

COLLEGE

EDUCATION, LANGAGES

DOCTORAL

INTERACTIONS, COGNITION

BRETAGNE

CLINIQUE, EXPERTISE

UBO

Université de Bretagne Occidentale



# THÈSE DE DOCTORAT DE

L'UNIVERSITE  
DE BRETAGNE OCCIDENTALE

ÉCOLE DOCTORALE N° 646

*Éducation, Langages, Interactions, Cognition, Clinique, Expertise*

Spécialités : Didactique de l'informatique

Par

**Matthieu BRANTHÔME**

**Apprentissage de la programmation informatique : analyses et ressources pour accompagner la transition collège-lycée**

TOME 2 : ANNEXES

Thèse présentée et soutenue à Rennes, le 11 octobre 2023

Unité de recherche : Centre de Recherche sur l'Éducation, les Apprentissages et la Didactique

## Rapporteurs avant soutenance :

Sébastien GEORGE    Professeur des Universités, Le Mans Université  
Vanda LUENGO       Professeure des Universités, Sorbonne Université

## Composition du Jury :

Présidente :	Patricia MARZIN-JANVIER	Professeure des Universités, Université de Bretagne Occidentale
Examineurs :	Gilles DOWEK	Directeur de recherche INRIA, professeur attaché ENS Paris-Saclay
	Sébastien GEORGE	Professeur des Universités, Le Mans Université
	Jean-Marie GILLIOT	Maître de conférences, IMT Atlantique
	Vanda LUENGO	Professeure des Universités, Sorbonne Université
	Agathe MERCERON	Professeure des Universités, Berliner Hochschule für Technik
Dir. de thèse :	Ghislaine GUEUDET	Professeure des Universités, Université Paris-Saclay
Co-dir. de thèse :	Cédric FLUCKIGER	Professeur des Universités, Université de Lille



# Table des annexes

<b>A</b>	<b>Modélisation praxéologique des documents prescriptifs .....</b>	<b>7</b>
A.1	Cycle 4 – Mathématiques .....	7
A.1.1	Programme d’enseignement MP4 (MEN, 2020).....	7
A.1.2	Ressource d’accompagnement R1M4 (MEN, 2016c) .....	9
A.2	Cycle 4 – Technologie.....	55
A.2.1	Programme d’enseignement PT4 (MEN, 2020).....	55
A.2.2	Ressource d’accompagnement R1T4 (MEN, 2016a) .....	60
A.2.3	Ressource d’accompagnement R2T4 (MEN, 2016b).....	66
A.3	Seconde – Mathématiques .....	77
A.3.1	Programme d’enseignement PMS (MEN, 2019b) .....	77
A.3.2	Ressource d’accompagnement R1MS (MEN, 2019a) .....	86
A.3.3	Ressource d’accompagnement R2MS (MEN, 2017).....	111
A.4	Seconde – Sciences numériques et Technologie .....	115
A.4.1	Programme d’enseignement PSS (MEN, 2019c).....	115
A.4.2	Ressource d’accompagnement R1SS (MEN, 2019d) .....	122
<b>B</b>	<b>Questionnaire enseignants .....</b>	<b>169</b>
B.1	Préambule.....	169
B.2	Groupe A : informations générales .....	170
B.3	Groupe B : discipline et niveau d’enseignement de la programmation informatique.....	173
B.4	Groupe C : contenus et connaissances .....	173
B.5	Groupe D : activité d’enseignement .....	175
B.6	Groupe E : dispositif d’enseignement .....	178
B.7	Groupe F : aspects matériels et logiciels .....	179
B.8	Groupe G : situation sanitaire .....	184
B.9	Remerciements.....	184
<b>C</b>	<b>Questionnaire élèves .....</b>	<b>185</b>
C.1	Connaissances préalables.....	185
C.2	Avis sur l’application .....	185
C.3	Remarques et suggestions.....	186
<b>D</b>	<b>Base de données V1.....</b>	<b>187</b>
D.1	Modèle relationnel.....	187
D.2	Modèle physique.....	187

D.3	Détail des tables.....	187
<b>E</b>	<b>Base données V2.....</b>	<b>189</b>
E.1	Modèle relationnel.....	189
E.2	Modèle physique.....	189
E.3	Détail des tables.....	190
<b>F</b>	<b>Traces d'activités .....</b>	<b>192</b>
F.1	Consultation de contenus.....	192
F.2	Copié-collé de contenus.....	194
F.2.1	Copie de contenu .....	194
F.2.2	Collage de contenu .....	196
F.3	Lancement de programmes.....	198
F.4	Navigation entre les niveaux .....	211
F.4.1	Niveau commencé .....	211
F.4.2	Niveau terminé.....	212
F.4.3	Niveau recommencé.....	213
F.4.4	Niveau repris .....	215
F.4.5	Niveau quitté {V2}.....	216
F.5	Aide .....	218
F.6	Aide {V2} .....	219
F.6.1	Aide demandée.....	220
F.6.2	Aide reçue .....	221
F.6.3	Aide lancée.....	223
F.6.4	Aide relancée .....	224
F.6.5	Aide quittée .....	226
F.7	Modification de la vitesse d'exécution .....	227
<b>G</b>	<b>Analyse à priori des situations .....</b>	<b>230</b>
G.1	Niveau 1 .....	230
G.1.1	Description .....	230
G.1.2	Savoirs en jeu.....	230
G.1.3	Variables didactiques .....	230
G.1.4	Stratégies gagnantes.....	230
G.2	Niveau 2.....	231
G.2.1	Description .....	231
G.2.2	Savoirs en jeu.....	232
G.2.3	Variables didactiques .....	232
G.2.4	Stratégies gagnantes.....	232
G.3	Niveau 3.....	233



G.3.1	Description .....	233
G.3.2	Savoirs en jeu.....	234
G.3.3	Variables didactiques .....	234
G.3.4	Stratégies gagnantes.....	235
G.4	Niveau 4.....	237
G.4.1	Description .....	237
G.4.2	Savoirs en jeu.....	237
G.4.3	Variables didactiques .....	238
G.4.4	Stratégies gagnantes.....	238
G.5	Niveau 5.....	243
G.5.1	Description .....	243
G.5.2	Savoirs en jeu.....	244
G.5.3	Variables didactiques .....	244
G.5.4	Stratégies gagnantes.....	244
G.6	Niveau 6.....	250
G.6.1	Description .....	250
G.6.2	Savoirs en jeu.....	250
G.6.3	Variables didactiques .....	250
G.6.4	Stratégies gagnantes.....	250
G.7	Niveau 7.....	251
G.7.1	Description .....	251
G.7.2	Savoirs en jeu.....	252
G.7.3	Variables didactiques .....	252
G.7.4	Stratégies gagnantes.....	252
G.8	Niveau 8.....	253
G.8.1	Description .....	253
G.8.2	Savoirs en jeu.....	254
G.8.3	Variables didactiques .....	254
G.8.4	Stratégies gagnantes.....	254
<b>H</b>	<b>Guide de démarrage .....</b>	<b>264</b>
<b>I</b>	<b>Mémo Python .....</b>	<b>266</b>
I.1	Notions de base .....	266
I.2	Variable.....	268
I.3	Conditionnelle .....	270
I.4	Boucle for.....	273
I.5	Boucle while.....	276
<b>J</b>	<b>Courriels .....</b>	<b>277</b>

