csv'
      file_ref = '../data/data_these/sol_ref.csv'
      file_clean = '../data/data_these/clean_nov23.csv'
      file_profils = '../data/data_these/profils_nov23.csv'
      file_users = '../data/data_these/users.csv'
```

```
[5]: dt_users = pd.read_csv(file_users)
      dt_users.head(5)
```

```
[5]:
```

	id_common	password	school_year	session	category	video
0	1	8hm6qrhv	ce1	2021-1	blanc	True
1	1	rfhu4i84	ce1	2021-2	blanc	True
2	1	m4phaen4	ce1	2021-3	blanc	True
3	2	f3avqege	ce1	2021-1	blanc	True
4	2	k5gx9bfn	ce1	2021-2	blanc	True

```
[6]: dt_clean = pd.read_csv(file_clean)
      dt_clean = dt_clean.merge(dt_users[['id_common', 'password']], on='password',
      ↪ how='left')
      dt_clean.head(5)
```

```
[6]:
```

	ID	password	id_group	id_problem	\
0	741412	33wxj8ri	603931811100778560	808561348827292307	
1	741426	33wxj8ri	603931811100778560	808561348827292307	
2	741481	33wxj8ri	603931811100778560	808561348827292307	
3	741496	33wxj8ri	603931811100778560	808561348827292307	
4	741578	33wxj8ri	603931811100778560	808561348827292307	

		problem	level	type	\
0	1	- Dans l'espace [2022-3 white]	menu	load	
1	1	- Dans l'espace [2022-3 white]	basic	load	
2	1	- Dans l'espace [2022-3 white]	basic	submission	
3	1	- Dans l'espace [2022-3 white]	easy	load	
4	1	- Dans l'espace [2022-3 white]	easy	submission	

	answer	score	\
0	NaN	0	
1	NaN	0	
2	<xml><block type="robot_start" id="space" x="0...	10	
3	NaN	0	
4	<xml><block type="robot_start" id="space" x="0...	20	

	date	time_session	time_attempt	time_load	context	grade	\
0	2022-05-12 08:48:37		0	0	3	space	5em
1	2022-05-12 08:48:40		3	3	16	space	5em
2	2022-05-12 08:48:56		19	16	4	space	5em
3	2022-05-12 08:49:00		23	4	23	space	5em
4	2022-05-12 08:49:23		46	23	5	space	5em

	id_sol	id_common
0	0	152
1	0	152
2	1	152
3	0	152
4	2	152

```
[7]: dt_profils = pd.read_csv(file_profils)
dt_profils = dt_profils.merge(dt_users[['id_common', 'password']],
                              on='password', how='left')
dt_profils.head(5)
```

```
[7]: password          id_problem          problem \
0 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
1 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
2 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
3 33wxj8ri  36614660447619275  3 - Pousser les caisses [2022-3 white]
4 33wxj8ri  613408804708837047  2 - Atteindre le drapeau [2022-3 white]
```

	level	profile	success	attempts_number	time_spent_on_question	\
0	basic	accommodation	True	6	77	
1	easy	expert	True	1	21	
2	medium	accommodation	True	17	319	
3	hard	échec	False	1	56	
4	basic	expert	True	1	36	

	mean_time_per_block	block_count_ref	ind_speed	grade	\
0	5.1	4.0	2.774510	5em	
1	5.1	7.0	-0.411765	5em	
2	5.1	13.0	3.811463	5em	
3	5.1	11.0	-0.001783	5em	
4	5.1	6.0	0.176471	5em	



	tx_reussite_concours	id_common
0	0.938	152
1	0.913	152
2	0.799	152
3	0.054	152
4	0.994	152

```
[8]: classes_college = ['6em', '5em', '4em', '3em']
```

```
[9]: problem_white_2022 = list(dt_clean[(dt_clean['problem'].str.contains('2022')) &
                                         (dt_clean['problem'].str.contains('white'))
                                         ]['problem'].unique())

problem_white_2022
```

```
[9]: ["1 - Dans l'espace [2022-3 white]",
      '2 - Atteindre le drapeau [2022-3 white]',
      '3 - Pousser les caisses [2022-3 white]',
      '4 - Peindre le dessin [2022-3 white]',
      "5 - Dans l'espace [2022-3 white]",
      '6 - Peindre le dessin [2022-3 white]',
      '1 - Chauffer le château [2022-1 white]',
      '2 - Collecter les pierres [2022-1 white]',
      '3 - Planter des fleurs [2022-1 white]',
      '4 - Planter des fleurs [2022-1 white]',
      '5 - Collecter les pierres [2022-1 white]',
      '6 - Chauffer le château [2022-1 white]',
      '1 - Ranger les billes [2022-2 white]',
      '2 - Distribuer le bois [2022-2 white]',
      '3 - Ranger les billes [2022-2 white]',
      '4 - Chauffer le château [2022-2 white]',
      '5 - Distribuer le bois [2022-2 white]',
      '6 - Chauffer le château [2022-2 white]']
```

```
[10]: problems_white_2022t1 = list(dt_clean[dt_clean['problem'].str.contains('2022-1_
→white')]['problem'].unique())

problems_white_2022t1
```

```
[10]: ['1 - Chauffer le château [2022-1 white]',
      '2 - Collecter les pierres [2022-1 white]',
      '3 - Planter des fleurs [2022-1 white]',
      '4 - Planter des fleurs [2022-1 white]',
      '5 - Collecter les pierres [2022-1 white]',
      '6 - Chauffer le château [2022-1 white]']
```

```
[11]: problems_white_2022t2 = list(dt_clean[dt_clean['problem'].str.contains('2022-2_
→white')]['problem'].unique())

problems_white_2022t2
```

```
[11]: ['1 - Ranger les billes [2022-2 white]',
      '2 - Distribuer le bois [2022-2 white]',
      '3 - Ranger les billes [2022-2 white]',
      '4 - Chauffer le château [2022-2 white]',
      '5 - Distribuer le bois [2022-2 white]',
      '6 - Chauffer le château [2022-2 white]']
```

```
[12]: problems_white_2022t3 = list(dt_clean[dt_clean['problem'].str.contains('2022-3_
→white')]['problem'].unique())
problems_white_2022t3
```

```
[12]: ["1 - Dans l'espace [2022-3 white]",
      '2 - Atteindre le drapeau [2022-3 white]',
      '3 - Pousser les caisses [2022-3 white]',
      '4 - Peindre le dessin [2022-3 white]',
      "5 - Dans l'espace [2022-3 white]",
      '6 - Peindre le dessin [2022-3 white]']
```

```
[13]: problems_white_2021 = list(dt_clean[(dt_clean['problem'].str.contains('2021'))&
      (dt_clean['problem'].str.contains('white'))
      ]['problem'].unique())
problems_white_2021.remove('8 - Distribuer le bois [white 2021]')
problems_white_2021
```

```
[13]: ['1 - Atteindre le drapeau [white 2021]',
      '2 - Planter des fleurs [white 2021]',
      '3 - Rejoindre la fusée [white 2021]',
      '4 - Rejoindre la fusée [white 2021]',
      '5 - Atteindre le drapeau [white 2021]',
      '6 - Planter des fleurs [white 2021]',
      '1 - Planter des fleurs [white 2021]',
      '2 - Atteindre un drapeau [white 2021]',
      '3 - Ranger les billes [white 2021]',
      '4 - Planter des fleurs [white 2021]',
      '5 - Collecter les pierres précieuses [white 2021]',
      '6 - Atteindre un drapeau [white 2021 new path]',
      '1 - Peindre le motif [white 2021]',
      '2 - Rejoindre la fusée [white 2021]',
      '3 - Peindre le motif [white 2021]',
      '4 - Jouer aux dominos [white 2021]',
      '5 - Planter des fleurs [white 2021]',
      '6 - Rejoindre la fusée [white 2021]']
```

```
[14]: problems_white_2021t1 = [
      '1 - Planter des fleurs [white 2021]',
      '2 - Atteindre un drapeau [white 2021]',
      '3 - Ranger les billes [white 2021]',
```

```
'4 - Planter des fleurs [white 2021]',  
'5 - Collecter les pierres précieuses [white 2021]',  
'6 - Atteindre un drapeau [white 2021 new path]'
```

## 1.0.2 Effectifs

Pour l'ensemble des classe de collège en 2022

```
[15]: dt_clean_2022_white_college = dt_clean[(dt_clean['problem'].  
      ↳isin(problem_white_2022)) &  
      (dt_clean['grade'].  
      ↳isin(classes_college))]
```

```
[16]: dt_clean_2022_white_college['password'].nunique()
```

```
[16]: 495
```

Tour 1 de 2022

```
[17]: dt_clean_2022t1_white_college = dt_clean[(dt_clean['problem'].  
      ↳isin(problems_white_2022t1)) &  
      (dt_clean['grade'].  
      ↳isin(classes_college))]
```

```
[18]: dt_clean_2022t1_white_college['password'].nunique()
```

```
[18]: 193
```

```
[19]: liste_par_classe_t1 = []  
      for classe in classes_college :  
          liste_par_classe_t1.  
      ↳append(dt_clean_2022t1_white_college[dt_clean_2022t1_white_college['grade']==classe]['password'].  
      ↳nunique())  
      liste_par_classe_t1
```

```
[19]: [48, 71, 24, 50]
```

Tour 2 de 2022

```
[20]: dt_clean_2022t2_white_college = dt_clean[(dt_clean['problem'].  
      ↳isin(problems_white_2022t2)) &  
      (dt_clean['grade'].  
      ↳isin(classes_college))]
```

```
[21]: dt_clean_2022t2_white_college['password'].nunique()
```

```
[21]: 179
```

```
[22]: liste_par_classe_t2 =[]
for classe in classes_college :
    liste_par_classe_t2.
    ↳append(dt_clean_2022t2_white_college[dt_clean_2022t2_white_college['grade']==classe]['passw
    ↳nunique())
liste_par_classe_t2
```

[22]: [43, 60, 19, 57]

Tour 3 de 2022

```
[23]: dt_clean_2022t3_white_college = dt_clean[(dt_clean['problem'].
    ↳isin(problems_white_2022t3)) &
                                         (dt_clean['grade'].
    ↳isin(classes_college))]
```

```
[24]: dt_clean_2022t3_white_college['password'].nunique()
```

[24]: 123

```
[25]: liste_par_classe_t3 =[]
for classe in classes_college :
    liste_par_classe_t3.
    ↳append(dt_clean_2022t3_white_college[dt_clean_2022t3_white_college['grade']==classe]['passw
    ↳nunique())
liste_par_classe_t3
```

[25]: [21, 66, 20, 16]

### 1.0.3 Évaluation de la maîtrise de la boucle en début de session pour les sujets scalarisés au collège

#### 1.0.4 Édition de 2022

```
[26]: def use_repeat (answer_xml) :
    return 'control_repeat' in answer_xml
```

```
[27]: list_problems_2022p1 = ['1 - Chauffer le château [2022-1 white]',
    '1 - Ranger les billes [2022-2 white]',
    '1 - Dans l'espace [2022-3 white]"]
level = "medium"
```

```
[28]: dt_2022p1v3 = dt_clean[(dt_clean['problem'].isin(list_problems_2022p1)) &
    (dt_clean['level'] == level) &
    (dt_clean['grade'].isin(classes_college)) &
    ~(dt_clean['answer'].isna())
    ]
```

```
[29]: dt_2022p1v3['use_repeat'] = dt_2022p1v3['answer'].apply(use_repeat)
dt_2022p1v3.head(5)
```

```
[29]:
```

	ID	password	id_group	id_problem	\
6	741752	33wxj8ri	603931811100778560	808561348827292307	
189	405799	33yjhik9	18142768599727689	916979020758482351	
190	405820	33yjhik9	18142768599727689	916979020758482351	
365	504567	34r3vvnw	892745511171656231	981839330688093742	
451	429481	363spjfx	433088959115828334	981839330688093742	

	problem	level	type	\
6	1 - Dans l'espace [2022-3 white]	medium	submission	
189	1 - Chauffer le château [2022-1 white]	medium	attempt	
190	1 - Chauffer le château [2022-1 white]	medium	submission	
365	1 - Ranger les billes [2022-2 white]	medium	submission	
451	1 - Ranger les billes [2022-2 white]	medium	attempt	

	answer	score	\
6	<xml><block type="robot_start" id="space" x="0...	30	
189	<xml><block type="robot_start" id="castle" x="...	0	
190	<xml><block type="robot_start" id="castle" x="...	30	
365	<xml><block type="robot_start" id="marbles" x=...	30	
451	<xml><block type="robot_start" id="marbles" x=...	0	

	date	time_session	time_attempt	time_load	context	\
6	2022-05-12 08:50:17	100	49	6	space	
189	2022-01-26 09:19:47	132	39	11	castle	
190	2022-01-26 09:19:58	143	11	7	castle	
365	2022-03-17 08:45:07	211	137	6	marbles	
451	2022-02-20 09:01:46	116	55	16	marbles	

	grade	id_sol	id_common	use_repeat
6	5em	3	152	True
189	6em	93	31	True
190	6em	94	31	True
365	4em	217	203	True
451	6em	270	8	True

```
[30]: dt_valid_sequence = dt_2022p1v3[(dt_2022p1v3['type'] == 'submission') &
↳ (dt_2022p1v3['use_repeat'] == False)]
dt_valid_sequence.drop_duplicates(subset=['password'], inplace=True)
len(dt_valid_sequence)
```

```
[30]: 24
```

```
[31]: list_password_sequence = list(dt_valid_sequence['password'].unique())
```

```
[32]: pd.crosstab(dt_valid_sequence['grade'], dt_valid_sequence['problem'] )
```

```
[32]: problem 1 - Chauffer le château [2022-1 white] \
grade
3em          3
4em          1
5em          9
6em          3
```

```
problem 1 - Ranger les billes [2022-2 white]
grade
3em          1
4em          1
5em          4
6em          2
```

```
[33]: dt_valid_repeat = dt_2022p1v3[(dt_2022p1v3['type'] == 'submission') &
↳ (dt_2022p1v3['use_repeat'] == True)]
dt_valid_repeat.drop_duplicates(subset=['password'], inplace=True)
dt_valid_repeat['password'].nunique()
```

```
[33]: 459
```

```
[34]: list_password_repeat = list(dt_valid_repeat['password'].unique())
```

```
[35]: pd.crosstab(dt_valid_repeat['grade'], dt_valid_repeat['problem'] )
```

```
[35]: problem 1 - Chauffer le château [2022-1 white] \
grade
3em          43
4em          23
5em          61
6em          44
```

```
problem 1 - Dans l'espace [2022-3 white] \
grade
3em          16
4em          20
5em          66
6em          21
```

```
problem 1 - Ranger les billes [2022-2 white]
grade
3em          53
4em          18
5em          53
6em          41
```

```
[36]: list_password_valid = list_password_sequence + list_password_repeat
```

```
[37]: dt_echec = dt_clean[(dt_clean['problem'].isin(list_problems_2022p1)) &
                        (dt_clean['level'] == level) &
                        (dt_clean['grade'].isin(classes_college)) &
                        ~(dt_clean['password'].isin(list_password_valid))
                        ]
dt_echec.drop_duplicates(subset=['password'], inplace=True)
len(dt_echec)
```

```
[37]: 7
```

```
[38]: list_password_echec = list(dt_echec['password'].unique())
```

```
[39]: pd.crosstab(dt_echec['grade'], dt_echec['problem'] )
```

```
[39]: problem 1 - Chauffer le château [2022-1 white] \
grade
3em          1
5em          0

problem 1 - Ranger les billes [2022-2 white]
grade
3em          3
5em          3
```

### 1.0.5 Étude des sujets qui ont validé avec une séquence ou qui sont en échec (collège, 2022)

```
[40]: dt_profils_repeat = dt_profils[(dt_profils['problem'].
    ↳isin(list_problems_2022p1)) &
    (dt_profils['level'] == 'medium') &
    (dt_profils['password'].
    ↳isin(list_password_repeat))
    ]
len(dt_profils_repeat)
```

```
[40]: 459
```

```
[41]: pd.crosstab(dt_profils_repeat['profile'], dt_profils_repeat['problem'])
```

```
[41]: problem          1 - Chauffer le château [2022-1 white] \
profile
accommodation          48
ajustement             33
expert                90
indéterminé           0
```

```

problem      1 - Dans l'espace [2022-3 white]  \
profile
accommodation      41
ajustement         43
expert             38
indéterminé       1

```

```

problem      1 - Ranger les billes [2022-2 white]
profile
accommodation      88
ajustement         33
expert             42
indéterminé       2

```

```
[42]: list_passwords_non_repeat = list_password_sequence + list_password_echec
```

```
[43]: list_problems_2022p3 = ['3 - Planter des fleurs [2022-1 white]',
                             '3 - Ranger les billes [2022-2 white]',
                             '3 - Pousser les caisses [2022-3 white]'
                             ]
```

```
[44]: list_id_common_non_repeat = list(dt_users[dt_users['password'].
    ↳isin(list_passwords_non_repeat)]['id_common']).unique()
len(list_id_common_non_repeat)
```

```
[44]: 28
```

```
[45]: dt_profils_non_repeat = dt_profils[(((dt_profils['problem'].
    ↳isin(list_problems_2022p1)) & (dt_profils['level'] == 'medium')) |
    ((dt_profils['problem'].
    ↳isin(list_problems_2022p3)) & (dt_profils['level'].isin(['easy'])))) &
    (dt_profils['id_common'].
    ↳isin(list_id_common_non_repeat))]
dt_profils_non_repeat = dt_profils_non_repeat.
    ↳sort_values(by='id_common')[['id_common', 'password', 'problem', 'level',
    ↳'profile']]
```

```
[46]: table_profils_non_repeat = pd.pivot_table(dt_profils_non_repeat,
    ↳index=['id_common'], columns=['problem', 'level'], values=['profile'],
    ↳aggfunc=lambda x: ' '.join(x))
dt_profils_passage_seq_boucle = pd.DataFrame(table_profils_non_repeat)
dt_profils_passage_seq_boucle.columns = ['2022-1 1 - Chauffer le château v3',
    "2022-3 1 - Dans l'espace v3",
    '2022-2 1 - Ranger les billes v3',
    '2022-1 3 - Planter des fleurs v2',
    '2022-3 3 - Pousser les caisses v2',
```



```

        '2022-2 3 - Ranger les billes v2', ]
dt_profils_passage_seq_boucle = dt_profils_passage_seq_boucle[['2022-1 1 ->
↳Chauffer le château v3',
        '2022-1 3 - Planter des fleurs v2',
        '2022-2 1 - Ranger les billes v3',
        '2022-2 3 - Ranger les billes v2',
        "2022-3 1 - Dans l'espace v3",
        '2022-3 3 - Pousser les caisses v2'
]]
dt_profils_passage_seq_boucle = dt_profils_passage_seq_boucle.
↳merge(dt_users[['id_common','school_year']], on='id_common', how='left').
↳drop_duplicates().sort_values(by='id_common')
dt_profils_passage_seq_boucle

```

```

[46]:      id_common 2022-1 1 - Chauffer le château v3 \
0          18                expert
4          30                NaN
6          40                accommodation
9          41                accommodation
12         45                expert
17         48                expert
20         52                expert
22         64                accommodation
25         89                NaN
27        100                expert
30        101                accommodation
32        104                ajustement
35        105                accommodation
38        108                expert
41        116                ajustement
43        126                accommodation
46        139                expert
48        145                expert
51        186                NaN
53        188                NaN
56        192                ajustement
58        206                ajustement
60        214                NaN
63        222                NaN
65        225                ajustement
68        239                NaN
70        248                accommodation
72        255                échec

```