J.1 Invitation participation enquête enseignants .....	277
J.2 Invitation participation expérimentation .....	277
J.3 Courriel de diffusion.....	278
<b>K Certificat de dépôt de code.....</b>	<b>279</b>
<b>L Rétroactions .....</b>	<b>280</b>
L.1 Tâche.....	280
L.2 Rétroactions niveau concept .....	282
L.3 Rétroaction niveau implémentation .....	283
L.4 Rétroaction niveau solution .....	284
<b>M Apprentissage automatique .....</b>	<b>286</b>
M.1 Détail des caractéristiques .....	286
M.2 Histogrammes des caractéristiques .....	289
M.3 Matrices de corrélation des caractéristiques .....	293

# A Modélisation praxéologique des documents prescriptifs

## A.1 Cycle 4 – Mathématiques

### A.1.1 Programme d'enseignement MP4 (MEN, 2020)

#### A.1.1.1 Organisations didactiques

Type de tâches	Technique(s)	Technologie(s)
<b>Enseigner l'informatique conjointement avec le cours de technologie (contenu)</b> <i>« Le programme de mathématiques est structuré selon cinq thèmes : [...] algorithmique et programmation qui entre dans le cadre d'un enseignement de l'informatique dispensé conjointement en mathématiques et en technologie. » p. 126</i>		
<b>Faire travailler les élèves dans une démarche de projet (activité)</b> <i>« Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples » p.136</i>	Faire développer aux élèves quelques programmes simples.	
<b>Ne pas viser une connaissance experte dans un langage particulier (contenu)</b> <i>« Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. » p. 136</i>		

<p><b>Proposer la programmation de jeux (activité)</b></p> <p>« Exemples d'activités possibles : jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe, jeu du cadavre exquis » p. 136</p>		
<p><b>Proposer de résoudre des problèmes à travers le développement de programmes simples (activité)</b></p> <p>« Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier » p.136</p> <p>« Attendus de fin de cycle : Écrire, mettre au point et exécuter un programme simple » p. 136</p> <p>« Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné. » p. 136</p>	<p>Écrire, mettre au point et exécuter un programme.</p>	

## A.1.2 Ressource d'accompagnement R1M4 (MEN, 2016c)

### A.1.2.1 Organisations didactiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Proposer aux élèves une démarche créative (activité)</b></p> <p>« L'apprentissage associe trois types essentiels de démarche : [...] une démarche de création : l'enseignement permet la réalisation de productions collectives (programmes, applications, animations, etc.), au cours desquelles les élèves développent leur autonomie, leur créativité et leur imagination, mais aussi le sens du travail collaboratif ; » p. 2-3</p>	<p>Réalisation de productions collectives</p>	<p>Développe autonomie, créativité, imagination et sens du travail collaboratif</p>
<p><b>Proposer aux élèves une démarche interdisciplinaire (contenu)</b></p> <p>« L'apprentissage associe trois types essentiels de démarche : [...] une démarche interdisciplinaire qui favorise la mise en œuvre de diverses activités de création numérique, en particulier dans le cadre des enseignements complémentaires. » p. 2-3</p>		<p>Favorise les activités de création numérique</p>
<p><b>Faire utiliser aux élèves le logiciel Scratch (artéfact)</b></p> <p>« Le logiciel Scratch offre un environnement d'édition et d'exécution des programmes. Il s'agit d'un logiciel gratuit et disponible sur toutes les plates-formes usuelles, choisi pour sa simplicité, sa fiabilité et sa robustesse dans la mise en œuvre. Il permet de travailler tous les concepts figurant au programme, en particulier la programmation événementielle et la gestion de scripts s'exécutant en parallèle. » p. 2</p>		<p>Scratch est gratuit, multi-plateforme, simple, fiable et robuste.</p> <p>Il permet de mettre en œuvre tous les concepts du programme</p>

<p><b>Faire travailler les élèves en mode débranché (artéfact/dispositif)</b></p> <p>« Les modalités de l'apprentissage correspondant peuvent être variées : <b>travail en mode débranché, c'est-à-dire sans utilisation d'un dispositif informatique</b> » p. 2</p> <p>« <b>Les mouvements relatifs [des lutins] correspondent bien à des instructions de déplacement données à un être humain : elles peuvent être simulées physiquement, par exemple sur un quadrillage dessiné dans la cour du collègue</b> » p. 10</p> <p>« <b>On peut [dans une activité de tracé par déplacement de lutin] commencer par une activité débranchée, où les élèves peuvent se répartir par paire : un élève écrit, à la main, un jeu d'instructions avancer, tourner, relever le stylo ou le descendre en position d'écriture, l'autre exécute en respectant scrupuleusement le « programme » de son camarade. On échange les rôles pour d'autres dessins.</b> » p. 13</p>	<p>Travailler sans utilisation d'un dispositif informatique</p> <p>Simuler physiquement les déplacements des lutins sur un quadrillage dessiné dans la cour.</p> <p>Dans le cadre d'une activité de tracé, mettre les élèves par binômes l'un écrivant le programme à la main, l'autre exécutant scrupuleusement les instructions données par son camarade.</p>	<p>Les mouvement relatifs correspondent à des instructions de déplacements données à des humains</p>
<p><b>Faire travailler les élèves individuellement ou en groupe. (dispositif)</b></p> <p>« Les modalités de l'apprentissage correspondant peuvent être variées : travail [...] <b>individuel ou en groupe</b> » p. 2</p>		
<p><b>Faire travailler les élèves en salle informatique ou en salle de classe (dispositif/artéfact)</b></p> <p>« Les modalités de l'apprentissage correspondant peuvent être variées : travail [...] <b>en salle informatique ou en salle banale</b> » p. 2</p>		
<p><b>Faire travailler les élèves sur tablette ou sur ordinateur (artefact)</b></p>		

<p>« Les modalités de l'apprentissage correspondant peuvent être variées : travail [...] <b>sur tablette ou sur ordinateur</b> » p. 2</p>		
<p><b>Proposer une pédagogie active (activité/dispositif)</b></p> <p>« [l'enseignement de l'informatique] fournit également l'occasion de <b>mettre en place des modalités d'enseignement fondées sur une pédagogie de projet, active et collaborative</b> » p. 2</p> <p>« <b>La mise en activité des élèves. Une séance d'apprentissage de l'algorithmique et de la programmation ne saurait se dérouler sous forme d'un cours descendant, magistral, où les élèves resteraient passifs. Ainsi, il serait par exemple inefficace de demander à des élèves de reproduire durant un tiers de la séance un programme que le professeur aurait expliqué au tableau durant les deux premiers tiers. Si chaque séance doit viser des objectifs de formation clairs et explicites, par exemple découvrir l'utilisation des variables, il convient de réserver l'essentiel du temps à une activité autonome des élèves. Une séance peut commencer par quelques minutes où le professeur expose une situation-problème qui introduit la notion visée : [...]. La séance peut se terminer par la récapitulation des instructions [...], éventuellement sous forme d'une fiche.</b> » p. 3-4</p>	<p>Pas de cours descendant, magistral avec des élèves passifs.</p> <p>Chaque séance vise des objectifs de formation clairs et explicite</p> <p>Le professeur commence par exposer une situation-problème liée à une notion</p> <p>Les élèves travaillent l'essentiel du temps en autonomie</p> <p>La séance se termine par une phase d'institutionnalisation</p>	<p>Les cours magistraux sont inefficaces dans le cadre de l'enseignement de l'algorithmique et de la programmation.</p>
<p><b>Proposer une pédagogie de projet (activité/dispositif)</b></p> <p>« Cet enseignement [...] fournit également l'occasion de mettre en place des <b>modalités d'enseignement fondées sur une pédagogie de projet, active et collaborative.</b> » p. 2</p> <p>« L'apprentissage associe trois types essentiels de démarche : <b>une démarche de projet active et collaborative qui suppose l'établissement d'objectifs partagés et la répartition des tâches, la communication entre élèves contributeurs d'un même projet</b> » p. 2</p>	<p>Établissement d'objectifs partagés</p> <p>Répartition des tâches</p> <p>Communication entre les élèves</p> <p>Le projet est suivi sur quelques séances. Sans être de trop grande envergure.</p>	<p>Le pédagogie de projet permet à chaque élève de construire des connaissances et compétences.</p> <p>Cela développe l'imagination, la créativité, l'autonomie et le sens du travail collaboratif.</p>

« L'enseignement de l'algorithmique et de la programmation se prête particulièrement à la pédagogie de projet. L'objectif est de permettre à chaque élève de construire des connaissances et des compétences tout en développant son imagination et sa créativité dans le cadre d'un projet suivi sur quelques séances. Comme l'indique le programme, cet enseignement doit se traduire par la réalisation de productions collectives dans le cadre d'activités de création numérique, au cours de laquelle les élèves développent leur autonomie, mais aussi le sens du travail collaboratif. Pour autant, il ne s'agit bien sûr pas de monter des projets de trop grande envergure, qui se dérouleraient sur une année entière. Une approche possible est la suivante : au cours d'une première séance, comme il est indiqué dans le paragraphe « La mise en activité des élèves », le professeur propose une activité, dont il a fixé clairement les objectifs de formation et les concepts nouveaux qui devront être installés. Une deuxième séance peut permettre à chaque élève de développer son programme dans les directions qu'il aura lui-même choisies : la séance permet au professeur d'outiller l'élève selon ses besoins pour faire aboutir son projet. Il peut aussi aider l'élève à le redéfinir si la piste sur laquelle il s'est engagé est inadaptée. Une dernière séance peut permettre la finalisation des projets. Le professeur peut aussi proposer une mise en commun des concepts et des techniques utilisés par les uns et les autres. Une mise en ligne de certains projets, sans distinction de niveau d'expertise, peut être envisagée afin de mettre en valeur les travaux des élèves. On peut également valoriser ces travaux par des ateliers-jeux, des expositions... » p. 4

« Il convient, comme on l'a déjà dit, de définir par avance les objectifs de formation correspondants et de prévoir une chronologie en trois phases : très courte introduction de présentation d'un concept ou

Au cours d'une première séance, le professeur propose une activité en exposant clairement les objectifs d'apprentissage

La deuxième séance permet à chaque élève de développer son programme avec l'aide du professeur qui apporte ponctuellement son aide.

La dernière séance permet de finaliser les projets et d'opérer une mise en commun des techniques

Il est possible de valoriser les travaux des élèves : mise en ligne, expositions, ateliers.



<p>d'un problème, activité autonome des élèves comportant éventuellement des défis ou d'autres éléments de différenciation, et, le cas échéant, phase finale d'institutionnalisation des nouveaux concepts, de mise au propre et par écrit des notions rencontrées. » p. 5</p>		
<p><b>Pratiquer la différenciation pédagogique (activité)</b></p> <p>« Mettre en œuvre la différenciation pédagogique [...] Il s'agit d'amener chaque élève à la meilleure maîtrise possible de tous ces attendus [le socle commun], dans le cadre d'un parcours de formation qui prend en compte ses acquis et ses marges de progression. La différenciation pédagogique permet de réaliser cet objectif et ne saurait se réduire à de la remédiation pour les élèves qui ont plus de besoins. Il s'agit d'accompagner chaque élève, en permettant aux meilleurs de construire des méthodes expertes dans la résolution d'un problème algorithmique, de développer des projets personnels ambitieux, d'assimiler des concepts de programmation riches. Il s'agit en même temps de conduire les élèves les plus en difficulté à une maîtrise suffisante des attendus du programme pour valider l'acquisition du socle commun de connaissances, de compétences et de culture. C'est ainsi qu'une séance peut comporter un « défi » permettant aux élèves qui auraient rapidement réalisé l'objectif commun de poursuivre en autonomie leur travail, pendant que le professeur se rend disponible auprès d'autres élèves. [...], le professeur guide l'élève dans la définition de son projet, en respectant ses choix, mais en les adaptant si besoin au degré de maîtrise effectivement atteint, que ce soit pour l'enrichir ou en réduire les objectifs. » p. 5</p>	<p>Prendre en compte les acquis et les marges de progression des élèves.</p> <p>Les meilleurs élèves peuvent utiliser des méthodes expertes et mener des projets ambitieux. Les élèves les plus faibles doivent atteindre la maîtrise du socle commun.</p> <p>Ne pas se limiter à de la remédiation pour les élèves en difficulté.</p> <p>Un séance peut comporter un défi au-delà de l'objectif commun pour permettre aux meilleurs élèves d'approfondir leurs connaissances.</p> <p>Le professeur se rend disponible pour aider les élèves à adapter leurs choix à leur degré de maîtrise.</p>	<p>La différenciation pédagogique permet d'amener chaque élève à la maîtrise du socle commun.</p>

<p>« Cette façon de procéder [proposer aux meilleurs d'utiliser une boucle pour éviter la répétition de code] : donner un même objectif à tous et proposer selon l'avancement de chacun des compléments de recherche, peut être utilisée régulièrement, tout au long du cycle.» p. 12</p>	<p>En cas de répétition de code, proposer aux meilleurs élèves d'utiliser une boucle pour éviter la répétition</p>	<p>Permet de donner un même objectif à tous, et aux élèves les plus avancés de continuer à chercher.</p>
<p>« Cette activité [proposer d'implémenter un jeu complexe] se prête facilement à un découpage, en ajoutant au fur et à mesure un personnage et des interactions, ce qui permet une pédagogie différenciée. »p. 15</p>	<p>Proposer la mise en œuvre d'un jeu complexe se prêtant facilement aux découpages</p>	<p>Permet de mettre en œuvre une pédagogie différenciée.</p>
<p><b>Laisser les élèves travailler en autonomie (dispositif)</b></p> <p>« L'apprentissage de l'algorithmique et de la programmation bénéficie d'une boucle essai-erreur qui est très courte : en cliquant sur un bloc ou un script on peut observer immédiatement l'effet sur les lutins, les tracés du stylo... D'autre part, aucun jugement de valeur n'est porté sur l'élève ou ses performances, la machine se contente d'effectuer les commandes et c'est l'élève lui-même qui constate s'il s'est trompé ou pas, et qui est invité à essayer une autre solution en cas d'échec. De plus, l'obtention d'une solution ne signifie pas la fin du processus de création, qui peut se poursuivre, pour en améliorer la qualité ou la performance. Pour le professeur, cela signifie qu'il accepte pour ses élèves la nécessité de phases de tâtonnement et d'essais-erreurs, durant lesquelles il intervient le moins possible. En particulier, il est recommandé de résister à l'envie de taper sur le clavier ou de manipuler la souris à la place de l'élève, et de se contenter de répondre oralement aux questions de l'élève, de lui rappeler éventuellement tel ou tel script déjà écrit qui pourrait l'inspirer ou encore de lui suggérer de nouvelles pistes pour résoudre</p>	<p>Laisser les élèves travailler à l'aide de boucles courtes de rétroaction sous forme d'essais-erreurs.</p> <p>Intervenir le moins possible pendant les phases de tâtonnement.</p> <p>Ne pas manipuler l'ordinateur à la place de l'élève.</p> <p>Répondre oralement aux questions en évoquant éventuellement la mémoire didactique ou en donnant de nouvelles pistes.</p>	<p>Le logiciel Scratch ne porte aucun jugement de valeur et se contente d'effectuer les commandes.</p>