```

      2022-1 3 - Planter des fleurs v2 2022-2 1 - Ranger les billes v3 \
0                expert                expert
4                expert                accommodation

```

6	ajustement	accommodation
9	accommodation	ajustement
12	expert	ajustement
17	expert	accommodation
20	ajustement	accommodation
22	expert	accommodation
25	accommodation	échec
27	ajustement	accommodation
30	accommodation	échec
32	expert	accommodation
35	ajustement	accommodation
38	ajustement	échec
41	ajustement	accommodation
43	ajustement	accommodation
46	expert	NaN
48	expert	NaN
51	NaN	ajustement
53	NaN	échec
56	expert	NaN
58	expert	NaN
60	NaN	ajustement
63	NaN	échec
65	ajustement	NaN
68	NaN	accommodation
70	expert	échec
72	NaN	NaN

2022-2 3 - Ranger les billes v2 2022-3 1 - Dans l'espace v3 \

0	expert	ajustement
4	ajustement	NaN
6	expert	accommodation
9	expert	ajustement
12	accommodation	expert
17	expert	accommodation
20	expert	NaN
22	ajustement	accommodation
25	accommodation	NaN
27	expert	accommodation
30	ajustement	NaN
32	expert	expert
35	accommodation	accommodation
38	ajustement	accommodation
41	ajustement	NaN
43	expert	expert
46	NaN	accommodation
48	NaN	NaN
51	ajustement	ajustement

53	accommodation	accommodation
56	NaN	NaN
58	NaN	NaN
60	indéterminé	ajustement
63	accommodation	accommodation
65	NaN	NaN
68	ajustement	accommodation
70	ajustement	NaN
72	NaN	NaN

	2022-3 3 - Pousser les caisses v2	school_year
0	ajustement	6em
4	NaN	3em
6	accommodation	6em
9	ajustement	4em
12	expert	5em
17	NaN	5em
20	NaN	3em
22	ajustement	5em
25	NaN	3em
27	ajustement	4em
30	NaN	5em
32	expert	5em
35	expert	5em
38	accommodation	5em
41	NaN	3em
43	expert	5em
46	expert	5em
48	NaN	6em
51	accommodation	6em
53	ajustement	3em
56	NaN	5em
58	NaN	3em
60	ajustement	5em
63	accommodation	5em
65	NaN	5em
68	ajustement	5em
70	NaN	3em
72	NaN	3em

```
[ ]: dt_profils_passage_seq_boucle.to_csv('../data/data_these/
↳ passage_sequence_boucle.csv', index=False)
```

```
[ ]: list_id_common_tour_1 = list(dt_valid_sequence[dt_valid_sequence['problem']=='1_
↳ Chauffer le château [2022-1 white]']['id_common'].unique())
list_id_common_tour_1.sort()
list_id_common_tour_1
```

```
[ ]: list_id_common_tour_2 = list(dt_valid_sequence[dt_valid_sequence['problem']=='1']
    ↳ Ranger les billes [2022-2 white]')['id_common'].unique()
list_id_common_tour_2.sort()
list_id_common_tour_2
```

### 1.0.6 Analyse de sessions d'élèves plus jeunes pour le tour 2021-1

```
[47]: file_clean_2021 = '../data/algorea_2021/clean_global_2021_1.csv'
file_users_2021 = '../data/algorea_2021/users_2021_1.csv'
file_profils_2021 = '../data/algorea_2021/profils_global_2021_1.csv'
```

```
[51]: dt_users_2021 = pd.read_csv(file_users_2021)
dt_clean_2021 = pd.read_csv(file_clean_2021)
```

```
[52]: len(dt_clean_2021)
```

```
[52]: 9438
```

```
[53]: dt_clean_2021 = dt_clean_2021.merge(dt_users_2021[['id_common', 'password']],
    ↳ on='password', how='left')
```

```
[54]: problem_2021t1p1 = '1 - Planter des fleurs [white 2021]'
level = "medium"
```

```
[55]: classes_etude_2021 = ['ce', 'cm', '6em']
```

```
[56]: list_participants_2021t1 = []
for classe in classes_etude_2021 :
    list_participants_2021t1.
    ↳ append(dt_clean_2021[dt_clean_2021['grade']==classe]['password'].nunique())
list_participants_2021t1
```

```
[56]: [69, 49, 9]
```

```
[59]: dt_2021t1p1v3 = dt_clean_2021[(dt_clean_2021['problem']==problem_2021t1p1) &
    (dt_clean_2021['level'] == level) &
    ~(dt_clean_2021['answer'].isna())
    ]
dt_2021t1p1v3.head(5)
```

```
[59]:
```

	ID	password	id_group	id_problem	\
7	281255	3dgdw5xv	825200314627543784	603617216933141191	
8	281320	3dgdw5xv	825200314627543784	603617216933141191	
11	281362	3dgdw5xv	825200314627543784	603617216933141191	
18	281447	3dgdw5xv	825200314627543784	603617216933141191	
19	281474	3dgdw5xv	825200314627543784	603617216933141191	

		problem	level	type	\
7	1	Planter des fleurs [white 2021]	medium	attempt	
8	1	Planter des fleurs [white 2021]	medium	attempt	
11	1	Planter des fleurs [white 2021]	medium	attempt	
18	1	Planter des fleurs [white 2021]	medium	attempt	
19	1	Planter des fleurs [white 2021]	medium	submission	

		answer	score	\
7	<xml><block type="robot_start" id="flowers" x=...		0	
8	<xml><block type="robot_start" id="flowers" x=...		0	
11	<xml><block type="robot_start" id="flowers" x=...		0	
18	<xml><block type="robot_start" id="flowers" x=...		0	
19	<xml><block type="robot_start" id="flowers" x=...		30	

	date	time_session	time_attempt	time_load	grade	context	\
7	2021-06-04 13:44:51	343	71	127	ce	flowers	
8	2021-06-04 13:46:58	470	127	14	ce	flowers	
11	2021-06-04 13:48:38	570	77	12	ce	flowers	
18	2021-06-04 13:51:48	760	88	63	ce	flowers	
19	2021-06-04 13:52:51	823	63	9	ce	flowers	

	id_answer	id_common
7	4	335
8	5	335
11	6	335
18	9	335
19	10	335

```
[60]: dt_2021t1p1v3['use_repeat'] = dt_2021t1p1v3['answer'].apply(use_repeat)
```

```
[61]: dt_valid_sequence_2021t1p1v3 = dt_2021t1p1v3[(dt_2021t1p1v3['type'] == '
↳ 'submission') & (dt_2021t1p1v3['use_repeat'] == False)]
dt_valid_sequence_2021t1p1v3.drop_duplicates(subset=['password'], inplace=True)
dt_valid_sequence_2021t1p1v3['grade'].value_counts()
```

```
[61]: ce    21
cm    15
Name: grade, dtype: int64
```

```
[62]: dt_valid_repeat_2021t1p1v3 = dt_2021t1p1v3[(dt_2021t1p1v3['type'] == '
↳ 'submission') & (dt_2021t1p1v3['use_repeat'] == True)]
dt_valid_repeat_2021t1p1v3.drop_duplicates(subset=['password'], inplace=True)
dt_valid_repeat_2021t1p1v3['grade'].value_counts()
```

```
[62]: ce    38
cm    31
```

```
6em      9
Name: grade, dtype: int64
```

```
[63]: list_password_sequence_2021t1p1v3 = list(dt_valid_sequence_2021t1p1v3['password']).unique()
      ↪ list_password_repeat_2021t1p1v3 = list(dt_valid_repeat_2021t1p1v3['password']).
      ↪ unique()
      list_password_valid_2021t1p1v3 = list_password_sequence_2021t1p1v3 + list
      ↪ list_password_repeat_2021t1p1v3
```

```
[64]: dt_echec_2021t1p1v3 = list(
      ↪ dt_clean_2021[(dt_clean_2021['problem']==problem_2021t1p1) &
                    (dt_clean_2021['level'] == level) &
                    ~(dt_clean_2021['password']
      ↪ isin(list_password_valid_2021t1p1v3))
      ]
      dt_echec_2021t1p1v3.drop_duplicates(subset=['password'], inplace=True)
      dt_echec_2021t1p1v3['grade'].value_counts()
```

```
[64]: ce      7
      cm      3
      Name: grade, dtype: int64
```

```
[65]: list_password_echec_2021t1p1v3 = list(dt_echec_2021t1p1v3['password']).unique()
```

### 1.0.7 Étude des sujets qui ont validé avec une séquence ou qui sont en échec : données de 2021-1

```
[66]: dt_profils_2021 = pd.read_csv(file_profils_2021)
      dt_profils_2021 = dt_profils_2021.merge(dt_users_2021[['id_common', 'password',
      ↪ 'school_year']], on='password', how='left')
```

```
[67]: dt_profils_repeat_2021t1p1v3 = dt_profils_2021[(dt_profils_2021['problem'] == list
      ↪ problem_2021t1p1) &
            (dt_profils_2021['level'] == 'medium') &
            (dt_profils_2021['password']
      ↪ isin(list_password_repeat_2021t1p1v3))
      ]
      pd.crosstab(dt_profils_repeat_2021t1p1v3['school_year'], list
      ↪ dt_profils_repeat_2021t1p1v3['profile'])
```

```
[67]: profile      accommodation  ajustement  expert  indéfini
      school_year
      6em           4              4          1         0
      ce           28              5          4         1
      cm           17              4          10        0
```

```
[68]: # cas particuliers d'élèves ayant fait deux sessions, gérés à la main
dt_profils_repeat_2021t1p1v3 = dt_profils_2021[(dt_profils_2021['problem'] ==
↳problem_2021t1p1) &
[dt_profils_repeat_2021['id_common'].isin([5,6])]
```

```
-----
NameError                                Traceback (most recent call last)
/tmp/ipykernel_29206/3909206514.py in <module>
      1 # cas particuliers d'élèves ayant fait deux sessions, gérés à la main
----> 2 dt_profils_repeat_2021[dt_profils_repeat_2021['id_common'].isin([5,6])]

NameError: name 'dt_profils_repeat_2021' is not defined
```

```
[69]: list_passwords_non_repeat_2021t1p1v3 = list_password_sequence_2021t1p1v3 +
↳list_password_echec_2021t1p1v3
```

```
[70]: problem_2021t1p3 = '3 - Ranger les billes [white 2021]'
```

```
[71]: list_id_common_2021 = list(dt_users_2021[dt_users_2021['password'].
↳isin(list_passwords_non_repeat_2021t1p1v3)]['id_common']).unique()
```

```
[72]: len(list_id_common_2021)
```

```
[72]: 46
```

```
[74]: dt_profils_non_repeat_2021t1 = dt_profils_2021[((dt_profils_2021['problem'] ==
↳problem_2021t1p1) & (dt_profils_2021['level'] == 'medium')) |
((dt_profils_2021['problem'] ==
↳problem_2021t1p3) & (dt_profils_2021['level'].isin(['easy']))) &
(dt_profils_2021['id_common'].
↳isin(list_id_common_2021))]
dt_profils_non_repeat_2021t1 = dt_profils_non_repeat_2021t1.
↳sort_values(by='id_common')[['id_common', 'password', 'problem', 'level',
↳'profile', 'school_year']]
```

```
[75]: table_profils_non_repeat_2021t1 = pd.pivot_table(dt_profils_non_repeat_2021t1,
↳index=['id_common'], columns=['problem', 'level'], values=['profile'],
↳aggfunc=lambda x: ' '.join(x))
dt_profils_passage_seq_boucle_2021t1 = pd.
↳DataFrame(table_profils_non_repeat_2021t1)
dt_profils_passage_seq_boucle_2021t1.columns = ['1 - Planter des fleurs version
↳3',
'3 - Ranger les billes version 2']
dt_profils_passage_seq_boucle_2021t1 = dt_profils_passage_seq_boucle_2021t1.
↳merge(dt_users_2021[['id_common', 'school_year']], on='id_common',
↳how='left').drop_duplicates().sort_values(by='id_common')
```

```
dt_profils_passage_seq_boucle_2021t1.head(5)
```

```
[75]: id_common 1 - Planter des fleurs version 3 3 - Ranger les billes version 2 \
0      1      accommodation      ajustement
1      3      ajustement      ajustement
2      4      ajustement      ajustement
3      5      expert accommodation      accommodation expert
5      6      ajustement échec      indéfini expert

school_year
0      ce
1      ce
2      cm
3      cm
5      cm
```

```
[76]: # gestion des deux cas particuliers à la main
dt_profils_passage_seq_boucle_2021t1.loc[3, '1 - Planter des fleurs version_
→3']='accommodation'
dt_profils_passage_seq_boucle_2021t1.loc[5, '1 - Planter des fleurs version_
→3']='échec'
dt_profils_passage_seq_boucle_2021t1.loc[3, '3 - Ranger les billes version_
→2']='accommodation'
dt_profils_passage_seq_boucle_2021t1.loc[5, '3 - Ranger les billes version_
→2']='indéfini'
```

```
[77]: dt_profils_passage_seq_boucle_2021t1.
→loc[dt_profils_passage_seq_boucle_2021t1['3 - Ranger les billes version 2'].
→isna(), '3 - Ranger les billes version 2']='non abordé'
```

```
[78]: dt_profils_passage_seq_boucle_2021t1.head(5)
```

```
[78]: id_common 1 - Planter des fleurs version 3 3 - Ranger les billes version 2 \
0      1      accommodation      ajustement
1      3      ajustement      ajustement
2      4      ajustement      ajustement
3      5      accommodation      accommodation
5      6      échec      indéfini

school_year
0      ce
1      ce
2      cm
3      cm
5      cm
```

Profils des sujets de ce qui ont validé le puzzle 2021t1p1v3 avec une séquence



```
[79]: dt_profils_passage_seq_boucle_2021t1_ce_seq =  
      ↪dt_profils_passage_seq_boucle_2021t1[  
        (dt_profils_passage_seq_boucle_2021t1['school_year']=='ce') &  
        ~(dt_profils_passage_seq_boucle_2021t1['1 - Planter des fleurs version_↪  
          ↪3']=='échec')]
```

```
[80]: dt_profils_passage_seq_boucle_2021t1_ce_seq['3 - Ranger les billes version 2'].  
      ↪value_counts()
```

```
[80]: expert          8  
      ajustement     7  
      non abordé     3  
      échec          2  
      accommodation  1  
      Name: 3 - Ranger les billes version 2, dtype: int64
```

Profils des sujets de ce qui sont en échec sur le puzzle 2021t1plv3

```
[81]: dt_profils_passage_seq_boucle_2021t1_ce_ech =  
      ↪dt_profils_passage_seq_boucle_2021t1[  
        (dt_profils_passage_seq_boucle_2021t1['school_year']=='ce') &  
        (dt_profils_passage_seq_boucle_2021t1['1 - Planter des fleurs version_↪  
          ↪3']=='échec')]
```

```
[82]: dt_profils_passage_seq_boucle_2021t1_ce_ech['3 - Ranger les billes version 2'].  
      ↪value_counts()
```

```
[82]: non abordé      6  
      accommodation  1  
      Name: 3 - Ranger les billes version 2, dtype: int64
```

Profils des sujets de cm qui ont validé le puzzle 2021t1plv3 avec une séquence

```
[83]: dt_profils_passage_seq_boucle_2021t1_cm_seq =  
      ↪dt_profils_passage_seq_boucle_2021t1[  
        (dt_profils_passage_seq_boucle_2021t1['school_year']=='cm') &  
        ~(dt_profils_passage_seq_boucle_2021t1['1 - Planter des fleurs version_↪  
          ↪3']=='échec')]
```

```
[84]: dt_profils_passage_seq_boucle_2021t1_cm_seq['3 - Ranger les billes version 2'].  
      ↪value_counts()
```

```
[84]: ajustement     6  
      expert        5  
      accommodation  2  
      non abordé    2  
      Name: 3 - Ranger les billes version 2, dtype: int64
```

Profils des sujets de cm qui sont en échec sur le puzzle 2021t1p1v3

```
[85]: dt_profils_passage_seq_boucle_2021t1_cm_ech =  
↳dt_profils_passage_seq_boucle_2021t1[  
    (dt_profils_passage_seq_boucle_2021t1['school_year']=='cm') &  
    (dt_profils_passage_seq_boucle_2021t1['1 - Planter des fleurs version_  
↳3'] == 'échec')]
```

```
[86]: dt_profils_passage_seq_boucle_2021t1_cm_ech['3 - Ranger les billes version 2'].  
↳value_counts()
```

```
[86]: indéfini      1  
non abordé      1  
échec          1  
Name: 3 - Ranger les billes version 2, dtype: int64
```

```
[87]: dt_profils_passage_seq_boucle_2021t1['3 - Ranger les billes version 2'].  
↳value_counts()
```

```
[87]: ajustement      13  
expert            13  
non abordé       12  
accommodation    4  
échec            3  
indéfini         1  
Name: 3 - Ranger les billes version 2, dtype: int64
```

### 1.0.8 Présence du code de la boucle dans la solution xml dès la première soumission

```
[88]: def use_valid_code (valid_code, answer_xml) :  
    return valid_code in answer_xml
```

collège 2022

```
[89]: valid_code_2022_1 = '<block type="control_repeat"><value name="TIMES"><shadow_  
↳type="math_number"><field name="NUM">6</field></shadow></value><statement_  
↳name="SUBSTACK"><block type="forward"></block></statement>'  
valid1_code_2022_2 = '<block type="control_repeat"><value name="TIMES"><shadow_  
↳type="math_number"><field name="NUM">4</field></shadow></value><statement_  
↳name="SUBSTACK"><block type="east"></block></statement>'  
valid2_code_2022_2 = '<block type="control_repeat"><value_  
↳name="TIMES"><shadowuse type="math_number"><field name="NUM">7</field></  
↳shadow></value><statement name="SUBSTACK"><block type="west">'  
valid_code_2022_3 = '<block type="control_repeat"><value name="TIMES"><shadow_  
↳type="math_number"><field name="NUM">6</field></shadow></value><statement_  
↳name="SUBSTACK"><block type="forward"></block></statement>'
```

```
[90]: dt_premieres_soumissions = dt_2022p1v3.drop_duplicates(subset=['password'])
len(dt_premieres_soumissions)
```

```
[90]: 490
```

```
[91]: dt_first_t1 = dt_premieres_soumissions[dt_premieres_soumissions['problem']=='1_
↳- Chauffer le château [2022-1 white]']
dt_first_t1['use_valid_repeat'] = dt_first_t1['answer'].apply(lambda x :_
↳use_valid_code (valid_code_2022_1, x))
dt_first_t1['use_valid_repeat'].value_counts()
```

```
[91]: True      111
False      77
Name: use_valid_repeat, dtype: int64
```

```
[92]: pd.crosstab(dt_first_t1['use_valid_repeat'], dt_first_t1['grade'])
```

```
[92]: grade          3em  4em  5em  6em
use_valid_repeat
False           20   13   29   15
True            27   11   41   32
```

```
[93]: dt_first_t2 = dt_premieres_soumissions[dt_premieres_soumissions['problem']=='1_
↳- Ranger les billes [2022-2 white]']
dt_first_t2['use_valid_repeat'] = dt_first_t2['answer'].apply(lambda x :_
↳use_valid_code (valid1_code_2022_2, x) | use_valid_code (valid2_code_2022_2,_
↳x) )
dt_first_t2['use_valid_repeat'].value_counts()
```

```
[93]: False      108
True         71
Name: use_valid_repeat, dtype: int64
```

```
[94]: pd.crosstab(dt_first_t2['use_valid_repeat'], dt_first_t2['grade'])
```

```
[94]: grade          3em  4em  5em  6em
use_valid_repeat
False           39   12   32   25
True            18    7   28   18
```

```
[95]: dt_first_t3 = dt_premieres_soumissions[dt_premieres_soumissions['problem']=='1_
↳- Dans l'espace [2022-3 white]"]
dt_first_t3['use_valid_repeat'] = dt_first_t3['answer'].apply(lambda x :_
↳use_valid_code (valid_code_2022_3, x))
dt_first_t3['use_valid_repeat'].value_counts()
```

```
[95]: True      95
      False    28
      Name: use_valid_repeat, dtype: int64
```

```
[96]: pd.crosstab(dt_first_t3['use_valid_repeat'], dt_first_t3['grade'])
```

```
[96]: grade          3em  4em  5em  6em
      use_valid_repeat
      False          5   6  14   3
      True           11  14  52  18
```

école élémentaire 2021-1

```
[97]: dt_premieres_soumissions_2021 = dt_2021t1p1v3.
      ↪drop_duplicates(subset=['password'])
```

```
[98]: valid_code_2021_1 = '<block type="control_repeat"><value name="TIMES"><shadow
      ↪type="math_number"><field name="NUM">5</field></shadow></value><statement
      ↪name="SUBSTACK"><block type="forward"></block></statement>'
```

```
[99]: dt_premieres_soumissions_2021['use_valid_repeat'] =
      ↪dt_premieres_soumissions_2021['answer'].apply(lambda x : use_valid_code
      ↪(valid_code_2021_1, x))
```

```
[100]: pd.crosstab(dt_premieres_soumissions_2021['use_valid_repeat'],
      ↪dt_premieres_soumissions_2021['grade'])
```

```
[100]: grade          6em  ce  cm
      use_valid_repeat
      False          4  53  30
      True           5  13  19
```

### 1.0.9 Passage à la boucle à deux instructions

Collège 2022