<p>la difficulté qu'il rencontre, mais sans intervenir directement sur la machine de l'élève.» p. 5</p>		
<p><b>Favoriser une méthode de programmation souple par essais-erreurs (activité/artéfact)</b></p> <p>« Le programme définit comme attendu de fin de cycle : « <b>Écrire, mettre au point et exécuter un programme simple.</b> » Il met l'accent sur l'analyse d'un problème en vue de concevoir un algorithme, mais aussi sur la phase de programmation effective, indissociable d'une étape de mise au point. Cependant, il ne faut pas nécessairement insister sur une chronologie précise, ces trois phases avançant souvent en parallèle : la mise au point comprend la phase d'essais-erreurs où l'élève construit petit à petit un programme qui répond au problème qu'il s'est posé. De même, au moment de la programmation effective l'élève peut se retrouver confronté à une question algorithmique qu'il n'avait pas prise en compte dès le départ. L'attendu de fin de cycle évoquant un programme simple, il n'est pas question de formaliser d'aucune manière la distinction algorithme/programme. En particulier, il ne s'agit pas de définir un langage de description d'algorithme (ni pseudo-langage ni organigramme) même si son utilisation peut être envisagée, ponctuellement, mais sans souci de normalisation. En revanche, il ne faut pas négliger la phase de mise au point : le professeur peut évoquer, si la situation s'y prête, l'utilisation d'un jeu de test. » p. 3</p>	<p>Ne pas rester sur un schéma linéaire : analyse du problème, conception d'un algorithme puis programmation effective. Mener ces phases en parallèle, en construisant un programme par essais-erreurs pour répondre au problème.</p>	
	<p>Pas de formalisation systématique des algorithmes en pseudo-langage ou organigramme, pas de distinction algorithme/programme.</p>	
	<p>Faire utiliser à l'occasion des jeux de tests</p>	<p>Cela permet d'insister sur la phase de mise au point</p>
<p><b>Faire travailler les élèves de façon collaborative (dispositif)</b></p>	<p>Faire comparer un programme avec celui d'autres élèves</p>	<p>Permet une première approche de la documentation d'un programme</p>

<p>« Il [l'enseignement de l'informatique] fournit également l'occasion de mettre en place des modalités d'enseignement fondées sur une pédagogie de projet, active et collaborative » p. 2</p> <p>« Les élèves sont amenés à comparer leurs solutions algorithmiques et leurs programmes, ce qui les conduit à une première approche, informelle, de la documentation d'un programme (commentaires et spécification). Le partage de programmes entre classes ou entre collègues donne l'occasion de réutiliser en tout ou en partie les programmes écrits par d'autres, illustre la notion de réseau et introduit les questions liées à son usage » p. 3</p>	<p>Faire partager ses programmes avec d'autres élèves dans la classe ou dans d'autres collèges.</p>	<p>Illustre la notion de réseau et les questions liées à son usage</p>
<p><b>Fournir ponctuellement aux élèves des programmes à compléter (activité)</b></p> <p>« Pour favoriser le travail autonome des élèves, il est possible de préparer la séance en proposant aux élèves, via le réseau ou une clé USB, une « trame » de programme, qu'ils seront amenés à compléter. Cette trame doit être limitée à ce que les élèves n'auraient pas le temps de préparer, ou aux objets qu'ils ne sont pas encore en mesure de programmer eux-mêmes. Il serait par exemple malavisé de faire travailler les élèves durant toute une séance à la création d'un arrière-plan : le professeur peut demander que ce travail soit effectué à la maison, ou proposer un arrière-plan qu'il met à disposition des élèves en début de séance. On verra plus loin tout l'intérêt qu'il peut y avoir que les élèves disposent d'un bloc utilisateur « reset », créé par le professeur, qui permette de réinitialiser systématiquement la configuration avant un travail sur le déplacement et les tracés dont on prévoit qu'il nécessitera de nombreuses tentatives avant d'aboutir. En revanche, il n'est pas souhaitable de systématiquement proposer aux élèves un</p>	<p>Proposer une trame de programme à compléter</p> <p>Proposer un arrière-plan ou proposer de le préparer à la maison</p> <p>Proposer un bloc « reset » qui réinitialise la configuration dans les exercices de tracé.</p> <p>Créer et fournir aux élèves un bloc « reset » permettant de réinitialiser la position du lutin à chaque exécution.</p>	<p>Favorise le travail en autonomie.</p> <p>La trame permet dans certains cas de gagner du temps ou de faire utiliser aux élèves des notions qu'ils ne sont pas capable de maîtriser.</p> <p>Permet de gagner du temps.</p> <p>Scratch ne réinitialise pas la position du lutin entre chaque exécution des scripts</p>

<p><b>programme incomplet</b> qu'ils seraient chargés de compléter, d'améliorer ou de corriger. En effet, la phase de mise au point qui accompagne toujours l'écriture d'un programme développe précisément ces compétences, et il ne faut pas perdre de vue l'objectif d'autonomie des élèves. Ceux-ci doivent être amenés, le plus souvent possible, à écrire eux-mêmes leurs propres programmes en développant leur inventivité, leur imagination et leur créativité. » p. 5-6</p> <p>« Une difficulté se rencontre fréquemment lors de la mise au point d'un programme de dessin : quand on clique sur le drapeau vert pour lancer un script, aucune initialisation particulière n'est effectuée par Scratch, qui reprend le lutin dans l'état courant (position et direction). Ce peut être une bonne pratique que le professeur crée un nouveau bloc-utilisateur « reset » que les élèves pourront systématiquement insérer en tête de leurs scripts, même sans entrer dans les détails de la conception d'un tel bloc lors de ses premières utilisations. » p. 10</p>	<p>Ne pas proposer systématiquement un programme incomplet.</p> <p>Les élèves doivent généralement écrire intégralement leurs programmes</p>	<p>La mise au point des programmes doit être travaillée.</p> <p>Les élèves doivent être autonomes</p> <p>Cela développe l'inventivité, l'imagination et la créativité.</p>
<p><b>Présenter les variables à l'aide de la métaphore de la « boîte » (contenu)</b></p> <p>« Il existe une façon à la fois claire et rigoureuse de présenter les variables informatiques. On dira qu'une variable <math>x</math> est une étiquette collée sur une boîte qui peut contenir différentes valeurs. Il est important de faire comprendre que le contenu de chaque boîte varie au cours de l'exécution d'un programme, ce qui n'est pas le cas de la variable mathématique (la variable <math>x</math> pour une fonction ne varie pas au cours du calcul, ou encore la lettre <math>x</math> associée à une grandeur dans un problème reste attachée à cette grandeur pendant les calculs) [...] Pour autant, il est important d'expliquer correctement la notion de</p>	<p>La variable est présentée comme une étiquette collée sur une boîte qui contient plusieurs valeurs.</p> <p>Il est possible de dessiner ces boîtes au tableau et de faire évoluer leur contenu au cours de l'exécution d'un programme.</p>	<p>Cela permet de bien comprendre que la valeur des variables varie au cours du temps, à la différence des mathématiques.</p>