```
[102]: list_problems_2022p3 = [
      '3 - Planter des fleurs [2022-1 white]',
      '3 - Ranger les billes [2022-2 white]',
      '3 - Pousser les caisses [2022-3 white]'
      ]
```

```
[103]: list_problems_2022p4 = ['4 - Planter des fleurs [2022-1 white]',
      '4 - Chauffer le château [2022-2 white]',
      '4 - Peindre le dessin [2022-3 white]']
```

```
[104]: dt_2022p4v1 = dt_clean[(dt_clean['problem'].isin(list_problems_2022p4)) &
    (dt_clean['level'] == 'basic') &
    (dt_clean['grade'].isin(classes_college)) &
    ~(dt_clean['answer'].isna())
    ]
```

```
[105]: dt_2022p4v1['use_repeat'] = dt_2022p4v1['answer'].apply(use_repeat)
dt_2022p4v1.head(5)
```

```
[105]:
```

	ID	password	id_group	id_problem	\
29	742897	33wxj8ri	603931811100778560	84414751794883575	
251	407602	33yjhik9	18142768599727689	329847446652039032	
252	407647	33yjhik9	18142768599727689	329847446652039032	
253	407679	33yjhik9	18142768599727689	329847446652039032	
382	505757	34r3vvnw	892745511171656231	980641712306666163	

	problem	level	type	\
29	4 - Peindre le dessin [2022-3 white]	basic	submission	
251	4 - Planter des fleurs [2022-1 white]	basic	attempt	
252	4 - Planter des fleurs [2022-1 white]	basic	attempt	
253	4 - Planter des fleurs [2022-1 white]	basic	submission	
382	4 - Chauffer le château [2022-2 white]	basic	attempt	

	answer	score	\
29	<xml><block type="robot_start" id="paint" x="0...	10	
251	<xml><block type="robot_start" id="flowers" x=...	0	
252	<xml><block type="robot_start" id="flowers" x=...	0	
253	<xml><block type="robot_start" id="flowers" x=...	10	
382	<xml><block type="robot_start" id="castle" x="...	0	

	date	time_session	time_attempt	time_load	context	\
29	2022-05-12 08:55:45	428	21	4	paint	
251	2022-01-26 09:36:21	1126	15	24	flowers	
252	2022-01-26 09:36:45	1150	24	15	flowers	
253	2022-01-26 09:37:00	1165	15	4	flowers	
382	2022-03-17 08:56:40	904	149	11	castle	

	grade	id_sol	id_common	use_repeat
29	5em	15	152	True
251	6em	137	31	False
252	6em	138	31	False
253	6em	139	31	False
382	4em	224	203	True

```
[106]: dt_2022p4v1_seq = dt_2022p4v1[(dt_2022p4v1['type'] == 'submission') &
    ↪(dt_2022p4v1['use_repeat'] == False)]
dt_2022p4v1_seq.drop_duplicates(subset=['password'], inplace=True)
```

```
len(dt_2022p4v1_seq)
```

[106]: 10

```
[107]: list_valid_seq_2022p4v1 = list(dt_2022p4v1_seq['password'].unique())
```

```
[108]: list_valid_seq_2022p4v1
```

```
[108]: ['33yjhik9',  
        '6hurambb',  
        '6qsy77e3',  
        '6wfy7b9e',  
        '6yr5eims',  
        'dv3ndbqb',  
        'ndgrjbfi',  
        'ucu35iw8',  
        'vhdn9vjn',  
        'x4fj9uj6']
```

```
[109]: list_seq_2022t1p4v1 = ['33yjhik9',  
                               '6hurambb',  
                               '6qsy77e3',  
                               '6wfy7b9e',  
                               '6yr5eims',  
                               'dv3ndbqb',  
                               'ndgrjbfi',  
                               'ucu35iw8',  
                               'vhdn9vjn']
```

*#rentré en dur à partir de la liste précédente*

```
[110]: list_id_common_seq_2022t1p4v1 = list(dt_clean[dt_clean['password'].  
→isin(list_seq_2022t1p4v1)]['id_common'].unique())
```

```
[111]: dt_profils_seq_t1p4 = dt_profils[(dt_profils['id_common'].  
→isin(list_id_common_seq_2022t1p4v1)) &  
                                         ((dt_profils['problem'].  
→isin(list_problems_2022p3)) & (dt_profils['level'] == 'easy')) |  
                                         ((dt_profils['problem'].  
→isin(list_problems_2022p4)) & (dt_profils['level'].isin(['basic', 'easy'])))  
                                         ]
```

```
[112]: table_profils_seq_t1p4 = pd.pivot_table(dt_profils_seq_t1p4,□  
→index=['id_common'], columns=['problem', 'level'], values=['profile'],□  
→aggfunc=lambda x: ' '.join(x))  
dt_profils_passage_deux_instr = pd.DataFrame(table_profils_seq_t1p4)  
dt_profils_passage_deux_instr.columns = ['2022t1s3v2 - Planter des fleurs',  
                                         "2022t3s3v2 - Pousser les caisses",
```

```

        '2022t2s3v2 - Ranger les billes',
        '2022t2s4v1 - Chauffer le château',
        '2022t2s4v2 - Chauffer le château',
        '2022t3s4v1 - Peindre le dessin',
        '2022t3s4v2 - Peindre le dessin',
        '2022t1s4v1 - Planter des fleurs',
        '2022t1s4v2 - Planter des fleurs']
dt_profils_passage_deux_instr = dt_profils_passage_deux_instr.
↳merge(dt_users[['id_common','school_year']], on='id_common', how='left').
↳drop_duplicates().sort_values(by='id_common')
dt_profils_passage_deux_instr = dt_profils_passage_deux_instr[[
        'id_common',
        'school_year',
        '2022t1s3v2 - Planter des fleurs',
        '2022t1s4v1 - Planter des fleurs',
        '2022t1s4v2 - Planter des fleurs',
        '2022t2s3v2 - Ranger les billes',
        '2022t2s4v1 - Chauffer le château',
        '2022t2s4v2 - Chauffer le château',
        "2022t3s3v2 - Pousser les caisses",
        '2022t3s4v1 - Peindre le dessin',
        '2022t3s4v2 - Peindre le dessin'
    ]]
dt_profils_passage_deux_instr

```

```

[112]:      id_common school_year 2022t1s3v2 - Planter des fleurs \
0          21          6em          ajustement
4          31          6em          accommodation
7          35          6em          expert
10         40          6em          ajustement
13         91          4em          ajustement
16        125          4em          accommodation
19        126          5em          ajustement
22        133          5em          expert
28        260          3em          accommodation

      2022t1s4v1 - Planter des fleurs 2022t1s4v2 - Planter des fleurs \
0          accommodation          échec
4          accommodation          expert
7          indéterminé          accommodation
10         accommodation          échec
13         accommodation          expert
16         accommodation          échec
19          indéterminé          ajustement
22          indéterminé          indéterminé
28         accommodation          accommodation

```

	2022t2s3v2 - Ranger les billes	2022t2s4v1 - Chauffer le château	\
0	expert	accommodation	
4	ajustement	accommodation	
7	ajustement	accommodation	
10	expert	accommodation	
13	expert	accommodation	
16	expert	ajustement	
19	expert	accommodation	
22	expert	ajustement	
28	NaN	NaN	

	2022t2s4v2 - Chauffer le château	2022t3s3v2 - Pousser les caisses	\
0	accommodation	NaN	
4	accommodation	accommodation	
7	expert	ajustement	
10	échec	accommodation	
13	ajustement	expert	
16	accommodation	accommodation	
19	expert	expert	
22	expert	expert	
28	NaN	NaN	

	2022t3s4v1 - Peindre le dessin	2022t3s4v2 - Peindre le dessin
0	NaN	NaN
4	ajustement	échec
7	ajustement	échec
10	expert	accommodation
13	ajustement	passé
16	expert	accommodation
19	expert	accommodation
22	expert	expert
28	NaN	NaN

```
[ ]: dt_profils_passage_deux_instr.to_csv('../data/data_these/
      ↳ passage_seq_boucle_deux_instr.csv', index=False)
```

tour 2021-1

```
[113]: dt_2021t1p4v1 = dt_clean_2021[(dt_clean_2021['problem']=='4 - Planter des
      ↳ fleurs [white 2021]') &
      (dt_clean_2021['grade'].
      ↳ isin(classes_etude_2021)) &
      (dt_clean_2021['level']=='basic') &
      (~dt_clean_2021['answer'].isna())
      ]
dt_2021t1p4v1['use_repeat'] = dt_2021t1p4v1['answer'].apply(use_repeat)
```



```
[114]: dt_2021t1p4v1['password'].nunique()
```

```
[114]: 95
```

```
[115]: dt_2021t1p4v1_valid = dt_2021t1p4v1[dt_2021t1p4v1['type']=='submission']  
dt_2021t1p4v1_valid = dt_2021t1p4v1_valid.drop_duplicates(subset=['password'])
```

```
[116]: list_passwords_valid_p4v1 = list(dt_2021t1p4v1_valid['password'].unique())
```

```
[117]: pd.crosstab(dt_2021t1p4v1_valid['use_repeat'], dt_2021t1p4v1_valid['grade'])
```

```
[117]: grade      6em  ce  cm  
use_repeat  
False         0  31  20  
True          9  10  24
```

```
[119]: list_passwords_2021t1p4v1_seq =  
↳list(dt_2021t1p4v1_valid[dt_2021t1p4v1_valid['use_repeat']==False]['password']).  
↳unique()  
list_id_common_2021t1p4v1_seq =  
↳list(dt_2021t1p4v1_valid[dt_2021t1p4v1_valid['use_repeat']==False]['id_common']).  
↳unique()
```

```
[120]: dt_2021t1p4v1[~dt_2021t1p4v1['password'].isin(list_passwords_valid_p4v1)].  
↳drop_duplicates(subset=['password'])['grade'].value_counts()
```

```
[120]: ce      1  
Name: grade, dtype: int64
```

```
[121]: list_passwords_p4v1 = list(dt_2021t1p4v1['password'].unique())
```

```
[122]: dt_clean_2021[~dt_clean_2021['password'].isin(list_passwords_p4v1)].  
↳drop_duplicates(subset=['password'])['grade'].value_counts()
```

```
[122]: ce      27  
cm       5  
Name: grade, dtype: int64
```

```
[123]: dt_2021t1p4v2 = dt_clean_2021[(dt_clean_2021['problem']=='4 - Planter des  
↳fleurs [white 2021]') &  
                                  (dt_clean_2021['level']=='easy')]  
dt_2021t1p4v2 = dt_2021t1p4v2[~dt_2021t1p4v2['answer'].isna()]  
dt_2021t1p4v2['use_repeat'] = dt_2021t1p4v2['answer'].apply(use_repeat)  
dt_2021t1p4v2_valid = dt_2021t1p4v2[dt_2021t1p4v2['type']=='submission']  
list_passwords_valid_p4v2 = list(dt_2021t1p4v2_valid['password'].unique())  
dt_2021t1p4v2_valid = dt_2021t1p4v2_valid.drop_duplicates(subset=['password'])  
pd.crosstab(dt_2021t1p4v2_valid['use_repeat'], dt_2021t1p4v2_valid['grade'])
```

```
[123]: grade      6em  ce  cm
      use_repeat
      True      9  29  42
```

```
[124]: dt_2021t1p4v2[~dt_2021t1p4v2['password'].isin(list_passwords_valid_p4v2)].
      ↪drop_duplicates(subset=['password'])['grade'].value_counts()
```

```
[124]: ce      7
      cm      2
      Name: grade, dtype: int64
```

```
[125]: list_passwords_p4v2 = list(dt_2021t1p4v2['password'].unique())
      dt_clean_2021[~dt_clean_2021['password'].isin(list_passwords_p4v2)].
      ↪drop_duplicates(subset=['password'])['grade'].value_counts()
```

```
[125]: ce      33
      cm      5
      Name: grade, dtype: int64
```

```
[126]: dt_profils_2021[(dt_profils_2021['problem']=='4 - Planter des fleurs [white_
      ↪2021]') &
      (dt_profils_2021['level']=='basic') &
      (dt_profils_2021['school_year'].isin(classes_etude_2021)) &
      (dt_profils_2021['success']==False)
      ]
```

```
[126]:      password      id_problem      problem \
61      3uut4f44  231773283829863694  4 - Planter des fleurs [white 2021]
394      8w6qyjj5  231773283829863694  4 - Planter des fleurs [white 2021]
879      jtcksywj  231773283829863694  4 - Planter des fleurs [white 2021]
1603     wwn36ph6  231773283829863694  4 - Planter des fleurs [white 2021]

      level profile  success  attempts_number  time_spent_on_question \
61      basic  passé    False                0                0
394      basic  échec    False                3               176
879      basic  passé    False                0                0
1603     basic  passé    False                0                0

      mean_time_per_block  block_count_ref  ind_speed  tx_reussite_concours \
61                        12.7            5.0  -1.000000            0.996
394                        9.6            5.0   2.666667            0.996
879                        13.5            5.0  -1.000000            0.996
1603                       12.6            5.0  -1.000000            0.996

      id_common  school_year
61              360          cm
394              331          ce
```

```
879          290          ce
1603         274          ce
```

```
[127]: len(list_passwords_2021t1p4v1_seq)
```

```
[127]: 51
```

```
[128]: dt_profils_2021t1p4v1_seq_p4v2 =
↳dt_profils_2021[(dt_profils_2021['problem']=='4 - Planter des fleurs [white_
↳2021']) &
          (dt_profils_2021['level']=='easy') &
          dt_profils_2021['password'].isin(list_passwords_2021t1p4v1_seq)]
```

```
[129]: pd.crosstab(dt_profils_2021t1p4v1_seq_p4v2['profile'],
↳dt_profils_2021t1p4v1_seq_p4v2['school_year'])
```

```
[129]: school_year    ce  cm
profile
accommodation      13   7
ajustement          2   2
expert              4   7
indéfini            1   2
passé               4   0
échec               7   2
```

```
[130]: dt_profils_2021t1p4v1_seq_p3v2 =
↳dt_profils_2021[(dt_profils_2021['problem']=='3 - Ranger les billes [white_
↳2021']) &
          (dt_profils_2021['level']=='easy') &
          dt_profils_2021['password'].isin(list_passwords_2021t1p4v1_seq)]
```

```
[131]: pd.crosstab(dt_profils_2021t1p4v1_seq_p3v2['profile'],
↳dt_profils_2021t1p4v1_seq_p3v2['school_year'])
```

```
[131]: school_year    ce  cm
profile
accommodation       5   1
ajustement          5   5
expert              12  10
indéfini            1   1
échec               3   1
```

```
[132]: list_passwords_no_success_p4v2 =
↳list(dt_profils_2021[(dt_profils_2021['problem']=='4 - Planter des fleurs_
↳[white 2021']) &
          (dt_profils_2021['level']=='easy') &
          (dt_profils_2021['school_year'].isin(classes_etude_2021)) &
```

```
(dt_profils_2021['success']==False)
]['password'])
```

```
[133]: dt_parcours_no_success_p4v2 = dt_profils_2021[(dt_profils_2021['password'].
↳isin(list_passwords_no_success_p4v2)) &
      (dt_profils_2021['problem'].isin(['3 - Ranger les billes [white_
↳2021]', '4 - Planter des fleurs [white 2021]')) &
      (dt_profils_2021['level'].isin(['basic','easy']))
      ]
```

```
[134]: pd.pivot_table(dt_parcours_no_success_p4v2, index=['id_common', 'school_year'],
↳columns=['problem', 'level'], values='profile', aggfunc=lambda x: ' '.
↳join(x))
```

```
[134]: problem          3 - Ranger les billes [white 2021]
level
id_common school_year
1          ce          accommodation      ajustement
283        ce          accommodation      accommodation
291        ce                      expert      expert
310        cm          accommodation      ajustement
320        ce          accommodation      ajustement
324        ce                      indéfini      expert
330        ce                      NaN          NaN
336        ce          ajustement      accommodation
337        ce          accommodation      accommodation
344        ce                      expert      expert
346        ce                      échec      échec
349        ce          accommodation      échec
350        ce                      échec      NaN
382        cm          expert          ajustement
```

```
problem          4 - Planter des fleurs [white 2021]
level
id_common school_year
1          ce          ajustement      passé
283        ce          expert      échec
291        ce          expert      échec
310        cm          expert      échec
320        ce          accommodation      passé
324        ce          expert      passé
330        ce          indéfini      échec
336        ce          expert      passé
337        ce          ajustement      passé
344        ce          expert      échec
346        ce          accommodation      échec
349        ce          expert      échec
```

350	ce	expert	échec
382	cm	expert	échec

[ ]:

# **Annexe 11 : traitements relatifs aux résolutions expertes**

# traitements\_resolutions\_expertes\_annexe\_these

August 1, 2024

## 1 Annexe : traitements relatifs à l'étude des résolutions expertes

```
[1]: import sys
      sys.path.append('../')

      from src import *
      import pandas as pd
      import re
      import seaborn as sns

      %load_ext autoreload
      %autoreload 2
```

pygame 2.1.2 (SDL 2.0.16, Python 3.9.15)

Hello from the pygame community. <https://www.pygame.org/contribute.html>

2024-08-01 08:44:41.011300: I tensorflow/core/platform/cpu\_feature\_guard.cc:193]

This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX2 FMA

To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.

2024-08-01 08:44:41.111740: E tensorflow/stream\_executor/cuda/cuda\_blas.cc:2981]

Unable to register cuBLAS factory: Attempting to register factory for plugin cuBLAS when one has already been registered

2024-08-01 08:44:41.596303: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot open shared object file: No such file or directory

2024-08-01 08:44:41.596357: W

tensorflow/stream\_executor/platform/default/dso\_loader.cc:64] Could not load dynamic library 'libnvinfer\_plugin.so.7'; dlerror: libnvinfer\_plugin.so.7: cannot open shared object file: No such file or directory

2024-08-01 08:44:41.596363: W

tensorflow/compiler/tf2tensorrt/utils/py\_utils.cc:38] TF-TRT Warning: Cannot dlopen some TensorRT libraries. If you would like to use Nvidia GPU with TensorRT, please make sure the missing libraries mentioned above are installed properly.

```
[2]: pd.options.mode.chained_assignment = None
```

```
[ ]: pd.set_option('display.max_rows',200) # augmente la taille maximale d'un  
      ↪ dataframe à afficher entièrement  
      # si le dataframe dépasse cette taille, il sera condensé, sinon, il sera  
      ↪ affiché entièrement
```

### 1.0.1 Initialisation

chargement du fichier des évènements repérés manuellement sur les enregistrements vidéos

```
[3]: dt = pd.read_csv('../data/data_these/schemes_expert_2024_03_02.csv')  
dt.head(5)
```

```
[3]:
```

	id_common	password	grade	id_problem	\
0	10	7gk6bygj	cm2	296089241927920425	
1	10	7gk6bygj	cm2	296089241927920425	
2	2	k5gx9bfn	ce1	296089241927920425	
3	2	k5gx9bfn	ce1	296089241927920425	
4	4	knb7bv65	cm1	296089241927920425	

	problem	num_prob	level	num_version	\
0	3 - Peindre le motif [white 2021]	3	basic	1	
1	3 - Peindre le motif [white 2021]	3	easy	2	
2	3 - Peindre le motif [white 2021]	3	basic	1	
3	3 - Peindre le motif [white 2021]	3	easy	2	
4	3 - Peindre le motif [white 2021]	3	basic	1	

	time_spent_on_question	ind_speed	Évènements	\
0	40	-0.35	2 / 3 / 1p / 6 / 4 / 7sg / 5a	
1	42	-0.57	7sg / 9 / 2 / 3 / 7pg / 1p / 6 / 5a	
2	50	-0.23	1p / C1 / lat / 4 / 2 / 3 / 6 / 5b	
3	64	-0.38	lat / 9 / 7pg / 1p / 2 / 3 / 6 / 5a	
4	50	0.16	lat / 1p / 2 / 4 / 3v / 6 / 5a	

	mean_time_per_block	block_count_ref	tx_reussite_concours	profile	\
0	12.3	5.0	0.991	expert	
1	12.3	8.0	0.983	expert	
2	13.0	5.0	0.991	expert	
3	13.0	8.0	0.983	expert	
4	8.6	5.0	0.991	expert	

	success	attempts_number
0	True	1
1	True	1
2	True	1
3	True	1



4      True                                  1

```
[4]: dt.columns
```

```
[4]: Index(['id_common', 'password', 'grade', 'id_problem', 'problem', 'num_prob',  
         'level', 'num_version', 'time_spent_on_question', 'ind_speed',  
         'Évènements', 'mean_time_per_block', 'block_count_ref',  
         'tx_reussite_concours', 'profile', 'success', 'attempts_number'],  
        dtype='object')
```

```
[5]: dt['problem'].value_counts()
```

```
[5]: 3 - Ranger les billes [2022-2 white]            34  
4 - Peindre le dessin [2022-3 white]            29  
4 - Chauffer le château [2022-2 white]        25  
3 - Planter des fleurs [2022-1 white]        23  
4 - Planter des fleurs [2022-1 white]        21  
3 - Pousser les caisses [2022-3 white]       16  
4 - Planter des fleurs [white 2021]        16  
3 - Peindre le motif [white 2021]        12  
3 - Ranger les billes [white 2021]        11  
5 - Distribuer le bois [2022-2 white]       11  
4 - Rejoindre la fusée [white 2021]       10  
5 - Collecter les pierres [2022-1 white]    10  
3 - Rejoindre la fusée [white 2021]       8  
4 - Jouer aux dominos [white 2021]        7  
5 - Dans l'espace [2022-3 white]        7  
5 - Planter des fleurs [white 2021]       3  
5 - Atteindre le drapeau [white 2021]      1  
Name: problem, dtype: int64
```

### 1.0.2 Structuration de la liste de puzzles

```
[6]: list_puzzles = {}  
list_puzzles["1_instr"] = [  
    ("3 - Ranger les billes [2022-2 white]", 1),  
    ("3 - Ranger les billes [2022-2 white]", 2),  
    ("3 - Planter des fleurs [2022-1 white]", 1),  
    ("3 - Planter des fleurs [2022-1 white]", 2),  
    ("3 - Pousser les caisses [2022-3 white]", 1),  
    ("3 - Pousser les caisses [2022-3 white]", 2),  
    ("3 - Peindre le motif [white 2021]", 1),  
    ("3 - Peindre le motif [white 2021]", 2),  
    ("3 - Ranger les billes [white 2021]", 1),  
    ("3 - Ranger les billes [white 2021]", 2),  
    ("3 - Rejoindre la fusée [white 2021]", 1),  
    ("3 - Rejoindre la fusée [white 2021]", 2),
```

```

]
list_puzzles["2_instr"] = [
  ("4 - Peindre le dessin [2022-3 white]", 1),
  ("4 - Chauffer le château [2022-2 white]", 1),
  ("4 - Planter des fleurs [2022-1 white]", 1),
  ("4 - Planter des fleurs [white 2021]", 1),
  ("4 - Planter des fleurs [white 2021]", 2),
  ("4 - Rejoindre la fusée [white 2021]", 1),
  ("4 - Rejoindre la fusée [white 2021]", 2),
  ("4 - Jouer aux dominos [white 2021]", 1),
  ("4 - Jouer aux dominos [white 2021]", 2)
]
list_puzzles["c11"] = list_puzzles["1_instr"] + list_puzzles["2_instr"]
list_puzzles["c12"] = [
  ("4 - Planter des fleurs [white 2021]", 3),
  ("4 - Rejoindre la fusée [white 2021]", 3),
  ("4 - Jouer aux dominos [white 2021]", 3),
  ("4 - Planter des fleurs [2022-1 white]", 2),
  ("4 - Planter des fleurs [2022-1 white]", 3),
  ("4 - Chauffer le château [2022-2 white]", 2),
  ("4 - Chauffer le château [2022-2 white]", 3),
  ("4 - Peindre le dessin [2022-3 white]", 2),
  ("4 - Peindre le dessin [2022-3 white]", 3),
]
list_puzzles["c13"] = [
  ("5 - Planter des fleurs [white 2021]", 1),
  ("5 - Planter des fleurs [white 2021]", 2),
  ("5 - Atteindre le drapeau [white 2021]", 1),
  ("5 - Atteindre le drapeau [white 2021]", 2),
  ("5 - Collecter les pierres [2022-1 white]", 1),
  ("5 - Collecter les pierres [2022-1 white]", 2),
  ("5 - Distribuer le bois [2022-2 white]", 1),
  ("5 - Distribuer le bois [2022-2 white]", 2),
  ("5 - Dans l'espace [2022-3 white]", 1),
  ("5 - Dans l'espace [2022-3 white]", 2)
]
list_puzzles["c123"] = list_puzzles["c12"] + list_puzzles["c13"]
list_puzzles["c1123"] = list_puzzles["c11"] + list_puzzles["c12"] +
↳list_puzzles["c13"]

```

```
[7]: liste_cles = ["1_instr", "2_instr", "c11", "c12", "c13", "c123", "c1123"]
```

```
[8]: set_cat = {}
for cle in liste_cles :
    set_cat[cle] = set(list_puzzles[cle])
```

### 1.0.3 Nombre de procédures expertes

```
[9]: data = {}
for cle in liste_cles :
    data[cle] = dt[(dt.apply(lambda row: (row['problem'], row['num_version'])
    ↪in set_cat[cle], axis=1)) &
        (dt['Évènements'].str.contains('5')) & # pour éliminer les
    ↪confrontations écartées, car ne comportant pas d'exécution
        (dt['Évènements'].str.contains('2')) & # contient une boucle
        (dt['Évènements'].str.contains('4')) & # pour écarter les
    ↪puzzles où le nb de répétitions est 10
        ~(dt['Évènements'].str.contains('det')) # pour écarter les
    ↪consultations du tutoriel
        ]
    print (" nombre de procédures expertes pour " + cle + " : " +
    ↪str(len(data[cle])))
```

```
nombre de procédures expertes pour 1_instr : 90
nombre de procédures expertes pour 2_instr : 57
nombre de procédures expertes pour c11 : 147
nombre de procédures expertes pour c12 : 38
nombre de procédures expertes pour c13 : 26
nombre de procédures expertes pour c123 : 64
nombre de procédures expertes pour c1123 : 211
```

### 1.0.4 Étude de l'ordre des actions lors de la programmation de la boucle

```
[10]: def extraire_phases(chaine, regex_phases):
    chiffres = re.findall(regex_phases, chaine)
    chiffres_str = ''.join(chiffres)
    return re.sub(r'(\.)\1+', r'\1', chiffres_str) # on enlève les doublons
    ↪quand ils se suivent
```

```
[11]: for cle in liste_cles :
    data[cle]['phases_toutes'] = data[cle]['Évènements'].apply(lambda x :
    ↪extraire_phases(x, r'[1234679]'))
```

```
[12]: # On extrait les chiffres, qui correspondent aux phases, en ne gardant qu'une
    ↪occurrence quand plusieurs chiffres semblables se suivent,
# car ici c'est l'ordre qui nous intéresse
for cle in liste_cles :
    data[cle]['phases_scheme_boucle'] = data[cle]['Évènements'].apply(lambda x :
    ↪extraire_phases(x, r'[1234]'))
```

```
[13]: len(data["c1123"]['phases_scheme_boucle'].value_counts())
```

```
[13]: 38
```

```
[14]: data["cl123"]["phases_scheme_boucle"].value_counts().head(15)
```

```
[14]: 2143      61
      1243      39
      2314      20
      243       17
      1423      15
      1234       9
      234        6
      3214       6
      31243      3
      324        3
      2134       3
      423        2
      3124       2
      214313     1
      24324243   1
      Name: phases_scheme_boucle, dtype: int64
```

Étude des caractéristiques de la séquence d'évènements

```
[15]: def caract_count (dict_dt, colonne, caract_regex) :
      for cle in liste_cles :
          nb_true = len(dict_dt[cle][dict_dt[cle][colonne].str.
      ↪contains(caract_regex)])
          total = len(dict_dt[cle])
          print (cle + " : " +
                str(nb_true) +
                " sur " +
                str(total) +
                " soit une fréquence de " +
                str(round(nb_true/total, 2)))
```

```
[16]: # Le dénombrement a lieu avant l'identification de motif
      caract_count (data, 'Évènements', r'4.*3')
```

```
1_instr : 69 sur 90 soit une fréquence de 0.77
2_instr : 44 sur 57 soit une fréquence de 0.77
cl1 : 113 sur 147 soit une fréquence de 0.77
cl2 : 25 sur 38 soit une fréquence de 0.66
cl3 : 19 sur 26 soit une fréquence de 0.73
cl23 : 44 sur 64 soit une fréquence de 0.69
cl123 : 157 sur 211 soit une fréquence de 0.74
```

```
[17]: # L'identification de motif a lieu avant le dénombrement
      caract_count (data, 'Évènements', r'3.*4')
```

```
1_instr : 24 sur 90 soit une fréquence de 0.27
```

2\_instr : 16 sur 57 soit une fréquence de 0.28  
cl1 : 40 sur 147 soit une fréquence de 0.27  
cl2 : 16 sur 38 soit une fréquence de 0.42  
cl3 : 12 sur 26 soit une fréquence de 0.46  
cl23 : 28 sur 64 soit une fréquence de 0.44  
cl123 : 68 sur 211 soit une fréquence de 0.32

```
[18]: # Procédures expertes accompagnées de verbalisations  
      caract_count(data, 'Évènements', 'v')
```

1\_instr : 29 sur 90 soit une fréquence de 0.32  
2\_instr : 17 sur 57 soit une fréquence de 0.3  
cl1 : 46 sur 147 soit une fréquence de 0.31  
cl2 : 9 sur 38 soit une fréquence de 0.24  
cl3 : 8 sur 26 soit une fréquence de 0.31  
cl23 : 17 sur 64 soit une fréquence de 0.27  
cl123 : 63 sur 211 soit une fréquence de 0.3

```
[19]: # Présence d'un schème de dénombrement  
      caract_count(data, 'Évènements', '1')
```

1\_instr : 83 sur 90 soit une fréquence de 0.92  
2\_instr : 49 sur 57 soit une fréquence de 0.86  
cl1 : 132 sur 147 soit une fréquence de 0.9  
cl2 : 25 sur 38 soit une fréquence de 0.66  
cl3 : 21 sur 26 soit une fréquence de 0.81  
cl23 : 46 sur 64 soit une fréquence de 0.72  
cl123 : 178 sur 211 soit une fréquence de 0.84

```
[20]: # Schème de dénombrement avec pointage et verbalisation  
      caract_count(data, 'Évènements', '1pv')
```

1\_instr : 20 sur 90 soit une fréquence de 0.22  
2\_instr : 10 sur 57 soit une fréquence de 0.18  
cl1 : 30 sur 147 soit une fréquence de 0.2  
cl2 : 6 sur 38 soit une fréquence de 0.16  
cl3 : 4 sur 26 soit une fréquence de 0.15  
cl23 : 10 sur 64 soit une fréquence de 0.16  
cl123 : 40 sur 211 soit une fréquence de 0.19

```
[21]: # Schème de dénombrement avec verbalisation seule  
      caract_count(data, 'Évènements', '1v')
```

1\_instr : 4 sur 90 soit une fréquence de 0.04  
2\_instr : 1 sur 57 soit une fréquence de 0.02  
cl1 : 5 sur 147 soit une fréquence de 0.03  
cl2 : 1 sur 38 soit une fréquence de 0.03  
cl3 : 1 sur 26 soit une fréquence de 0.04

cl23 : 2 sur 64 soit une fréquence de 0.03  
cl123 : 7 sur 211 soit une fréquence de 0.03

```
[22]: # Verbalisation de la totalisation au moment de remplir le champ du bloc répéter  
      caract_count(data, 'Évènements', '4v')
```

1\_instr : 11 sur 90 soit une fréquence de 0.12  
2\_instr : 5 sur 57 soit une fréquence de 0.09  
cl1 : 16 sur 147 soit une fréquence de 0.11  
cl2 : 4 sur 38 soit une fréquence de 0.11  
cl3 : 3 sur 26 soit une fréquence de 0.12  
cl23 : 7 sur 64 soit une fréquence de 0.11  
cl123 : 23 sur 211 soit une fréquence de 0.11

```
[23]: # Verbalisation de l'édition du motif  
      caract_count(data, 'Évènements', '3v')
```

1\_instr : 11 sur 90 soit une fréquence de 0.12  
2\_instr : 6 sur 57 soit une fréquence de 0.11  
cl1 : 17 sur 147 soit une fréquence de 0.12  
cl2 : 7 sur 38 soit une fréquence de 0.18  
cl3 : 5 sur 26 soit une fréquence de 0.19  
cl23 : 12 sur 64 soit une fréquence de 0.19  
cl123 : 29 sur 211 soit une fréquence de 0.14

```
[24]: # Verbalisation de la mobilisation de la répétition  
      caract_count(data, 'Évènements', '2v')
```

1\_instr : 1 sur 90 soit une fréquence de 0.01  
2\_instr : 2 sur 57 soit une fréquence de 0.04  
cl1 : 3 sur 147 soit une fréquence de 0.02  
cl2 : 0 sur 38 soit une fréquence de 0.0  
cl3 : 1 sur 26 soit une fréquence de 0.04  
cl23 : 1 sur 64 soit une fréquence de 0.02  
cl123 : 4 sur 211 soit une fréquence de 0.02

```
[25]: # Pointage ou survol de la grille pour prise d'information, en-dehors du schème  
      ↳ de dénombrement  
      caract_count(data, 'Évènements', '7')
```

1\_instr : 33 sur 90 soit une fréquence de 0.37  
2\_instr : 17 sur 57 soit une fréquence de 0.3  
cl1 : 50 sur 147 soit une fréquence de 0.34  
cl2 : 24 sur 38 soit une fréquence de 0.63  
cl3 : 19 sur 26 soit une fréquence de 0.73  
cl23 : 43 sur 64 soit une fréquence de 0.67  
cl123 : 93 sur 211 soit une fréquence de 0.44

```
[26]: # Verbalisation lors de la prise d'information --> vérifier le codage
      caract_count(data, 'Évènements', '7pv')
```

```
1_instr : 1 sur 90 soit une fréquence de 0.01
2_instr : 0 sur 57 soit une fréquence de 0.0
cl1 : 1 sur 147 soit une fréquence de 0.01
cl2 : 0 sur 38 soit une fréquence de 0.0
cl3 : 0 sur 26 soit une fréquence de 0.0
cl23 : 0 sur 64 soit une fréquence de 0.0
cl123 : 1 sur 211 soit une fréquence de 0.0
```

```
[27]: # Prise d'information juste avant le placement d'une instruction dans la boucle
      caract_count(data, 'phases_toutes', '73')
```

```
1_instr : 1 sur 90 soit une fréquence de 0.01
2_instr : 3 sur 57 soit une fréquence de 0.05
cl1 : 4 sur 147 soit une fréquence de 0.03
cl2 : 14 sur 38 soit une fréquence de 0.37
cl3 : 12 sur 26 soit une fréquence de 0.46
cl23 : 26 sur 64 soit une fréquence de 0.41
cl123 : 30 sur 211 soit une fréquence de 0.14
```

### 1.0.5 Étude des instructions hors de la boucle

```
[28]: dt_avant = {}
      for cle in liste_cles :
          dt_avant[cle] = data[cle][data[cle]['Évènements'].str.contains('9')]
          print (cle + ' : ' + str(len(dt_avant[cle])))
```

```
1_instr : 50
2_instr : 5
cl1 : 55
cl2 : 2
cl3 : 3
cl23 : 5
cl123 : 60
```

```
[29]: dt_apres = {}
      for cle in liste_cles :
          dt_apres[cle] = data[cle][data[cle]['Évènements'].str.contains('6')]
          print (cle + ' : ' + str(len(dt_apres[cle])))
```

```
1_instr : 84
2_instr : 14
cl1 : 98
cl2 : 1
cl3 : 5
```

```
c123 : 6
c1123 : 104
```

```
[30]: dt_hors = {}
      for cle in liste_cles :
          dt_hors[cle] = data[cle][[(data[cle]['Évènements'].str.contains('6')) |
                                   (data[cle]['Évènements'].str.contains('9'))
                                   ]
          ]
      print (cle + ' : ' + str(len(dt_hors[cle])))
```

```
1_instr : 84
2_instr : 15
c11 : 99
c12 : 3
c13 : 8
c123 : 11
c1123 : 110
```

```
[31]: dt_pas_hors = {}
      for cle in liste_cles :
          dt_pas_hors[cle] = data[cle][(~data[cle]['Évènements'].str.contains('6')) &
                                       (~data[cle]['Évènements'].str.contains('9'))
                                       ]
      print (cle + ' : ' + str(len(dt_pas_hors[cle])))
```

```
1_instr : 6
2_instr : 42
c11 : 48
c12 : 35
c13 : 18
c123 : 53
c1123 : 101
```

```
[32]: #Prise d'information lorsqu'il y a des instructions avant la boucle
      caract_count(dt_avant, 'Évènements', '7')
```

```
1_instr : 25 sur 50 soit une fréquence de 0.5
2_instr : 3 sur 5 soit une fréquence de 0.6
c11 : 28 sur 55 soit une fréquence de 0.51
c12 : 2 sur 2 soit une fréquence de 1.0
c13 : 2 sur 3 soit une fréquence de 0.67
c123 : 4 sur 5 soit une fréquence de 0.8
c1123 : 32 sur 60 soit une fréquence de 0.53
```

```
[33]: #Prise d'information lorsqu'il y a des instructions après la boucle
      caract_count(dt_apres, 'Évènements', '7')
```

```
1_instr : 33 sur 84 soit une fréquence de 0.39
```



2\_instr : 4 sur 14 soit une fréquence de 0.29  
cl1 : 37 sur 98 soit une fréquence de 0.38  
cl2 : 1 sur 1 soit une fréquence de 1.0  
cl3 : 5 sur 5 soit une fréquence de 1.0  
cl23 : 6 sur 6 soit une fréquence de 1.0  
cl123 : 43 sur 104 soit une fréquence de 0.41

```
[34]: #Prise d'information lorsqu'il y a des instructions hors de la boucle  
      caract_count(dt_hors, 'Évènements', '7')
```

1\_instr : 33 sur 84 soit une fréquence de 0.39  
2\_instr : 4 sur 15 soit une fréquence de 0.27  
cl1 : 37 sur 99 soit une fréquence de 0.37  
cl2 : 3 sur 3 soit une fréquence de 1.0  
cl3 : 7 sur 8 soit une fréquence de 0.88  
cl23 : 10 sur 11 soit une fréquence de 0.91  
cl123 : 47 sur 110 soit une fréquence de 0.43

```
[35]: #Prise d'information lorsqu'il n'y a pas d'instructions hors de la boucle  
      caract_count(dt_pas_hors, 'Évènements', '7')
```

1\_instr : 0 sur 6 soit une fréquence de 0.0  
2\_instr : 13 sur 42 soit une fréquence de 0.31  
cl1 : 13 sur 48 soit une fréquence de 0.27  
cl2 : 21 sur 35 soit une fréquence de 0.6  
cl3 : 12 sur 18 soit une fréquence de 0.67  
cl23 : 33 sur 53 soit une fréquence de 0.62  
cl123 : 46 sur 101 soit une fréquence de 0.46

```
[36]: # Prise d'information juste avant de placer une instruction dans la boucle  
      # Programmes sans instruction hors de la boucle  
      caract_count(dt_pas_hors, 'phases_toutes', '73')
```

1\_instr : 0 sur 6 soit une fréquence de 0.0  
2\_instr : 2 sur 42 soit une fréquence de 0.05  
cl1 : 2 sur 48 soit une fréquence de 0.04  
cl2 : 13 sur 35 soit une fréquence de 0.37  
cl3 : 10 sur 18 soit une fréquence de 0.56  
cl23 : 23 sur 53 soit une fréquence de 0.43  
cl123 : 25 sur 101 soit une fréquence de 0.25

```
[37]: # Prise d'information juste avant de placer une instruction dans la boucle  
      # Programmes avec instructions hors de la boucle  
      caract_count(dt_hors, 'phases_toutes', '73')
```

1\_instr : 1 sur 84 soit une fréquence de 0.01  
2\_instr : 1 sur 15 soit une fréquence de 0.07  
cl1 : 2 sur 99 soit une fréquence de 0.02

```
cl2 : 1 sur 3 soit une fréquence de 0.33
cl3 : 2 sur 8 soit une fréquence de 0.25
cl23 : 3 sur 11 soit une fréquence de 0.27
cl123 : 5 sur 110 soit une fréquence de 0.05
```

```
[38]: # Prise d'information juste avant de placer une instruction dans la boucle
# Programmes avec instructions avant la boucle
caract_count(dt_avant, 'phases_toutes', '73')
```