<p>variable, en illustrant le propos par exemple avec des boîtes dessinées au tableau. On fait évoluer le contenu en simulant l'exécution d'un programme. On fait ainsi explicitement la différence avec l'utilisation des variables en mathématiques.» p. 7</p>		
<p><b>Présenter l'exécution d'un programme comme une suite d'état des boîtes (contenu)</b></p> <p>« Pour expliquer le fonctionnement d'un programme, il est donc nécessaire de parler d'états ou de configurations successifs, la description d'une configuration précisant en particulier les étiquettes et les contenus de chaque boîte de mémoire. Ainsi est introduite une notion de temporalité, par la suite des états (ou configurations) successifs. [...] La notion d'états (ou configurations) successifs peut être introduite, pour expliquer le comportement d'un programme. » p. 8</p>	<p>On présente l'exécution séquentiel d'un programme comme une suite de configuration des boîtes et de leurs contenus</p>	
<p><b>Introduire les boucles lorsque les élèves en ressentent le besoin (contenu/activité)</b></p> <p>« L'objectif n'est pas que les élèves utilisent nécessairement un bloc de répétition avant qu'ils en ressentent eux-mêmes l'utilité : la différenciation pédagogique joue ici pleinement son rôle, et le professeur propose ces blocs aux seuls élèves qui en font la demande, et ne procède à leur institutionnalisation que dans une séance ultérieure » p. 13</p>	<p>Proposer des situations permettant l'utilisation de boucles</p> <p>L'élève doit ressentir l'utilité d'un bloc de répétition.</p> <p>Proposer le bloc aux seuls élèves qui en font la demande.</p> <p>Procéder à l'institutionnalisation dans un second temps.</p>	<p>Permet la différenciation pédagogique</p>
<p><b>Introduire la notion de variable lors d'une activité de tracé ou le développement d'un jeu (contenu/activité)</b></p>	<p>Proposer dans le cadre des activités de tracé, une</p>	<p>Permet à chaque élève d'avancer à son rythme (différenciation pédagogique)</p>

<p>« On peut dans une séance suivante, introduire la notion de variable et proposer ainsi de reprendre la figure des 4 bâtons, et en proposer d'autres, comme les suivantes : là encore, chaque élève doit pouvoir avancer à son rythme, et ne devrait pas être contraint à utiliser trop tôt des procédures expertes avant d'avoir maîtrisé les commandes élémentaires de mouvement et de tracé. » p. 14</p>	<p>complexification rendant possible l'usage des variables</p>	
<p>« On rencontre [dans une activité de développement de jeu] également naturellement la notion de variable [pour la gestion du score][...] On peut d'ores et déjà ajouter une variable [...] qui permet de gérer un score.» p. 15</p>	<p>Proposer une gestion du score lors de la programmation d'un jeu. Proposer de gestion temporelle à l'aide d'une variable</p>	<p>Permet d'amener le notion de variable de façon naturel</p>
<p>« Cela permet d'introduire le chronomètre de Scratch, mais également une variable qui conservera l'heure où le poisson a été attrapé » p. 15</p>	<p>Proposer de distinguer la portée des variables : locale ou globale dans le cadre d'un jeu</p>	<p>Permet une première introduction aux concepts de la programmation-objet</p>
<p>« On peut d'ores et déjà ajouter une variable globale qui permet de gérer un score [...] Cela permet d'introduire le chronomètre de Scratch, mais également une variable qui conservera l'heure où le poisson a été attrapé : le script du poisson étant le seul qui ait besoin de cette variable, on la définira comme variable privée du lutin, à la différence du score que plusieurs lutins ont besoin de modifier ou de consulter, qui doit donc être une variable publique (ou globale). Cette distinction est une première introduction aux concepts de la programmation-objet qui n'a évidemment pas à être formalisée. » p 15-16</p>		
<p><b>Introduire le parallélisme (contenu)</b></p>	<p>Ne pas formaliser la notion de parallélisme.</p>	



« On introduit ici la notion de parallélisme, c'est-à-dire d'exécution simultanée de deux scripts différents. Le terme « parallélisme » n'est d'ailleurs pas une connaissance attendue des élèves, c'est l'utilisation de scripts parallèles qui figure seule parmi les objectifs de formation. Bien sûr, tout développement théorique sur le parallélisme est à proscrire. En l'occurrence, le résultat est assez imprévisible, car les deux scripts devraient se dérouler à peu près à la même vitesse, et on ne peut deviner ce qu'annoncera le chat... Il est sans doute préférable de n'évoquer le parallélisme que dans des situations où les différents scripts amenés à se dérouler en parallèle sont déclenchés par des événements différents. [Exemple voir ci-contre] On pourrait bien sûr n'écrire qu'un script, la boucle « répéter indéfiniment » comprenant les deux conditionnelles de façon consécutive : ce serait cependant un peu cacher le fonctionnement de l'algorithme, en compliquer la lecture et la compréhension. C'est ainsi un exemple typique d'utilisation pertinente de la notion de parallélisme » p. 8-9

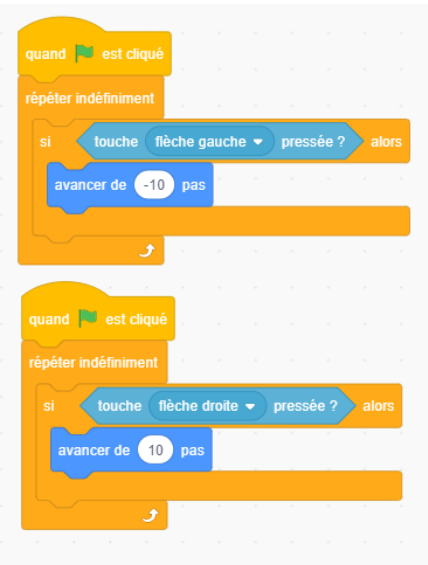
« L'apprentissage de la programmation en Scratch peut se développer [...] suivant une progression liée à des concepts de programmation [...] : l'initiation à la programmation parallèle : déclenchement par le même événement de deux ou plusieurs séquences d'instructions ; » p. 3

« il peut être utile de réfléchir au nombre de scripts qui se déroulent en parallèle à un moment donné » p 16

Utiliser le parallélisme pour des scripts déclenchés par des événements différents

Permet d'éviter les accès concurrents aux mêmes variables qui rendent imprévisibles les résultats

Exemple « pertinent d'utilisation » du parallélisme :



Les scripts sont déclenchés par deux événements différents (flèche gauche et flèche droite). Le fait de séparer les conditionnelles dans deux boucles « répéter indéfiniment » aide à la compréhension de l'algorithme

Déclencher plusieurs blocs d'instructions par un même événement.