```
1_instr : 1 sur 50 soit une fréquence de 0.02
2_instr : 1 sur 5 soit une fréquence de 0.2
cl1 : 2 sur 55 soit une fréquence de 0.04
cl2 : 1 sur 2 soit une fréquence de 0.5
cl3 : 1 sur 3 soit une fréquence de 0.33
cl23 : 2 sur 5 soit une fréquence de 0.4
cl123 : 4 sur 60 soit une fréquence de 0.07
```

```
[39]: # Prise d'information juste avant de placer une instruction dans la boucle
# Programmes avec instructions après la boucle
caract_count(dt_apres, 'phases_toutes', '73')
```

```
1_instr : 1 sur 84 soit une fréquence de 0.01
2_instr : 1 sur 14 soit une fréquence de 0.07
cl1 : 2 sur 98 soit une fréquence de 0.02
cl2 : 0 sur 1 soit une fréquence de 0.0
cl3 : 1 sur 5 soit une fréquence de 0.2
cl23 : 1 sur 6 soit une fréquence de 0.17
cl123 : 3 sur 104 soit une fréquence de 0.03
```

```
[40]: # Prise d'information à l'issue de la programmation de la boucle,
# afin de placer les instructions après la boucle
caract_count(dt_apres, 'phases_toutes', '76')
```

```
1_instr : 13 sur 84 soit une fréquence de 0.15
2_instr : 1 sur 14 soit une fréquence de 0.07
cl1 : 14 sur 98 soit une fréquence de 0.14
cl2 : 0 sur 1 soit une fréquence de 0.0
cl3 : 3 sur 5 soit une fréquence de 0.6
cl23 : 3 sur 6 soit une fréquence de 0.5
cl123 : 17 sur 104 soit une fréquence de 0.16
```

```
[41]: # Prise d'information à l'issue de la programmation de la boucle,
# afin de placer les instructions après la boucle
caract_count(dt_avant, 'phases_toutes', '79')
```

```
1_instr : 19 sur 50 soit une fréquence de 0.38
2_instr : 2 sur 5 soit une fréquence de 0.4
cl1 : 21 sur 55 soit une fréquence de 0.38
```

```
cl2 : 1 sur 2 soit une fréquence de 0.5
cl3 : 1 sur 3 soit une fréquence de 0.33
cl23 : 2 sur 5 soit une fréquence de 0.4
cl123 : 23 sur 60 soit une fréquence de 0.38
```

```
[42]: # Prise d'information à l'issue de la programmation de la boucle,
# afin de placer les instructions après la boucle
caract_count(dt_avant, 'phases_toutes', '97')
```

```
1_instr : 14 sur 50 soit une fréquence de 0.28
2_instr : 1 sur 5 soit une fréquence de 0.2
cl1 : 15 sur 55 soit une fréquence de 0.27
cl2 : 2 sur 2 soit une fréquence de 1.0
cl3 : 1 sur 3 soit une fréquence de 0.33
cl23 : 3 sur 5 soit une fréquence de 0.6
cl123 : 18 sur 60 soit une fréquence de 0.3
```

### 1.0.6 Analyse des procédures expertes spécifiques aux puzzles de la classe 1

```
[43]: data["cl1"]["phases_scheme_boucle"].value_counts()
```

```
[43]: 2143      47
1243      32
1423      14
2314       9
1234       9
243        7
234        4
2134       3
31243      3
3124       2
423        2
3214       2
324        2
12314      1
3142       1
21423      1
213434     1
12143      1
124313     1
1324       1
21243      1
234123     1
12413      1
23143      1
Name: phases_scheme_boucle, dtype: int64
```

Lorsqu'il y a plusieurs occurrences de 3 ou 4, on les enlève à la main dans le report pour le tableau

dans le texte de la thèse

```
[44]: # identification du motif en dernière position
data["c11"]['phases_scheme_boucle'].str.endswith('3').sum()
```

```
[44]: 112
```

```
[45]: # ordre des phases lorsque l'identification du motif n'est pas en dernière_
      ↪position
data["c11"][~data["c11"]['phases_scheme_boucle'].str.
      ↪endswith('3')]['phases_scheme_boucle'].value_counts()
```

```
[45]: 2314      9
      1234      9
      234       4
      2134      3
      324       2
      3214      2
      3124      2
      1324      1
      12314     1
      213434    1
      3142      1
      Name: phases_scheme_boucle, dtype: int64
```

```
[46]: # le schème de dénombrement est présent
data["c11"]['phases_scheme_boucle'].str.contains('1').value_counts()
```

```
[46]: True      132
      False    15
      Name: phases_scheme_boucle, dtype: int64
```

```
[47]: org_sujets = pd.DataFrame(pd.crosstab(data["c11"]['id_common'],
      ↪data["c11"]['phases_scheme_boucle']).to_records())
org_sujets.head(5)
```

```
[47]:
```

	id_common	12143	12314	1234	12413	1243	124313	1324	1423	21243	...	\
0	1	0	0	0	0	2	0	0	0	0	...	...
1	2	0	0	0	0	2	0	0	5	0	...	...
2	3	0	1	1	0	6	0	0	0	0	...	...
3	4	0	0	0	0	5	0	0	0	0	...	...
4	5	0	0	0	0	5	0	0	0	0	...	...

	23143	234	234123	243	3124	31243	3142	3214	324	423
0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0

```
4      0      0      0      0      0      1      0      0      0      0
```

```
[5 rows x 25 columns]
```

Analyse des procédures expertes : boucle simple

```
[48]: len(data["1_instr"][data["1_instr"]['phases_scheme_boucle'].str[-1]=='3'])
```

```
[48]: 69
```

```
[49]: org_sujets = pd.DataFrame(pd.crosstab(data["1_instr"]['id_common'],  
→data["1_instr"]['phases_scheme_boucle']).to_records())  
org_sujets
```

```
[49]:
```

	id_common	12143	12314	1234	12413	1243	124313	1324	1423	21243	\
0	1	0	0	0	0	2	0	0	0	0	
1	2	0	0	0	0	1	0	0	2	0	
2	3	0	1	1	0	4	0	0	0	0	
3	4	0	0	0	0	3	0	0	0	0	
4	5	0	0	0	0	3	0	0	0	0	
5	6	0	0	1	0	0	0	0	0	0	
6	7	0	0	1	1	0	0	0	0	0	
7	8	0	0	0	0	1	0	0	0	0	
8	9	0	0	1	0	3	1	1	0	0	
9	10	0	0	1	0	0	0	0	0	0	
10	11	0	0	0	0	0	0	0	0	0	
11	12	0	0	0	0	1	0	0	0	0	
12	13	0	0	0	0	0	0	0	0	1	
13	16	0	0	0	0	1	0	0	0	0	
14	17	0	0	0	0	0	0	0	0	0	
15	18	0	0	0	0	0	0	0	0	0	
16	19	0	0	2	0	0	0	0	0	0	
17	20	0	0	0	0	0	0	0	0	0	
18	133	0	0	0	0	0	0	0	0	0	
19	134	1	0	0	0	1	0	0	0	0	
20	135	0	0	0	0	0	0	0	4	0	
21	136	0	0	0	0	0	0	0	2	0	

	2134	2143	2314	234	234123	243	31243	3214	324
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	2	0	0	0	0	0	0	0
4	0	2	0	0	0	0	1	0	0
5	1	2	2	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0
7	0	4	0	0	0	0	0	0	0

8	1	2	0	0	0	0	0	0	0
9	0	5	2	0	0	0	0	0	0
10	0	2	0	0	0	0	0	0	0
11	0	4	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	0	0
14	0	0	0	1	0	1	0	0	0
15	0	0	0	0	0	1	0	0	2
16	0	1	0	0	0	0	1	0	0
17	0	1	1	1	0	1	0	0	0
18	0	4	0	0	0	0	0	0	0
19	0	2	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0

```
[50]: data["1_instr"]['phases_toutes'].value_counts().head(15)
```

```
[50]: 921436      7
      12436      7
      21436      6
      912436     5
      7921436    3
      1243       2
      14236      2
      2143       2
      92436      2
      921463     2
      142376     2
      923146     2
      12346      2
      9797123476  1
      97972921436  1
      Name: phases_toutes, dtype: int64
```

Analyse des procédures expertes : boucles avec deux instructions

```
[51]: data["2_instr"]['phases_toutes'].value_counts().head(15)
```

```
[51]: 2143      10
      1243      7
      243       3
      1423      3
      72143     3
      72314     2
      21436     2
      12436     2
      71243     2
```

```
73124      2
23143      1
2364       1
23614      1
721346     1
12346      1
Name: phases_toutes, dtype: int64
```

Analyse des procédures expertes : classe 2

```
[52]: data["c12"]['phases_toutes'].value_counts().head(15)
```

```
[52]: 2314      3
      243      3
      1243     3
      3214     3
      2143     2
      72743    1
      724367242743  1
      72714243  1
      2473     1
      272473   1
      7324     1
      7237374   1
      247243743  1
      7142373   1
      7271979147373  1
Name: phases_toutes, dtype: int64
```

```
[53]: data["c12"]['phases_scheme_boucle'].value_counts()
```

```
[53]: 243      7
      2143    6
      2314    6
      1243    6
      3214    4
      243243  1
      24243   1
      32314   1
      1423    1
      2424343  1
      234     1
      324     1
      214243  1
      24324243  1
Name: phases_scheme_boucle, dtype: int64
```

```
[54]: # On enlève les boucles imbriquées
dt_cl2_boucle_simple = data["cl2"][data["cl2"]['phases_scheme_boucle'].str.
    ↪count('2') == 1]
dt_cl2_boucle_simple['phases_scheme_boucle'].value_counts()
```

```
[54]: 243      7
      2143     6
      2314     6
      1243     6
      3214     4
      32314    1
      1423     1
      234      1
      324      1
      Name: phases_scheme_boucle, dtype: int64
```

```
[55]: len(dt_cl2_boucle_simple)
```

```
[55]: 33
```

```
[56]: # On compte les enregistrements pour lesquels le dénombrement a lieu avant
    ↪l'identification de motif
len(dt_cl2_boucle_simple[dt_cl2_boucle_simple['phases_scheme_boucle'].str.
    ↪contains('4.*3')])
```

```
[56]: 20
```

```
[57]: # On compte les enregistrements pour lesquels l'identification de motif a lieu
    ↪avant le dénombrement
len(dt_cl2_boucle_simple[dt_cl2_boucle_simple['phases_scheme_boucle'].str.
    ↪contains('3.*4')])
```

```
[57]: 13
```

Analyse des procédures expertes : classe 3

```
[58]: data["cl3"]['phases_scheme_boucle'].value_counts()
```

```
[58]: 2143      8
      2314     5
      243      3
      214313    1
      23234314243  1
      12432432423  1
      21431243     1
      232134314    1
      2142143     1
```



```
124313214      1
234             1
1243           1
3234           1
Name: phases_scheme_boucle, dtype: int64
```

```
[59]: # On enlève les boucles imbriquées
dt_cl3_boucle_simple = data["cl3"][data["cl3"]['phases_scheme_boucle'].str.
    ↪count('2') == 1]
dt_cl3_boucle_simple['phases_scheme_boucle'].value_counts()
```

```
[59]: 2143      8
      2314      5
      243       3
      214313    1
      234       1
      1243      1
      3234      1
Name: phases_scheme_boucle, dtype: int64
```

```
[60]: len(dt_cl3_boucle_simple)
```

```
[60]: 20
```

```
[61]: # On compte les enregistrements pour lesquels le dénombrement a lieu avant
    ↪l'identification de motif
len(dt_cl3_boucle_simple[dt_cl3_boucle_simple['phases_scheme_boucle'].str.
    ↪contains('4.*3')])
```

```
[61]: 13
```

```
[62]: # On compte les enregistrements pour lesquels l'identification de motif a lieu
    ↪avant le dénombrement
len(dt_cl3_boucle_simple[dt_cl3_boucle_simple['phases_scheme_boucle'].str.
    ↪contains('3.*4')])
```

```
[62]: 7
```

### 1.0.7 Camemberts de répartition de caractéristiques pour les trois classes

```
[63]: def graphique_repartition(dt_caract, list_couleurs) :

    # Données pour les graphiques
    tailles = [30, 30, 30]
    etiquettes = dt_caract.index
    cols = dt_caract.columns
```

```

# Création de la figure et des axes
fig, axs = plt.subplots(1, len(cols), figsize=(12, 4))

# Création des graphiques en forme de "pie"
for i in range(len(cols)):
    axs[i].pie(dt_caract[cols[i]], autopct='%1.1f%%', startangle=90,
    → colors= list_couleurs)
    axs[i].set_xlabel(cols[i], fontsize=12)

fig.legend(etiquettes, loc='upper center', fontsize=10)

#plt.savefig("export_fig.png")

plt.show()

```

```

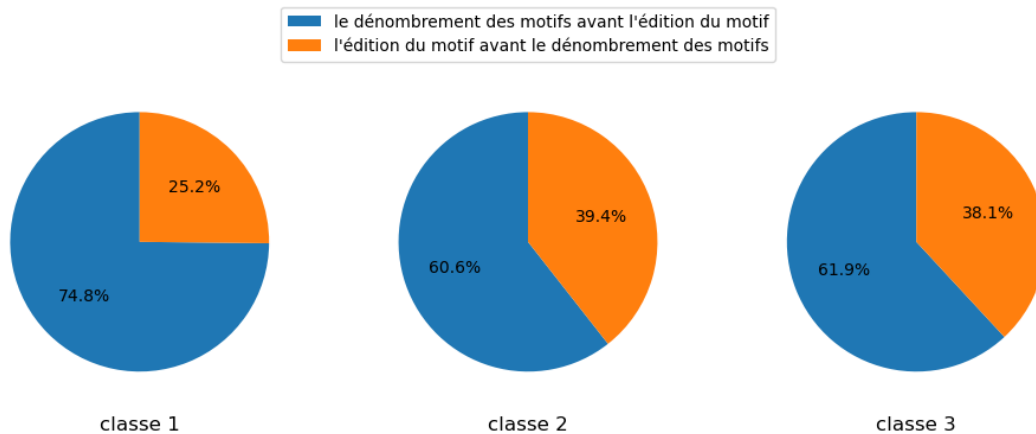
[64]: ordre_phases = pd.DataFrame({'classe 1':[110, 37], 'classe 2':[20, 13], 'classe
    → 3':[13, 8]},
                                index=['le dénombrement des motifs avant l\'édition
    → du motif',
                                        'l\'édition du motif avant le dénombrement
    → des motifs'])

```

```

[65]: graphique_repartition(ordre_phases, ["#1f77b4", "#ff7f0e"])

```

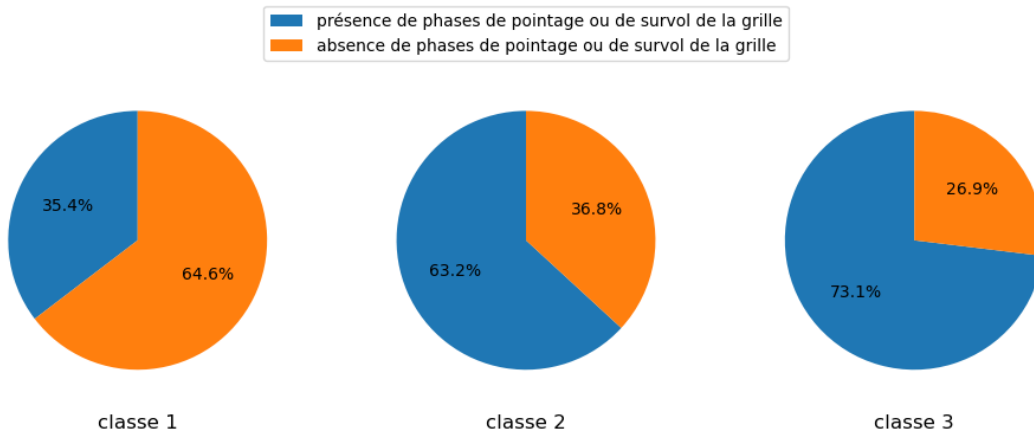


```

[66]: analyse_grille = pd.DataFrame({'classe 1':[52, 95], 'classe 2':[24, 14],
    → 'classe 3':[19, 7]},
                                index=['présence de phases de pointage ou de survol
    → de la grille',
                                        'absence de phases de pointage ou de survol
    → de la grille'])

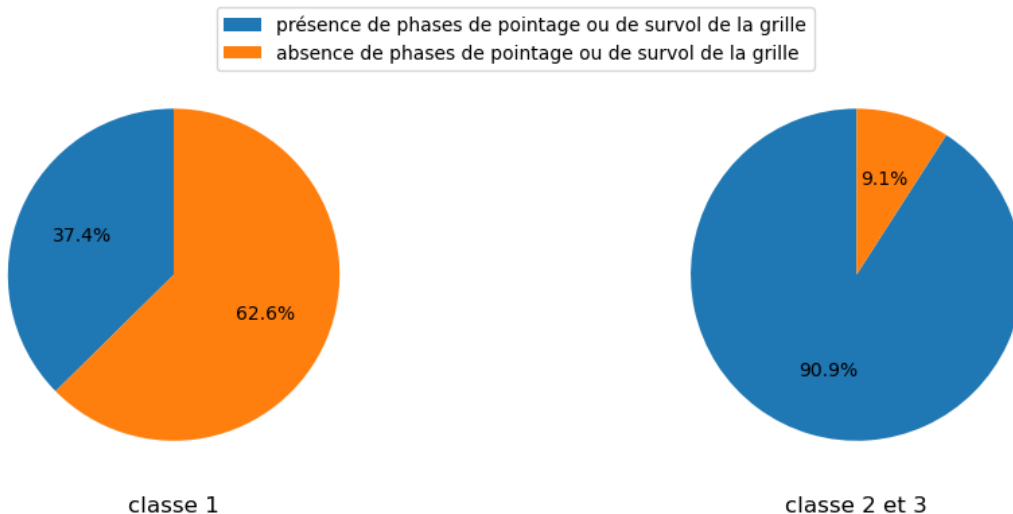
```

```
[67]: graphique_repartition(analyse_grille, ["#1f77b4", "#ff7f0e"])
```



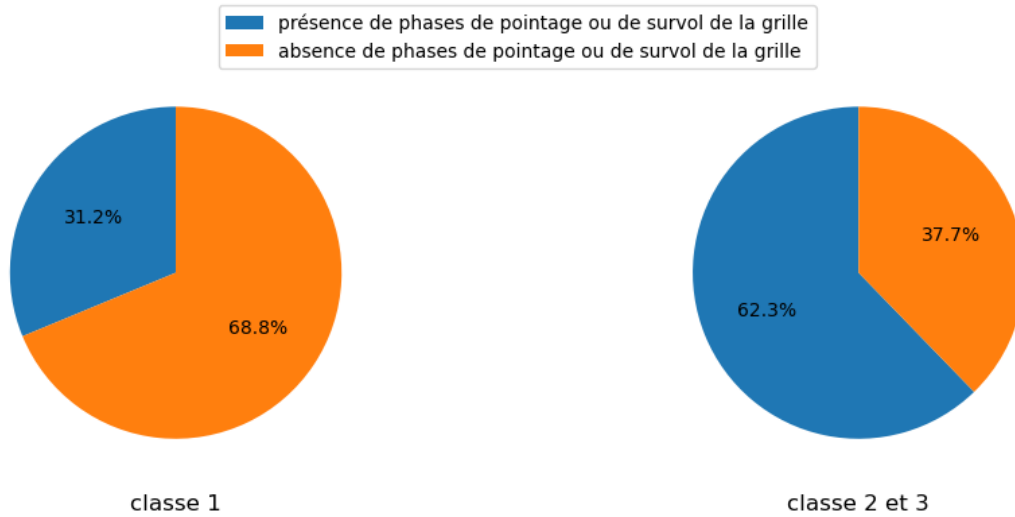
```
[68]: analyse_grille_hors = pd.DataFrame({'classe 1':[37, 62], 'classe 2 et 3':[10, 1]},  
                                         index=['présence de phases de pointage ou de survol de la grille',  
                                                'absence de phases de pointage ou de survol de la grille'])
```

```
[69]: # Pour les programmes avec des instructions hors de la boucle  
graphique_repartition(analyse_grille_hors, ["#1f77b4", "#ff7f0e"])
```



```
[70]: analyse_grille_sans_hors = pd.DataFrame({'classe 1': [15, 33], 'classe 2 et 3':  
→ [33, 20]},  
→ index=['présence de phases de pointage ou de survol_␣  
→ de la grille',  
→ 'absence de phases de pointage ou de survol_␣  
→ de la grille'])
```