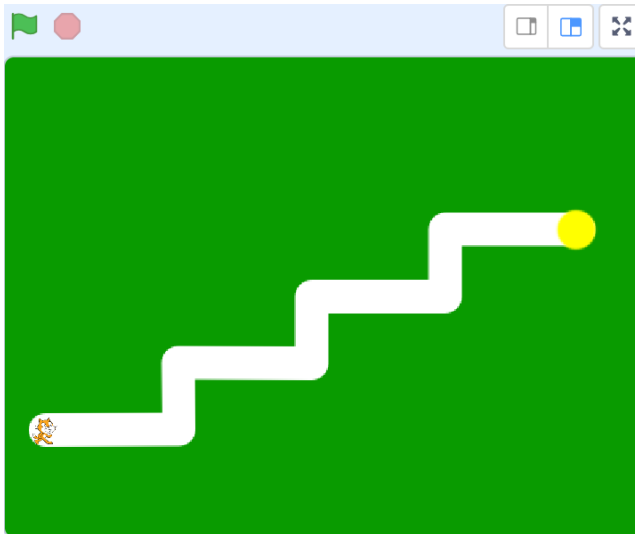
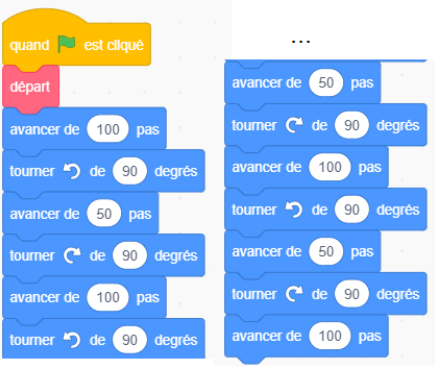
Réfléchir au nombre de scripts qui s'exécutent en parallèle à un instant.



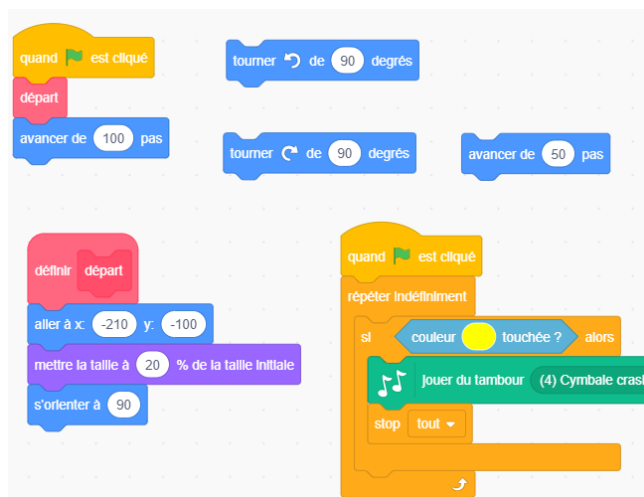
<p><b>Initier les élèves à la programmation événementielle (contenu)</b></p> <p>« L'apprentissage de la programmation en Scratch peut se développer [...] suivant une progression liée à des concepts de programmation [...] : <b>la programmation événementielle : conception de séquences d'instructions déclenchées par un événement (appui de touche, son reçu par le microphone, motif « touché » par un personnage, etc.)</b> » p. 3</p> <p>« <b>On peut commencer avec le seul lutin habituel (le chat). On demande d'abord aux élèves de gérer son déplacement à l'aide des quatre flèches du clavier, ce qui relève de la programmation événementielle.</b> » p. 15</p> <p>« <b>Il peut être intéressant de comparer la gestion directe d'un événement comme l'appui d'une touche avec celle du contact détecté par la valeur d'un capteur</b> » p. 15</p>	<p>Faire déclencher des blocs d'instructions par un événement (touche frappée, son reçu, etc.)</p> <p>Demander aux élèves de programmer le pilotage des déplacements d'un lutin au claviers</p> <p>Comparer les déclencheurs d'événements : appui sur un touche, valeur d'un capteur</p>	<p>Relève de la programmation événementielle</p>
<p><b>Initier les élèves à la programmation-objet (contenu)</b></p> <p>« Le professeur pourra envisager une <b>initiation de ses élèves à certains concepts simples de la programmation-objet par la programmation des « lutins » de Scratch, c'est-à-dire les objets tels que les différents personnages, [...], mais aussi tout simplement en associant des scripts distincts aux différents lutins d'une même activité</b> » p. 3</p> <p>« <b>La distinction entre les scripts de chaque lutin est également un élément qui mérite d'être souligné</b> » p. 15</p>	<p>Faire programmer les « lutins » en tant qu'objets.</p> <p>Distinguer les scripts portés par des lutins différents.</p>	

<p><b>Travailler la création de blocs personnels (contenu)</b></p> <p>« Blocs créés par les utilisateurs. La création d'un bloc-utilisateur permet d'aborder la notion de typage des arguments, puisque Scratch autorise des blocs à plusieurs arguments, chacun d'entre eux étant d'un type bien défini, à choisir parmi nombre, chaîne de caractères ou booléen. Une attention particulière sera portée au nommage d'un bloc-utilisateur et de ses arguments, Scratch permettant d'introduire du texte entre plusieurs arguments [...] La création d'un bloc-utilisateur peut être envisagée dans le cas de la réutilisation fréquente d'un script. Elle permet donc également d'aborder les notions de réutilisation du code, de partage de programme et de spécification : pour partager un programme, il est important de le documenter. Cela passe par un nommage pertinent et parlant et le typage explicite des arguments, comme on l'a déjà dit, mais aussi par la description (en français) de l'action réalisée par le bloc-utilisateur et la déclaration des hypothèses qu'on fait éventuellement sur les arguments. Là encore, l'objectif n'est pas de formaliser des protocoles de documentation, mais de sensibiliser à son importance, à l'occasion d'un partage effectif de « morceaux de code » entre élèves. » p. 11</p>	<p>Faire créer des blocs par les élèves en cas de réutilisation fréquente de code.</p>	<p>Permet d'aborder la notion de typage des arguments (typage demandé à la création des arguments).</p> <p>Permet d'aborder les notions de réutilisation de code et de partage de programme.</p>
	<p>Faire nommer les blocs et introduire du texte entre les arguments de façon pertinente de manière à expliciter l'action du bloc.</p>	<p>Permet de sensibiliser la élèves à l'importance de la documentation du code dans l'optique de son utilisation par un tiers.</p>