```
[71]: # Pour les programmes sans instructions hors de la boucle  
graphique_repartition(analyse_grille_sans_hors, ["#1f77b4", "#ff7f0e"])
```



## **Annexe 12 : planches de la tâche de catégorisation**

# Planches pour la catégorisation des situations

Ci-dessous sont présentées les planches qui ont été utilisées pour la tâche de catégorisation de grilles de puzzles de programmation. Chaque planche est accompagnée des critères de l'analyse a priori qui ont guidé sa conception.

## Planche n°1

Cette planche est toujours la première qui est présentée au sujet. Elle vise de permettre des possibilités de catégorisation variées.

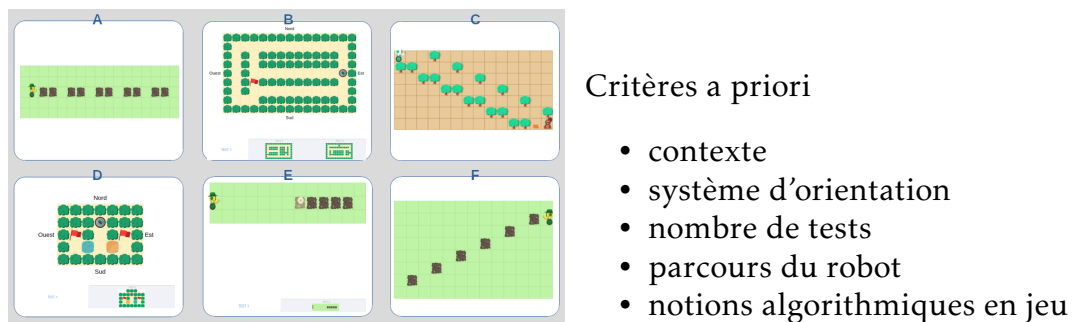


FIGURE 1 – Planche n°1

## Planche n°2

Cette planche est présentée lorsque le sujet catégorise les grilles de la planche n°1 selon le contexte.

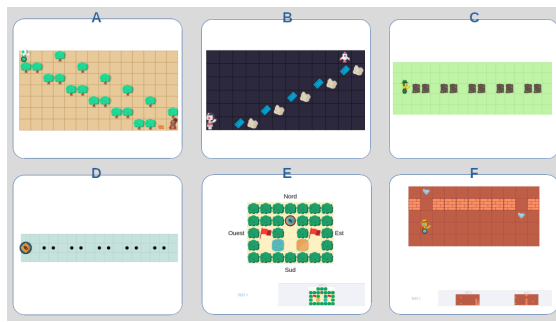


FIGURE 2 – Planche n°2

Critères a priori

- **contexte : toutes les grilles sont issues d'un contexte différent**
- système d'orientation
- nombre de tests
- parcours du robot
- notions algorithmiques en jeu

### Planche n°3

Cette planche est présentée lorsque les grilles d'une planche sont catégorisées selon le contexte.

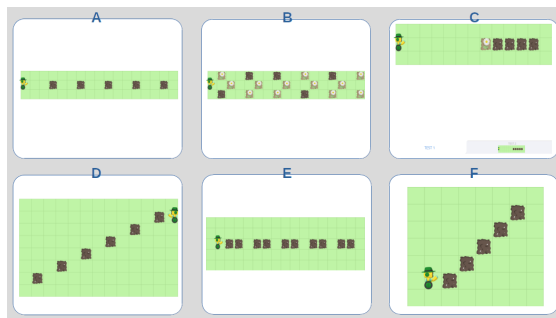


FIGURE 3 – Planche n°3

Critères a priori

- **contexte : toutes les grilles sont issues du même contexte**
- système d'orientation
- nombre de tests
- parcours du robot
- notions algorithmiques en jeu

### Planche n°4

Cette planche est présentée lorsque les grilles de la planche précédemment présentée ont été catégorisées selon le nombre de tests. La catégorisation selon le contexte est possible.

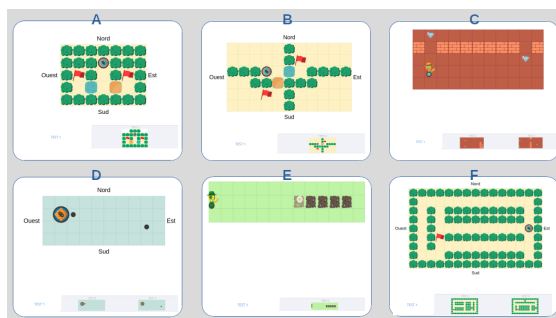


FIGURE 4 – Planche n°4

Critères a priori

- contexte
- système d'orientation
- **nombre de tests : toutes les grilles ont plusieurs tests**
- parcours du robot
- notions algorithmiques en jeu

## Planche n°5

Cette planche est présentée lorsque le sujet revient à une catégorisation selon le contexte, après avoir catégorisé selon le nombre de tests.

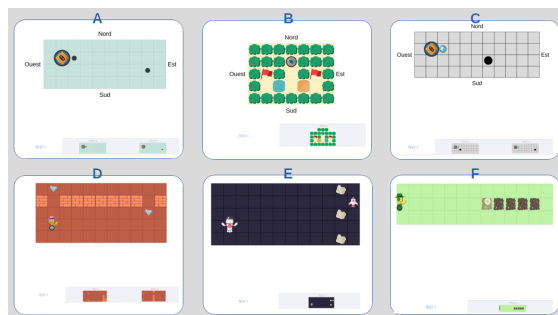


FIGURE 5 – Planche n°5

### Critères a priori

- **contexte : toutes les grilles sont issues d'un contexte différent**
- système d'orientation
- **nombre de tests : toutes les grilles contiennent plusieurs tests**
- parcours du robot
- notions algorithmiques en jeu



## Planche n°6

Cette planche est présentée lorsque les grilles de la planche précédemment présentée ont été catégorisées en fonction du type de parcours du robot : en ligne, en escalier... La catégorisation selon le contexte est possible. Celle en fonction du nombre de tests ne l'est pas.

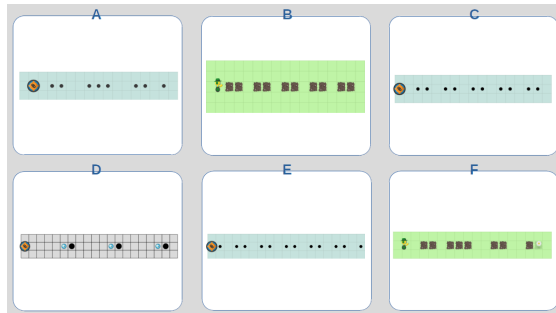


FIGURE 6 – Planche n°6

Critères a priori

- contexte
- système d'orientation
- nombre de tests
- **parcours du robot : tous les parcours sont unidirectionnels**
- notions algorithmiques en jeu

## Planche n°7

Cette planche est présentée lorsque le sujet revient à une catégorisation selon le contexte après avoir catégorisé selon le type de parcours du robot. La catégorisation selon le nombre de tests est neutralisée aussi.

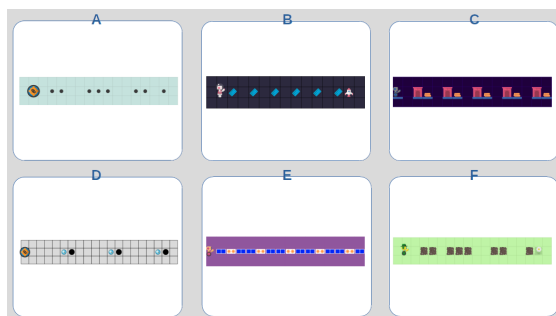


FIGURE 7 – Planche n°7

Critères a priori

- **contexte : toutes les planches sont issues d'un contexte différent**
- système d'orientation
- **nombre de tests : pas de grille avec tests multiples**
- **parcours du robot : tous les parcours sont unidirectionnels**
- notions algorithmiques en jeu

## Planche n°8

Cette planche est présentée lorsque le sujet revient à une catégorisation selon le contexte après avoir catégorisé selon le type de parcours du robot, ou qu'il trouve un critère de catégorisation qui regroupe certains contextes. La catégorisation selon le nombre de tests est neutralisée aussi.



FIGURE 8 – Planche n°8

Critères a priori

- **contexte : toutes les planches sont issues du même contexte**
- système d'orientation
- **nombre de tests : pas de grille avec tests multiples**
- **parcours du robot : tous les parcours sont unidirectionnels**
- notions algorithmiques en jeu

## Planche n°9

Cette planche vise plus particulièrement l'identification de motifs, précisément la disposition des motifs visuels sur la grille. Plusieurs autres critères de catégorisation sont neutralisés.

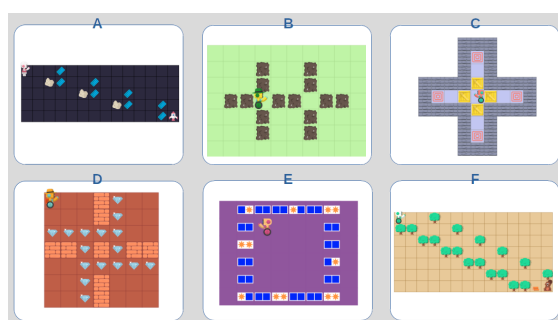


FIGURE 9 – Planche n°9

Critères a priori

- **contexte : toutes les planches sont issues d'un contexte différent**
- **système d'orientation : toutes les grilles sont en orientation relative**
- **nombre de tests : pas de grille avec tests multiples**
- **parcours du robot : tous les parcours sont en 2D, avec un motif qui se répète, mais avec des dispositions différentes des motifs visuels sur la grille.**
- notions algorithmiques en jeu

## Planche n°10

Cette planche vise l'identification de motifs, en lien avec le parcours du robot. Plusieurs autres critères de catégorisation sont neutralisés.

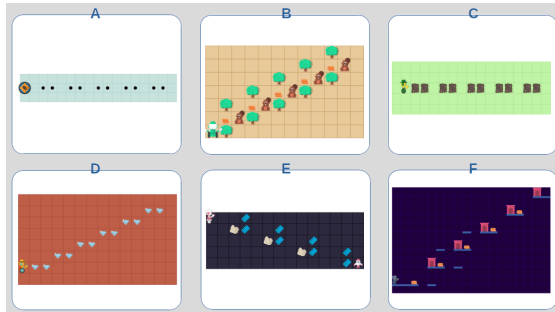


FIGURE 10 – Planche n°10

Critères a priori

- **contexte : toutes les planches sont issues d'un contexte différent**
- système d'orientation
- **nombre de tests : pas de grille avec tests multiples**
- parcours du robot
- notions algorithmiques en jeu

## Planche n°11

Cette planche vise à amener le sujet à distinguer les notions algorithmiques en jeu : répétition vs structure conditionnelle, à partir de la présence ou l'absence de régularité sur la grille.

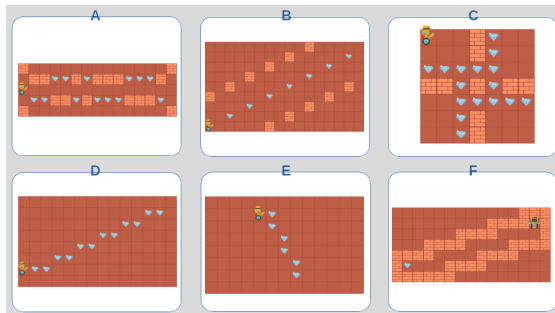


FIGURE 11 – Planche n°11

Critères a priori

- **contexte : toutes les planches sont issues du même contexte**
- système d'orientation
- **nombre de tests : pas de grille avec tests multiples**
- parcours du robot
- notions algorithmiques en jeu

## Planche n°12

Cette planche vise à amener le sujet à distinguer les notions algorithmiques en jeu : répétition vs procédure, à partir de la présence ou l'absence de régularité sur la grille.

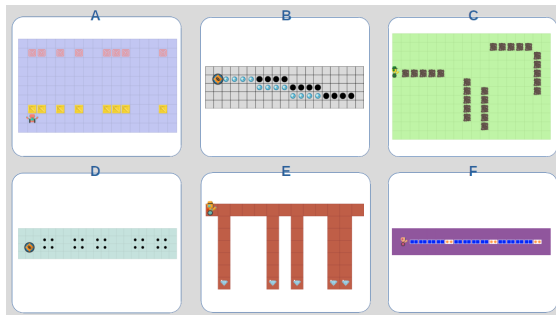


FIGURE 12 – Planche n°12

Critères a priori

- **contexte : toutes les planches sont issues d'un contexte différent**
- système d'orientation
- **nombre de tests : pas de grille avec tests multiples**
- parcours du robot
- notions algorithmiques en jeu

## Planche n°13

Cette planche vise à amener le sujet à distinguer les notions algorithmiques en jeu : boucle simple vs boucle avec plusieurs instructions, en fonction de la structure du motif visuel.

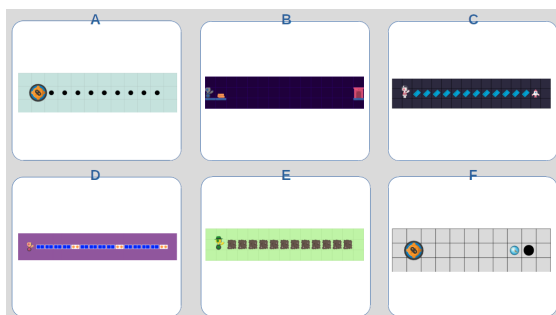


FIGURE 13 – Planche n°13

Critères a priori

- **contexte : toutes les planches sont issues d'un contexte différent**
- système d'orientation
- **nombre de tests : pas de grille avec tests multiples**
- **parcours du robot : tous les parcours sont unidirectionnels**
- notions algorithmiques en jeu

## **Annexe 13 : transcription de la catégorisation des situations**

# Annexe : Tâche de catégorisation de situations

## Introduction

La tâche de catégorisation de situations de programmation s'est déroulée pendant l'expérimentation complémentaire de l'été 2022. Elle suit immédiatement une session de 45 minutes, au cours de laquelle les sujets ont été confrontés à des puzzles avec des grilles du même type. Les entretiens ont été enregistrés par Zoom, les planches étant montrées aux sujets par partage d'écran.

La tâche de la catégorisation de situations de programmation consiste à « dire à l'expérimentatrice les grilles qui vont ensemble, et à dire ce qui fait mettre ces grilles ensemble », autrement dit à concevoir des catégories, en donnant le critère de catégorisation.

18 sujets sont impliqués, ce qui a permis de collecter 168 catégorisations réparties comme suit :

- Léa : 8 planches
- Élana : 8 planches
- Ali : 10 planches
- Alix : 10 planches
- Lou : 13 planches
- Louna : 9 planches
- Alexis : 11 planches
- Colin : 13 planches
- Axel : 6 planches
- Noé : 9 planches
- Paola : 9 planches
- Sam : 9 planches
- Louane : 13 planches
- Mickaël : 8 planches
- Timéo : 10 planches
- Robin : 12 planches
- Ugo : 10 planches

## Extrait des transcriptions

Le document de transcriptions original comporte 180 pages. Nous en donnons un extrait avec quelques catégorisations représentatives.

Pour tous les dialogues, E désigne l'expérimentatrice et S désigne le sujet.

### Sujet 2 : Léa, CE2

#### Planche 1



[phase en amont où le sujet ne voyait pas les grilles, car il n'était pas retourné sur la fenêtre de Zoom]

E – Tu as six grilles avec des robots qui ressemblent.. qui sont des grilles que tu as rencontrées.. ou peut-être pas tout à fait.. lors des parcours de programmation..

S – La B je l'ai vue.. la A aussi.. la D aussi.. la E aussi.. euh pas la E

E – Alors tu ne les as peut-être pas toutes vues aujourd'hui.. c'est depuis que tu en fais.. ça fait longtemps que tu en..

S – Ah bah du coup y'a la C.. la F que je reconnais.. y'a que la E que je ne reconnais pas

E – Ok.. alors c'est pas grave s'il y en a que tu reconnais pas.. c'est possible que tu ne les aies pas vues.. mais elles ressemblent en tout cas à des grilles que tu as vues sur les parcours de programmation

S – Et par exemple la E c'est une seize fois trois

E – Mm

S – C'est l'autre..

E – Et donc la question.. c'est donc quelles sont les grilles que tu rangerais ensemble.. et de m'expliquer comment tu choisies.. qu'est-ce qui te fais les mettre ensemble

S – En fait là j'ai deux idées

E – Alors vas-y je t'écoute

S – Je mettrais la A avec la E et la F.. la B avec la D et la C toute seule

E – Oui et comment tu fais ça ?

S – Parce que c'est par sujet

E – Ok oui.. donc ça c'est ta première idée .. très bien

S – Et la deuxième.. c'est de mettre par exemple la C avec la F..

E – Oui

S – La E avec la A..

E – Oui

S – La B avec la C et la F aussi.. et la D toute seule

E – Alors et explique moi

S – Parce que c'est par à peu près la taille.. c'est à peu près la même à chaque

E – D'accord.. la taille de la grille.. ok.. la B la C et la F qu'est-ce que tu dirais pour la taille ?

S - Hum.. à peu près..

E – À peu près hein..

S – Je dirais sept fois.. sept fois quinze

E – Mm.. ok .. d'accord

### **Planche 3**



S – Toutes du même sujet

E – Mm

S – F avec D parce que c'est des escaliers

E – Mm

S – A avec E parce que c'est la même chose sauf que dans E y'a un bout de terre en plus

E – Oui.. qu'est-ce qui est la même chose dans A et dans E alors ?

S – C'est que c'est répéter Mm planter avancer planter etc

E – D'accord.. donc c'est quoi qui est pareil ? Tu appellerais ça comment répéter avancer planter

S – C'est répétitif en fait

E – Oui

S – Et après je mettrais aussi C avec B parce que y'a des cases avec des fleurs et des cases sans fleur

E – Ah oui.. très bien



**Planche 8**



S – Ça va plutôt vite de faire ça quand même..

E – Mm

S – Euh.. mm.. euh.. [silance].. je mettrais peut-être la A et la E ensemble.. puis après la B la C la D et la F ensemble

E – Mm

S – La A et la E ensemble parcequ'il y a des cases qui les.. qui les.. y'a des cases entre les.. y'a des cases de terre qui séparent les cases de terre

E – Mm

S – Enfin on peut comme la B la C et la D.. enfin y'a moins de cases de terre.. puis après la B la C la D et la F y'a beaucoup plus de terre et.. et voilà

E – Alors quand tu dis beaucoup plus de terre

S – Ben c'est qu'il y a plus de blocs.. enfin plus de blocs.. plus de cases où y'a de la terre

E – Et ta limite elle est..

S – Ma limite.. quelle limite ?

E – Ben quand tu dis plus.. comment tu.. comment tu sépars tes deux groupes ?

S – Mm.. bah par exemple.. dans le groupe A et le groupe E y'a cinq euh.. cases de terre et par exemple dans les autres c'est euh.. au-dessus de cinq ou..

E – D'accord mais tu me dis.. au début tu me dis que dans le A et le E il y a des cases vides entre les cases de terre

S – Oui

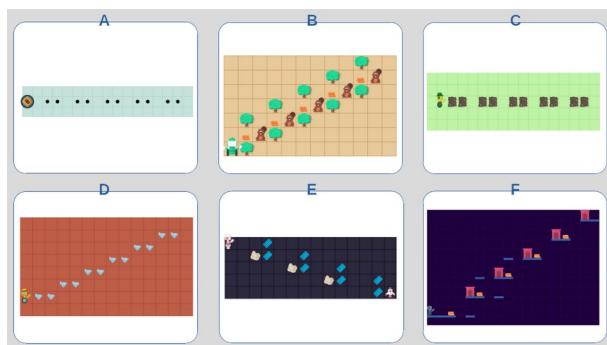
E – C'est ça.. entre toutes les cases de terre

S – Oui enfin.. y'en a pas une y'en a plusieurs.. par exemple sur la B la C et la D y'en aura une ou deux

E – D'accord.. alors que sur la A et la E y'en a plusieurs.. au moins deux

S – Oui voilà

## Planche 10



[silence]

S – Euh.. ben j’vais faire le même principe s’ils sont en diagonale ou pas diagonale.. donc j’vais mettre D euh.. B D E F ensemble et A et C ensemble

E – D’accord donc.. si tu me rappelles l’explication..

S – Ben déjà B D E F les objets ou.. ou les p’tits feux enfin les cheminées sont en diagonale.. et puis après A et C c’est en horizontal donc.. ouais voilà

E – D’accord donc là la disposition on va dire des éléments.. en A et C sont horizontals.. et en B D E F sont en diagonale.. ok

S - Oui

**Planche 1**



E - J'vais te montrer des planches comme ça

S – Mm

E – Avec six grilles de parcours de programmation que tu as déjà fait ou qui ressemblent à ceux que tu as déjà fait.. donc les grilles.. pour pouvoir les appeler.. elles ont une lettre A B C D E F

S – Mm

E – Et ce que je te demande.. c'est de faire des groupes.. c'est de ranger ensemble.. les grilles.. que tu penses qui ont quelque chose en commun.. et de me dire.. ce qui te.. comment tu réfléchis pour les ranger.. ok ?