## A.1.2.2 Organisations informatiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Projet 1 : déplacer un lutin selon un parcours prédéfini</b></p> <p>Il s'agit de déplacer le lutin dans un « labyrinthe » en combinant des déplacements unitaires relatifs (avancer/tourner) afin de l'amener à la cible jaune.</p>  <p>Le professeur fournit : un squelette de programme qui contient : l'arrière-plan / un bloc départ permettant de réinitialiser la position du lutin / un script détectant le succès / le début du script à compléter et les instructions élémentaire (duplicables par clic-droit) permettant de déplacer le lutin. Il s'agit donc pour l'élève de combiner les</p>	<p><b>Technique 1 :</b> enchaîner les déplacements unitaires.</p> <p>« [...] <b>combiner des déplacements (« avancer de » et « tourner ») pour amener le chat à la cible jaune</b> » p. 11</p>  <p><b>Ingrédients élémentaires :</b></p> <p>Attendre la survenue d'un évènement</p> <p>Utiliser un bloc construit sans entrée (utiliser une fonction sans paramètre sans retour)</p>	<p>Permet de prendre en main le logiciel scratch (vocabulaire, fonctionnement des blocs, exécution et enregistrement des scripts)</p>

instructions en piochant parmi celles proposées à la suite de l'appel du bloc « départ' »



« Cet exemple de première séance a comme **objectif principal la prise en main par les élèves du logiciel**. Il s'agit :

- de **commencer à utiliser le vocabulaire de Scratch** pour savoir nommer les objets de l'interface;
- de **comprendre le fonctionnement des blocs**, leur « aimantation », leur duplication par un clic droit dans la fenêtre de script ;
- de **savoir exécuter et mettre au point un script, de déclencher son exécution par un clic sur le drapeau vert** ;
- de **enregistrer son programme**. » p. 11

Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x13

Technique 2 : utiliser une boucle pour répéter 3 fois un mouvement



Ingrédients élémentaires :

Attendre la survenue d'un évènement

Utiliser un bloc construit sans entrée (utiliser une fonction sans paramètre sans retour)

Utiliser une boucle for

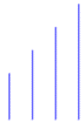
Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x4

La boucle permet d'éviter les répétitions de code.

« Une autre idée consiste à **simplifier leur script à l'aide d'un bloc de répétition** » p. 12



**Étape 2 : Faire reproduire au lutin un tracé donné plus complexe**



Le bloc reprendre au début est fourni

Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x15

Technique 1 : tracer chaque bâton tour à tour.


Ingrédients élémentaires :

Attendre la survenue d'un évènement

Utiliser un bloc construit sans entrée (utiliser une fonction sans paramètre sans retour)

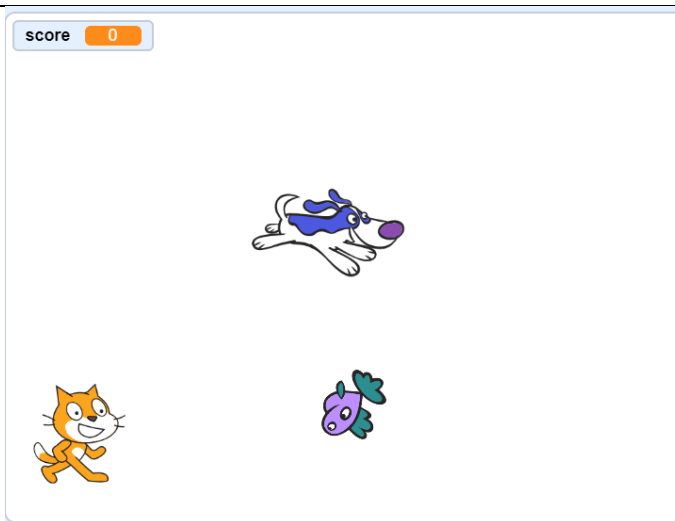
Solution simple.

« [cette figure] peut être obtenue de nombreuses façons différentes. Par exemple, [...] on pourra simplement utiliser le script ci-dessous. » p. 14

	<p>Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour) x7</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x20</p>	
	<p><i>Technique 2</i> : utiliser une boucle et une variable pour la taille des bâtons</p>  <p>The image shows a Scratch script starting with a 'when clicked' event. It sets a variable 'distance' to 50 and enters a 'repeat 4 times' loop. Inside the loop, it sets the pen to 'writing', turns 90 degrees, moves 'distance' steps, turns 180 degrees, lifts the pen, moves 'distance' steps, turns 90 degrees, moves 25 steps, and adds 25 to 'distance'.</p> <p><u>Ingrédients élémentaires</u> :</p>	<p>La boucle permet de simplifier le script en évitant les répétitions.</p> <p>« Des élèves peuvent chercher à utiliser des répétitions pour simplifier leur script, se confrontant à une difficulté : la taille variable des 4 bâtons à dessiner. » p. 14</p>

	<p>Attendre la survenue d'un évènement</p> <p>Utiliser un bloc construit sans entrée (utiliser une fonction sans paramètre sans retour)</p> <p>Utiliser une boucle for</p> <p>Manipuler une variable de type entier</p> <p>Modifier la valeur d'une variable (affectation)</p> <p>Utiliser la valeur d'une variable x2</p> <p>Modifier la valeur d'une variable (ajout)</p> <p>Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour) x2</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x6</p>	
<p><b>Projet 3 : programmer un jeu permettant de diriger un personnage en interaction aux d'autres éléments</b></p>		<p>Permet un découpage facile (favorise la pédagogie différenciée) et de travailler la gestion des déplacements, les interactions entre lutins, la</p>





Le chat est dirigeable à l'aide des touches du clavier. Le chien réalise un mouvement uniforme (droite-gauche). Si le chat le touche, il perd un point. Le poisson réalise un mouvement aléatoire. Si le chat le touche, il gagne un point. Il faut atteindre un score supérieur à 100 pour gagner, à l'inverse, un score négatif entraîne une défaite.

« Cette activité se prête facilement à un découpage, en ajoutant au fur et à mesure un personnage et des interactions, ce qui permet une pédagogie différenciée. Elle permet de revoir la gestion des déplacements et introduit l'utilisation conjointe de plusieurs personnages, ainsi que leur interaction. On rencontre également naturellement la notion de variable, et l'utilisation du chronomètre intégré à Scratch. » p. 15

notion de variable et l'utilisation du chronomètre.

**Étape 1 : Programmer le contrôle des déplacements d'un lutin (chat) à l'aide des touches du clavier.**

Script du chat

« On peut commencer avec le seul lutin habituel (le chat). On demande d'abord aux élèves de gérer son déplacement à l'aide des quatre flèches du clavier, ce qui relève de la programmation événementielle. » p. 15



Ingrédients élémentaires :

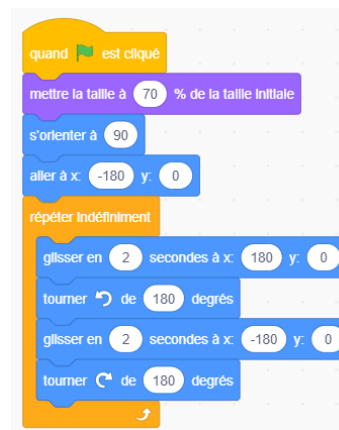
Attendre la survenue d'un événement x4

Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x4

**Étape 2 : Ajouter un lutin (chien) aux déplacements uniformes**

« On ajoute un nouveau lutin, un chien, qui doit effectuer indéfiniment des allers-retours du bord gauche au bord droit de l'écran, ce qui est facilement réalisé à l'aide d'une répétition infinie et des blocs « tourner » et « glisser » » p. 15


*Script du chien*



Ingrédients élémentaires :