S – Ok.. donc faut que j'range ça ?

E – Voilà.. donc on commence par cette grille-là.. comment tu rangerais les grilles de cette planche ?

S – Ben moi j'mettrais A E F ensemble et DBC ensemble parce que c'est..

E – Mm

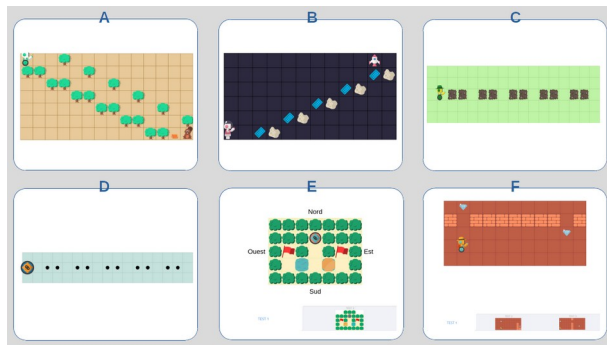
S – Ben là c'est de la terre.. A E F c'est de la terre.. et D B C c'est des arbres.. mais on pourrait aussi tous les mettre dans la même famille parce que c'est tous en rapport avec la nature.. enfin c'est tous avec.. faut de la terre pour planter un arbre enfin voilà donc moi.. soit je les mettrais comme j'ai dit.. soit les mettre tous ensemble.. ce serait pas rangé uniquement

E – Alors y'a plein de manières de ranger.. y'a beaucoup d'idées différentes qui peuvent être exprimées.. donc oui pourquoi.. pourquoi pas.. et la précision que je ne t'ai pas encore donnée aussi.. c'est que c'est toi qui décide de la taille des groupes.. donc tu peux faire des groupes de un deux trois jusqu'à six.. c'est à dire que tu peux en laisser tout seul.. tu peux me dire je les mets tous ensemble.. ou tu peux faire des groupes de deux de trois de quatre..

S – Ben soit je mets A E F ensemble et D B C soit je les mets tous ensemble

E – D'accord parfait

## Planche 2



E – C’est toujours la même question.. les planches vont changer mais la question est toujours la même.. c’est comment tu rangerais ces grilles de parcours de programmation que tu as déjà vues ?

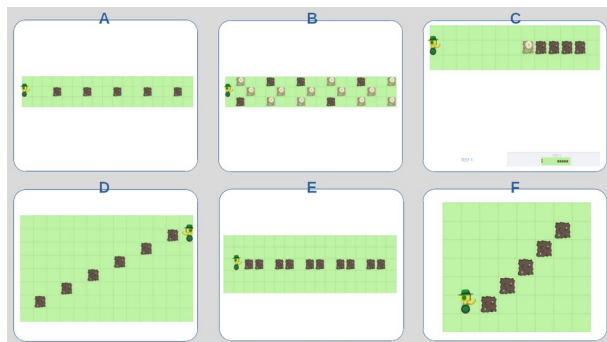
S – Ouuh là.. je rangerais A B C E F ensemble et D tout seul parce que D c’est.. D il faut faire avec nord sud est ouest.. et pas avec avancer reculer tourner à droite.. ah non E aussi.. donc E et D ensemble et A B C F ensemble

E – D’accord donc D et E.. donc.. pour les directions nord sud est ouest.. et les autres ?

S – Les autres parce que c’est avancer reculer tourner à droite

E - Ok

## Planche 3



E – La suivante..

S – Alors.. euh.. [silence] j’mettrais B et E ensemble.. C tout seul.. et F D A ensemble

E – Oui

S – Parce que F D A.. y’a qu’une seule case et des cases entre la prochaine case où il y a de la terre

E – Oui

S – B et E y’a.. il faut.. enfin y’a.. c’est plus compliqué.. y’a plus de cases de terre collées.. il faut faire plus un grand algorithme difficile.. et C tout seul parce qu’il y a plein de cases à côté et euh.. y’a déjà.. y’a plusieurs tests

E – Alors.. si je résume donc.. A D et F c’est ça ?

S – Oui

E – Après B et C c’est ça ?

S – Euh oui

E – Oui

S – Euh non B et E

E – B et E.. et C tout seul

S – Oui

E – Alors B et E.. tu as parlé qu'il faut faire un grand algorithme.. est-ce que tu peux m'en dire plus ?

S – Ben c'est pas comme le A où faut faire juste avancer avancer et déposer une graine.. c'est pas ça.. faut faire ça.. voilà c'est pas un truc aussi simple.. faut faire avancer tourner à droite.. semer une graine tourner à gauche avancer avancer tourner à gauche avancer [baisse la voix]

E – Donc ça c'est pour..

S – C'est plus compliqué

E – Ça c'est pour le B

S – Oui

E – Et le E ?

S – Et le E ben c'est parce que.. c'est pas comme le A.. y'a deux cases de terre collées et du coup bah c'est.. non en vrai je mettrais juste le C tout seul parce que.. et le B tout seul

E – Ok

S – Parce que le E ben c'est aussi simple..

E – Oui.. alors.. mais alors.. comment .. quel critère tu utilises pour mettre A E D et F ensemble.. parce que celui que tu as donné au début.. qui est qu'il y a du vert entre les zones de terre ne fonctionne pas pour E

S – Oui

E – Moi je veux bien mais il faut que tu changes la manière dont tu l'expliques

S – Non mais en fait je sais la manière comment je vais faire.. j'vais prendre un.. on va faire A E C ensemble parce que c'est sur une seule ligne droite

E – Mm

S – Et que D B et F.. c'est en diagonale.. donc ben je vais ranger comme ça.. A E et C.. et D B F

E – Alors je suis d'accord.. mais je ne vois pas de diagonale sur B

S – Sur B ?.. bah en fait j'veux dire que les cases de terre où il faut poser une graine ne sont pas alignées.. sur la même ligne

E – D'accord.. ok donc là.. là ça me convient.. donc AC F les cases de terre sont alignées sur la même ligne et elles ne le sont pas sur B D et F

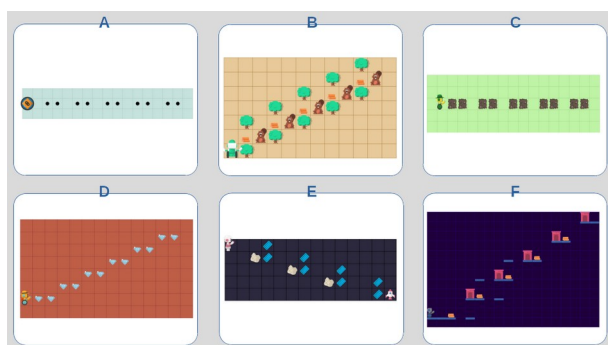
S – Euh

E – C'est ça ?

S – Oui

E - Ok

## Planche 10



S – Alors.. mm.. [silence].. je mettrais A C D E ensemble.. ah ben car.. à chaque fois qu’il faut faire quelque chose y’a deux cases collées.. enfin deux..

E – Mm

S – Cases collées ouais.. et B et F ensemble parce que c’est déjà un genre d’escalier qu’il faut faire.. mm voilà

E – Alors.. oui.. alors je vois bien pour les deux cases collées.. je vois bien aussi pour les escaliers.. sauf que.. pour D et E tu as un escalier aussi

S – Oui [silence].. euh.. on va mettre A et C ensemble.. et B E D F ensemble.. parce que c’est question d’escalier

E – Oui.. et A et C alors ?

S – A et C ben ensemble parce que y’a deux cases à chaque fois.. par exemple le robot dans A il faut qu’il peigne deux cases d’affilée et après une case vide deux cases d’affilée etc.. et dans C c’est pareil sauf qu’à la place de peindre il faut juste qu’il dépose une.. une graine donc.. c’est pareil le nombre de cases y’a entre l’écart et le nombre de cases où il faut qu’il fasse quelque chose dessus

E – Mm.. alors ce.. parce que ton critère.. tu me dis.. il y a deux cases de suite où il faut que le robot fasse quelque chose dessus.. pour E on a ça aussi

S – Oui.. mais c’est pas aligné sur la même ligne

E – D’accord donc le critère ce serait quoi ?

S – Que euh.. pour A et C que ce soit aligné sur la même ligne avec deux cases à chaque fois où il faut qu’il peigne ou qu’il dépose une graine

E – Mm

S – Et que dans le reste.. eh ben c’est en escalier et que ce n’est pas tout aligné sur la même ligne

E – D’accord.. et qu’est-ce que tu appelles en escalier ?

S – Ben quand genre que ça fasse comme le D.. il faut monter un genre d’escalier où faut ramasser les.. ça fait un peu comme des marches

E – Mm

S – Voilà

E – Donc et.. qu’est-ce qui fait comme des marches ?

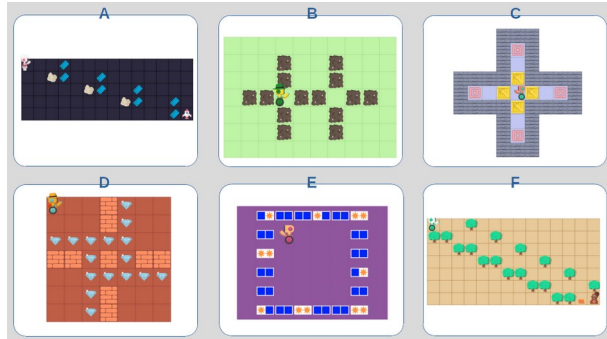
S – Comment ?

E – Mm.. qu'est-ce qui fait comme des marches.. ce sont les.. ce sont les éléments ?

S – Euh oui les éléments ça fait comme des marches

E – D'accord.. ok

### Planche 9



[silence]

S – Je mettrais B C D ensemble.. ah non.. pas ensemble.. euh.. B et C ensemble.. ben parce que il est.. non ouais non.. j'veis mettre B C E ensemble parce que le robot est au milieu.. du parcours

E – Mm

S – Et A D F ensemble.. parce que le robot.. est au bord du parcours.. enfin genre il est pas dans.. au milieu du parcours.. il est sur le bord à la dernière case.. on le voit

E – Ok bon ça c'est.. c'est un critère aussi donc là.. tu as considéré la position du robot

S – Oui

E – Très bien.. tu vois tu as plein.. tu peux avoir plein d'idées différentes.. plein de manières de ranger.. et moi ce qui m'intéresse c'est quelles.. comment.. ta manière de ranger les grilles de parcours de programmation que tu as déjà fait

### Planche 5



[silence]

S – Euh.. je mettrais A et C ensemble.. parce que.. ben euh le robot doit effectuer le même parcours.. faire avancer un nombre de fois puis tourner à droite.. donc j'les mettrais ensemble pour ça.. après euh.. je mettrais.. le B et le D ensemble.. parce que euh.. il y a plusieurs tests.. et que à chaque fois faut.. mettre en.. si machin truc faire sinon.. parce que ben sinon.. on fonce dans des murs dans des

tests.. et ça marche pour d'autres.. voilà j'les mettrais ensemble pour ça.. et.. F.. euh.. est-ce que c'est possible qu'il y a un parcours qui soit dans deux familles ?

E – Alors.. non faut éviter

S – Ok donc ben soit F tout seul et E tout seul.. ou soit les mettre ensemble pour euh.. pour euh.. deux tests

E – Alors.. oui mais B aussi a deux tests

S – Ouais non mais du coup les mettre ou.. non les mettre ensemble ou.. mm.. et ben les mettre tous seuls.. mettre F tout seul et E tout seul

E – Alors quel est le critère que tu utilises pour ranger du coup ?

S – Pour ranger E et F ?

E – Ben pour ranger.. toutes les grilles

S – Ah ben.. L A et le C c'est parce que il faut faire le même parcours.. le B et le D c'est parce qu'il faut.. y'a dans.. y'a dans les.. plusieurs tests.. plusieurs tests.. eh ben il y a.. faut faire un si sinon.. sinon tu peux foncer dans les blocs dans un test et marcher dans l'autre.. et après E tout seul parce que euh.. parce qu'il est seul dans l'espace.. et F tout seul parce que c'est le seul où il faut déposer des graines

E – Mm.. alors là tu m'as dit pour B.. ton critère.. enfin pour B et D.. ce serait de.. d'utiliser un si sinon

S – Oui

E – Il y a que dans ce deux grilles-là..

S – Non

E – Que tu utilises un si sinon

S – Non.. ou sinon euh.. ranger B E F ensemble parce qu'ils ont que deux tests.. A et C ensemble parce que euh.. c'est euh .. ils ont le même parcours.. et D tout seul parce que c'est le seul avec des briques

E – Oui alors ce qu'il y a c'est que tes critères ne sont pas.. homogènes c'est à dire que.. pour B E et F tu me parles de nombre de tests.. pour A et C tu me parles de parcours.. et pour D tu me parles d'éléments de décor.. donc si tu me parles de nombre de tests il faut me parler de nombre de tests pour toutes les grilles.. si tu me parles de parcours il faut me parler de parcours pour toutes les grilles.. et si tu me parles d'éléments de décor il faut le faire aussi pour toutes les grilles

S – Bon alors je mettrais A B C ensemble et D E F ensemble parce que dans le D.. enfin D E F c'est des vrais petits robots.. et dans le A le B et le c'est juste un petit rond avec un robot dessus mais.. pas le vrai robot

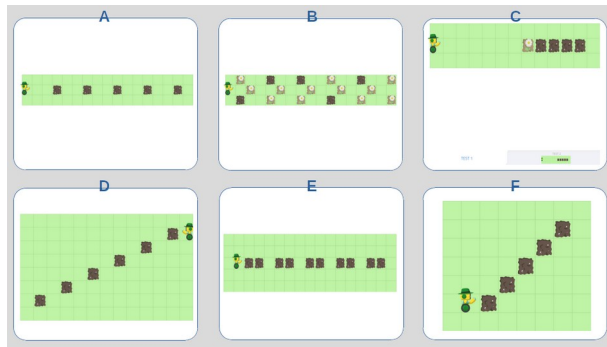
E – Ok.. donc là tu.. tu changes complètement .. de critère qui est là le.. le type de robot.. donc ça ça marche.. là là pour le coup ça marche.. A B et C donc c'est le robot rond.. et D E F c'est.. comme tu l'appelles le vrai robot.. alors je sais pas ce que tu entends par vrai.. qu'est-ce que ça veut dire vrai

S – Enfin c'est pas un vrai robot mais c'est.. un robot avec des bras.. des jambes et une tête.. pas un p'tit rond

E – D'accord.. ok



### Planche 3



E – Lesquelles tu rangerais ensemble ? [Long temps de silence] Tu la vois ?

S – Oui

E – Mm

S – La A la D la E et la F

E – Mm

S – Ensemble si on se trouve face à telle situation il faut faire ça sinon autre chose

E – Oui

S – Et après..

E – Sur quel critère ?

S – Ben parce que c'est une..

E – Parce que c'est..

S – Ben c'est une boucle

E – Je n'avais pas compris le mot boucle.. oui

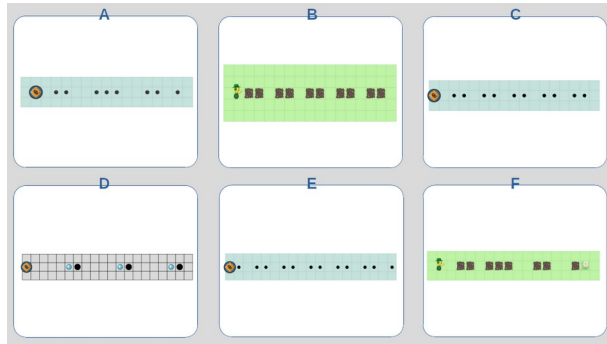
S – Et peut-être la B et la C ensemble

E – Mm

S – Parce que.. c'est en fait heu.. si on se trouve face à telle situation il faut faire ça sinon autre chose en fait

E - Ok

**Planche 6**



E – Ici lesquelles tu rangerais ensemble ?

S – La A et la F

E – Oui

S – Bah parce que.. dans tous les cas.. les deux elles sont aléatoires.. c'est à dire

E – Qu'est-ce que tu veux dire par aléatoire ?

S – Bah c'est à dire que y'a des.. y'a des endroits où faut faire quelque chose et d'autres où faut rien faire mais c'est pas régulier

E – D'accord.. et les autres ?

S – La A et la F.. à part que c'est pas le même concept mais c'est la même disposition

E – Mm.. oui et les autres alors ?

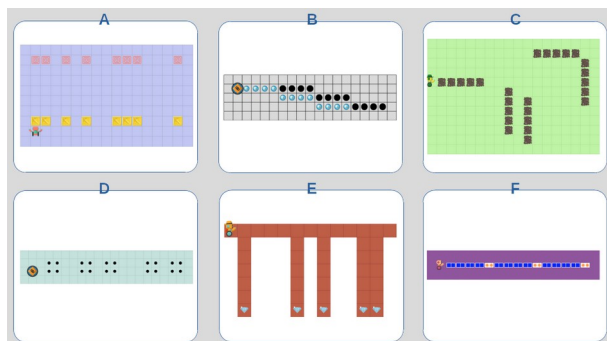
S – Et ben ensemble.. la B la C la D et la E

E – Mm

S – Ben toujours parce que c'est un mouvement répété

E - Ok

**Planche 12**



S – J'mets ensemble la A la D et la E

E – Oui

S – Parce que sur toutes en fait il faut faire le même mouvement mais pas au même moment

E – Mm..

S – Après..

E – Comment tu ferais ça alors ?

S – Ben faire si tel chose faire le mouvement et sinon.. sinon avancer.. après ensemble la B et la F

E – Parce que ?

S – Bah en fait.. il faut mettre une boucle et dedans mettre une autre boucle qui fait plusieurs fois

E – Mm

S – Une action et cette action la refaire plusieurs fois

E – Mm

S – Et la C.. toute seule

E – Oui.. comment tu la décrirais la C ?

S – Ça reste la même action.. ça pourrait se faire par j’sais pas une variable mais heu.. mais il faut se placer à chaque fois à des endroits différents pour le faire

E – Une variable ?

S – Ben oui non ?

E – Décris-moi comment tu utiliserais une variable ?

S – Ben mettre.. enfin une variable.. [inaudible] ben par exemple la variable c’est planter les graines.. et répéter cinq fois avancer planter la graine

E – Et ce serait.. et le bloc il serait de quelle couleur ?

S – Violet

E – Oui donc c’est pas une variable

S – Ah c’est pas ça ?

E – Les variables c’est les blocs orange.. si c’est un bloc violet c’est une fonction

S – Ah désolée

E – C’est pour ça que je comprenais pas et que je t’ai demandé d’expliquer parce que je me disais que t’avais bien compris quelque chose mais que tu n’utilisais pas le bon mot.. donc est-ce que tu peux me réexpliquer comment tu ferais ?

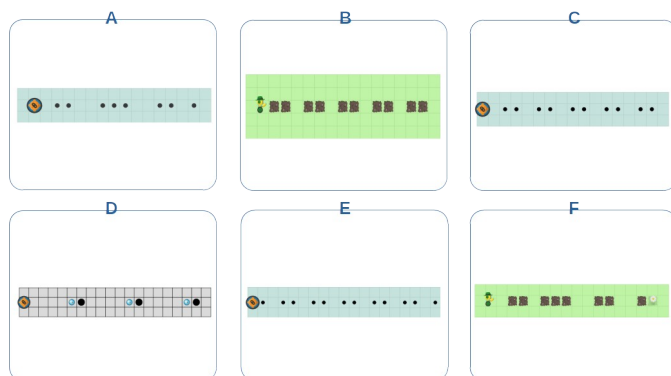
S – Ben faire une fonction qui fait répéter cinq fois avancer semer la graine

E – D’accord

S – Mais pour faire cette fonction il faut d’abord se déplacer.. mais ça il faut le faire dans le bloc mais sans une boucle

E – Mm.. ok.. bon

**Planche 6**



E – Alors.. maintenant ?

S – Euh.. ben ils vont tous ensemble.. euh attends

E – Alors ils vont tous ensemble.. cela veut dire que tu as trouvé un point commun déjà.. quel est le point commun ?

S – Faut utiliser un bloc.. enfin c'est tous des pattern.. non c'est tous sur une même ligne

E – Mm

S – Et euh.. ben faut utiliser un si.. enfin non.. attends.. c'est.. c'est que des pattern qui.. non.. ah.. faut faire un si.. na na na.. euh.. sur case alors.. euh.. par exemple s'il y a un carré de terre sur cette case.. alors aller.. alors planter une fleur

E – D'accord donc tu résoudrais avec un si dans une boucle répéter.. c'est ça que tu veux dire

S – Oui

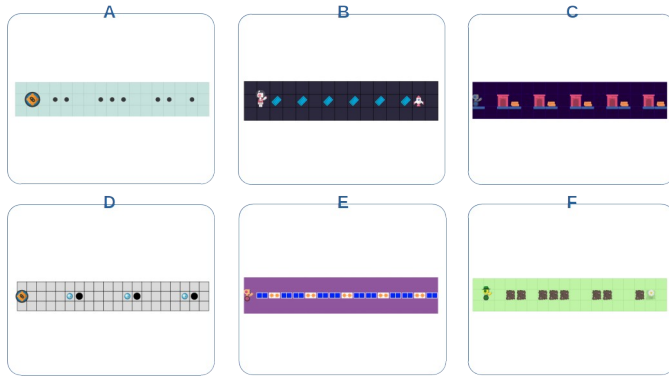
E – Tous ?

S – Euh.. oui.. enfin celui avec les b.. ouais non en fait.. non.. non celui avec les billes c'est un pattern.. euh.. oh non celui qui est avec les billes c'est pareil.. oui ils vont tous au même endroit

E – D'accord.. donc celui avec les billes.. il aurait quelle structure ton programme ?

S – Non enfin non ils sont tous dans le même groupe.. enfin toutes les grilles sont dans la même catégorie

## Planche 7



E – Et là ?

S – Euh..

E – Tu les rangerais comment ?

S – C’est la même chose.. enfin ils vont tous au même endroit

E – Alors c’est quoi leur point commun ?

S – Ben la même que tout à l’heure.. faut utiliser un si dans une boucle

E – Tu résoudrais tous ces problèmes en utilisant un si dans une boucle ? Tu mettrais si quoi alors ?

S – Si euh.. nanana sur la case alors faire l’action qui correspond

E – Alors ce serait quoi cette action pour chacun des cas ?

S – Le premier c’est peindre la case.. le deuxième c’est ramasser l’objet.. le troisième c’est prendre le bois reculer et mettre dans la cheminée

E – Alors c’est si quoi aussi ?

S – Si il y a du bois sur la case

E – Oui mais euh.. et le trou entre le robot et la cheminée ?

S – Ah oui.. si il y a un trou devant construire un pont

E – Mm..

S - Le D c’est s’il y a la bille sur une case ramasser la bille.. avancer la mettre dans le trou

E – Mm

S – Le E c’est quoi ?.. c’est..

E – Alors le E c’est.. le robot doit ramasser seulement les dominos bleus.

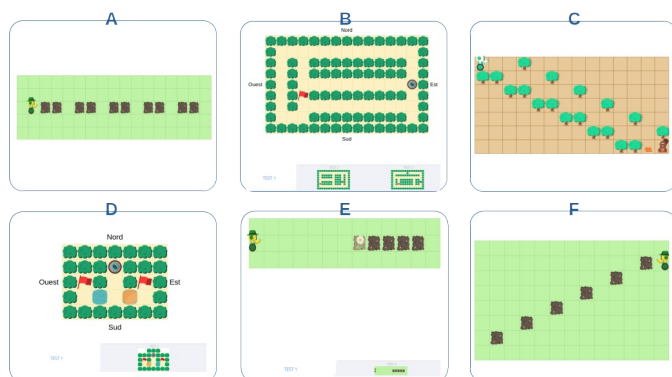
S – Ben si euh.. ben si il est sur une case.. s’il est sur un domino bleu.. ben le ramasser

E – Alors si tu as ce capteur alors oui ça marche

S – Et si.. et le dernier c’est si il est sur une case avec de la terre.. planter une fleur

E - Ok

**Planche 1**



E – Il y a 6 grilles avec des robots.. comme tu as vu.. qu'on appelle A B C D E F.. pour pouvoir dire laquelle c'est très vite.. et ce que je te demande.. c'est lesquelles tu rangerais ensemble.

S – Alors.. euh.. la A

E – Oui

S – La E

E – Mm

S – Et la F

E – Mm.. et qu'est-ce qui te fait les mettre ensemble la A la E et la F ?

S – Bah déjà la couleur ici

E – Oui

S – Et puis les pots.. les pots de terre

E – D'accord

S – Le robot

E – Ok.. d'accord.. est-ce qu'il y en a d'autres que tu mettrais ensemble ?

S – Oui B et D

E – Mm

S – Et C.. bah.. C je le mettrais tout seul

E – C tu le laisserais tout seul.. alors B et D tu les mettrais ensemble pour quelle raison ?

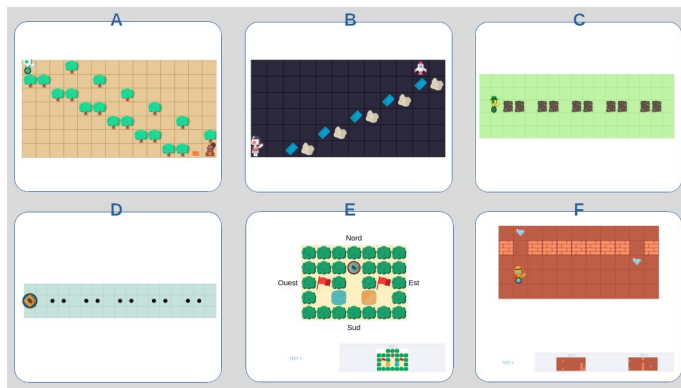
S – Parce que y'a les mêmes buissons.. ils doivent tous les deux avoir un drapeau

E – Mm

S – Y'a nord est ouest sud... nord

E – D'accord

## Planche 2



E - C'est toujours la même question.. c'est lesquelles tu mettrais ensemble ?

S – Aucun

E – Tu peux réfléchir un petit peu.. mais.. quelles grilles tu mettrais.. tu rangerais ensemble ?

S – La D et la E

E – La D et la E oui.. est-ce que tu peux m'expliquer ?

S – Parce que y'a le même bonhomme

E – Y'a le même bonhomme ? Le même robot tu veux dire ?

S – Oui

E – Ok.. Mm

S – Après la.. la A la C.. la A la C et la F

E – Oui

S – Et puis la B toute seule

E – Et la B toute seule.. alors la A la C et la F qu'est-ce qui te fait les mettre ensemble ?

S – Bah.. y'a le.. y'a le robot pareil

E – D'accord c'est le robot

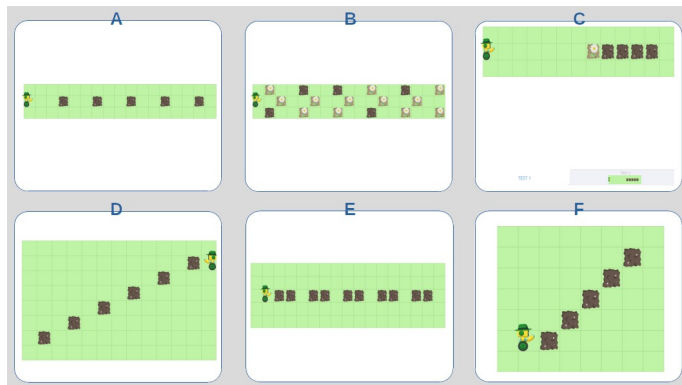
S – Mais pas de la même couleur

E – Sauf qu'il n'est pas de la même couleur mais.. c'est le robot qui est pareil.. et donc le B.. tu la mets toute seule pour quelle raison ?

S - Parce que.. euh.. elle est pas comme la A.. le bonh.. le robot B n'est pas comme le robot A le C et le F

E – D'accord.. donc c'est une histoire de robot

### Planche 3



E – Une autre.. Alors ici.. lesquelles tu rangerais ensemble ?

S – La A la B la C la D la E et la F

E – Alors tu mettrais tout le monde dans le même groupe.. pourquoi tu mettrais tout le monde dans le même groupe ?

S – Mm.. parce qu’il y a le même robot .. et y’a toujours les cases de terre

E – D’accord

S – Et y’a toujours.. y’a toujours ça qu’est vert

E – Et y’a toujours la grille verte.. et si tu devais quand même les séparer.. si tu devais faire au moins deux groupes.. comment tu ferais ?

S – J’aurais la D et la F

E – La D et la F.. alors explique-moi comment tu as réfléchi pour mettre la D et la F ensemble ?

S – Parce que elles sont toutes les deux..

E – Elles sont toutes les deux.. oui.. c’est une bonne idée..

S – Elles sont toutes les deux.. je sais plus le nom..

E – Tu sais plus le nom.. alors tu peux.. si tu sais plus le nom tu peux expliquer

S – Elles sont toutes les deux.. diagonale !

E – Ah c’est ce mot là que tu cherchais.. d’accord.. qu’est-ce qui est en diagonale ?

S – Bah la B la F

E – Oui mais sur les images ?

S – Les pots de terre

E – D’accord.. les pots de terre qui sont en diagonale.. ok.. bon et dans celles qui restent alors ?.. est-ce que..

S – La A et la E

E – Oui

S – Ah non.. la C et la E



E – La C et la E

S – Et la A et la B

E – Alors explique-moi..

S – Heu ben..

E – La C et la E..

S – Parce qu’elles ont une même.. ah bah non j’me suis trompée.. non la A la B et la E ensemble

E – Oui

S – Non.. la A la B et la C ensemble et la E toute seule

E – Oui.. si tu veux mais faut me dire ce qui te fait les mettre ensemble

S – Bah.. elles ont le même nombre de côtés.. trois

E – Elles ont le même nombre de.. ?

S – Côtés

E – De côtés.. de côtés ? qu’est-ce que tu appelles un côté ?

S – Ben là par exemple pour la A.. il y en a trois d’un côté et trois de l’autre.. pour la B pareil.. trois d’un côté trois de l’autre

E – Trois quoi d’un côté ?.. ah la A la B et la C tu veux dire.. trois d’un côté c’est trois cases de côté ?

S – Oui

E – Ok !.. très bien.. donc la grille elle fait trois cases de hauteur.. c’est ça que tu veux dire ?

S – Oui

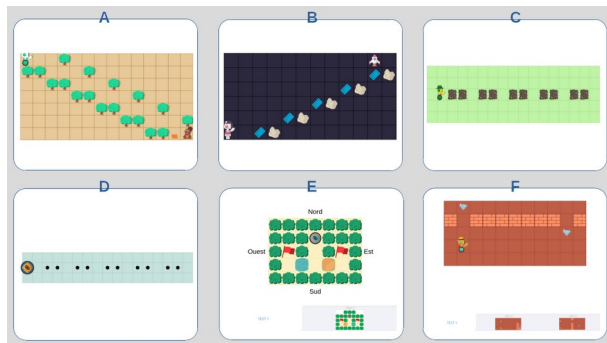
E – D’accord

S – Alors que la E non

E – Et la E non.. elle est plus grande.. mm.. très bien

## Sujet 19 : Louane, CM2

### Planche 2



E – Alors sur celle-là.. comment tu pourrais ranger les grilles de parcours de programmation ?  
[silence] Qu'est-ce qui pourrait te faire mettre certaines grilles ensemble ?

S – Et ben moi je voudrais mettre A B C et D ensemble.. et la E et la F ensemble

E – Oui

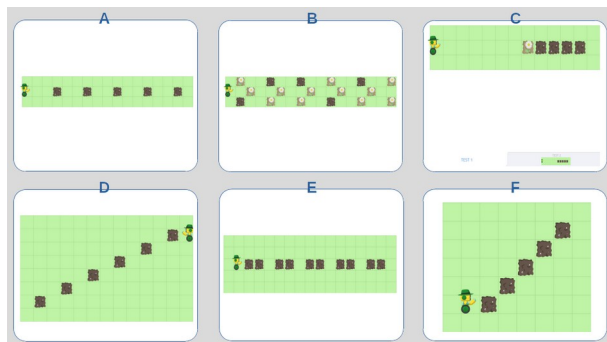
S – Parce que la E et la F y'a plusieurs tests

E – D'accord.. et les autres ?

S – Ben les autres c'est parce qu'il n'y a qu'un.. il n'y a qu'une seule option

E – Voilà.. y'a qu'un seul test.. quand il y a qu'un seul test on ne met pas test un.. ok.. ça aussi tu vois c'est un autre critère.. c'est pas le même.. mais on peut ranger les grilles par nombre de tests

### Planche 3



E – Donc à nouveau c'est toujours la même question.. comment tu rangerais..

S – Euh la A la D la E et la F ensemble.. et la B et la C ensemble

E – oui

S – Parce que la B et la C on voit qu'ils sont déjà commencés

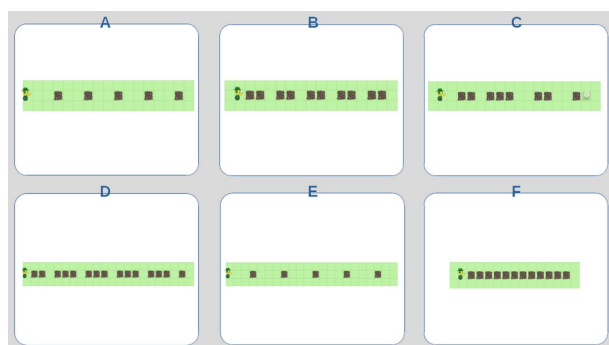
E – Ils sont déjà commencés.. qu'est-ce que tu entends par là ?

S – Ils sont.. y'a déjà des fleurs dessus

E – Ok.. ok donc la B et la C parce qu'il y a déjà des fleurs dessus.. et les autres il n'y en a pas encore.. c'est ça.. c'est ça que tu veux me dire ?

S – Oui

## Planche 8



E – Celle-là ?

[silence]

S – J’vais mettre A B C D et E ensemble.. et la F toute seule.. parce que.. A B C D E.. l’action.. on la fait pas.. j’sais pas.. non en fait non..

E – Tu peux aller jusqu’au bout de ton idée..

S – Non mais ah.. j’vais mettre la.. A.. B.. E et F ensemble

E – Mm

S – Parce qu’on peut.. euh.. ben on peut faire directement en.. comment on dit.. en.. avec la case.. répéter plusieurs fois

E – le bloc oui.. avec le bloc répéter plusieurs fois

S – Oui

E – Oui

S – On peut le faire directement avec ça.. et du coup la.. la C.. la D.. j’vais les mettre.. j’vais les mettre toutes seules

E – Ok.. quand tu.. quand tu dis on peut faire directement avec ça.. qu’est-ce que tu veux dire par directement ?.. ça m’intéresse beaucoup.. qu’est-ce que tu entends par ce mot-là.. directement ?

S – Euh.. c’est.. c’est de commencer par.. par ce bloc

E – Mm

S – Et en fait de.. De garder ce bloc jusqu’à.. jusqu’à la fin du programme.. mais en fait j’vais juste.. ben j’vais juste garder la B E et F.. parce que la A.. si on fait avec ça.. ça fonctionnera pas

E – Mm.. je vois pas bien.. pourquoi la A ça fonctionnera pas ?

S – Parce que la A on peut pas.. on peut pas commencer ce bloc

E – Mm.. tu commencerais par quoi ?

S – Ben je commencerais par.. par avancer

E – Ok.. et après ?

S – Et après j’mettrais le blocs avec mm.. ben j’mettrais le bloc répéter.. plusieurs fois avec les.. ce qu’il faut faire

E – Mm donc pour le A tu.. si je comprends bien.. tu m'arrêtes si c'est pas ça.. tu amènerais le robot sur la première zone de terre.. et ensuite.. c'est à partir de là que tu.. tu commencerais à mettre des choses dans le bloc répéter ?

S – Non

E – Non c'est pas ça ?

S – Non j'veux juste l'amener.. sur la case.. sur la case juste après là où il est.. et après je commence à..

E – D'accord.. ok.. alors si.. si on reprend du coup tu fais quoi comme groupes.. et quelq.. quel ? avec quelle explication ?

S – Alors la B E et F parce que On peut faire.. on peut.. on peut faire..

E – Oui

S – Tout le programme avec heu ? avec.. juste un répéter plusieurs fois avec.. les.. les choses a répéter

E – Mm

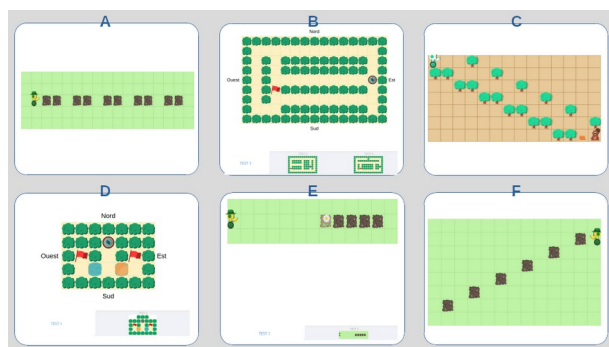
S – Et du coup A.. A C et D.. et ben j'les mettrais tous seuls parce que.. ou sinon.. non j'mettrais A C et D ensemble parce qu'on peut pas.. on peut pas les.. on peut pas faire.. on peut pas faire avec le bloc répéter plusieurs fois

E – Ok ça ça marche

## Relevé des critères de catégorisation

Nous avons commencé un relevé des critères de catégorisation donnés par les sujets. Nous en montrons un extrait pour la planche 1. Nous n'avons pas encore passé en revue les répliques de tous les sujets. Mener ce travail à terme et en faire l'analyse est l'une des perspectives à court terme à l'issue de la recherche doctorale.

### Planche 1



Critère	Catégorisation des sujets
Contexte narratif	<ul style="list-style-type: none"> <li>• AEF/BD/C : « c'est par sujet » (Léa)</li> <li>• tous : « c'est tous en rapport avec la nature » (Lou)</li> </ul>
Aspect de la grille	<ul style="list-style-type: none"> <li>• BCF/AE/D : « c'est à peu près la taille » (Léa)</li> </ul>
Aspect des éléments de la grille	<ul style="list-style-type: none"> <li>• BD : « y'a le labyrinthe » (Ali)</li> <li>• AEF : « c'est de la terre », BCD : « c'est des arbres » (Lou)</li> <li>• DE : « sur une y'a un obstacle et sur l'autre y'en a pas par exemple.. et il faut trouver une manière de l'éviter dans tous les cas » (Louna)</li> </ul>
Aspect du robot	<ul style="list-style-type: none"> <li>• ABCF/ DE : « parce que y'a les robots et D E parce que y'a le p'tit robot là.. le truc bizarre.. le truc en rond là » (Ali)</li> </ul>
Actions du robot	<ul style="list-style-type: none"> <li>• AEF : « y'a les robots planteurs » (Ali)</li> <li>• ACEF : « le robot doit planter une graine sur le.. sur la terre.. .. et j'mettrais aussi le C dedans.. parce qu'il doit ramasser le bois pour le donner au castor.. c'est comme si on ramassait une graine et qu'on la donnait à la terre donc je dis que c'est un peu pareil.. », BD : « le robot doit aller jusque.. jusqu'au drapeau » (Alix)</li> </ul>
Nombre de tests	<ul style="list-style-type: none"> <li>• BDE/ACF : « des tests / c'est pas des tests » (Élana)</li> </ul>
Disposition des éléments sur la grille	<ul style="list-style-type: none"> <li>• AE : « ils sont tous les deux sur une même ligne », CF : « ils sont en escalier » (Alexis)</li> </ul>
Notions algorithmiques explicites	<ul style="list-style-type: none"> <li>• ACF : « patterns qui se répètent », « avec un bloc répéter », (B)DE : « en fonction de s'il y a un obstacle ou pas on va à un certain endroit », « avec un si sinon » (Axel)</li> <li>• AF : « il faut répéter. » (Élana)</li> <li>• ACF : « c'est une répétition du même mouvement », D : « plutôt faire le si sinon » (Louna)</li> <li>• D : « il faudrait utiliser par exemple des si » (Alexis)</li> </ul>