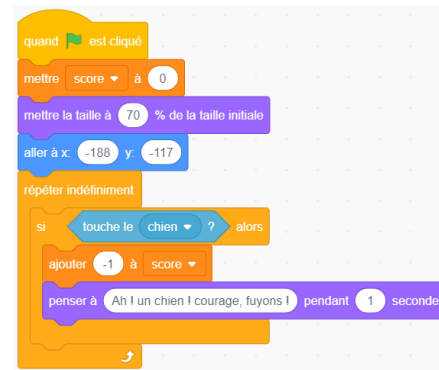
Attendre la survenue d'un événement

Utiliser une boucle non-for

	<p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x7</p>	
<p><b>Étape 3 : Gérer l'interaction entre les lutins chien et chat</b></p> <p>« On souhaite alors que chaque lutin réagisse au moment où il rencontre l'autre : le chien pourra simplement dire bonjour et le chat exprimer sa frayeur. La « rencontre » est détectée par le capteur « touché ? » et il faudra la tester au sein d'une répétition infinie » p. 15</p> <p>« On peut d'ores et déjà ajouter une variable globale qui permet de gérer un score, chaque rencontre du chat et du chien faisant baisser le score. » p. 15</p>	<p><b>Variables</b></p> <p><input checked="" type="checkbox"/> score</p> <p><b>Ingrédients élémentaires :</b></p> <p>Déclarer une variable globale</p> <p><i>Script du chien</i></p>  <p><b>Ingrédients élémentaires :</b></p> <p>Attendre la survenue d'un évènement</p> <p>Utiliser une boucle non for</p> <p>Utiliser une structure conditionnelle à une branche</p> <p>Tester la survenue un évènement</p>	

Agir sur un lutin avec paramètre  
(utiliser une méthode d'un objet  
avec paramètre sans retour)

*Script du chat*



Ingrédients élémentaires :

Attendre la survenue d'un événement

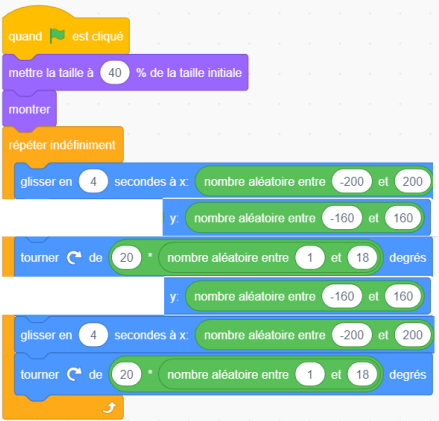
Manipuler une variable de type entier

Modifier la valeur d'une variable (affectation)

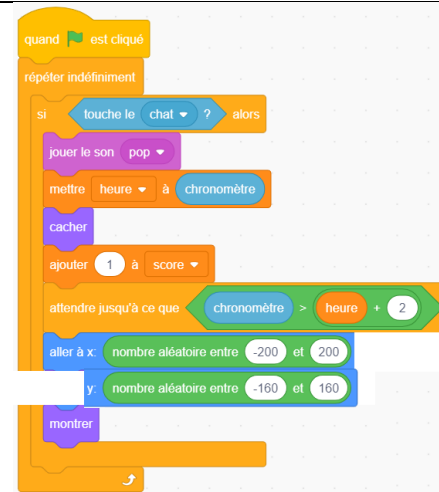
Utiliser une boucle non for

Utiliser une structure conditionnelle à une branche

Tester la survenue d'un événement

	<p>Modifier la valeur d'une variable (ajout)</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x3</p>	
<p><b>Étape 4 : Ajouter un lutin (poisson) aux déplacements aléatoires</b></p> <p>« On propose alors d'ajouter un troisième personnage, un poisson. Il s'agit de programmer un déplacement aléatoire dans le plan de ce lutin. » p. 15</p>	<p><i>Script du poisson</i></p>  <p><u>Ingrédients élémentaires :</u></p> <p>Attendre la survenue d'un évènement</p> <p>Utiliser une boucle non-for</p> <p>Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour)</p>	

	<p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour)</p> <p>Utiliser un générateur de nombres aléatoires (utiliser une fonction avec paramètre avec retour) x6</p> <p>Multiplier des nombres x2</p>	
<p><b>Étape 5 : Gérer l'interaction entre les lutins poisson et chat</b></p> <p>« On gère l'interaction chat-poisson : à chaque contact, on incrémente le score, et on souhaite faire disparaître le poisson (le chat l'a mangé) et le faire réapparaître à un endroit aléatoire après un délai fixé. La difficulté tient au délai fixé. Cela permet d'introduire le chronomètre de Scratch, mais également une variable qui conservera l'heure où le poisson a été attrapé : le script du poisson étant le seul qui ait besoin de cette variable, on la définira comme variable privée du lutin, à la différence du score que plusieurs lutins ont besoin de modifier ou de consulter, qui doit donc être une variable publique (ou globale). Cette distinction est une première introduction aux concepts de la programmation-objet qui n'a évidemment pas à être formalisée. » p 15-16</p>	<p><b>Variables</b></p> <p><input type="checkbox"/> heure</p> <p><b>Ingédients élémentaires :</b></p> <p>Déclarer une variable locale à un lutin</p> <p>Script du poisson</p>	<p>Le temps de disparition du poisson entre deux interactions avec le chat permet d'introduire le chronomètre de Scratch et la notion de portée des variables (publique/privée=globale/locale), ce qui constitue une première approche de la programmation objet</p>



### Ingrédients élémentaires :

Attendre la survenue d'un évènement

Manipuler une variable de type entier x2

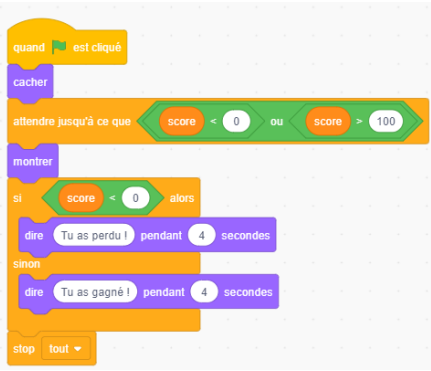
Manipuler une variable de type flottant

Utiliser une boucle non for x 2

Tester la survenue d'un évènement

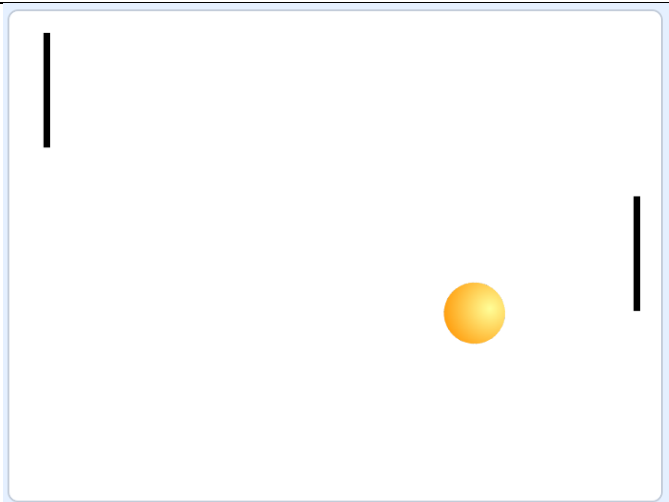
Utiliser une structure conditionnelle à une branche

Tester une inégalité sur des nombres

	<p>Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour) x2</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour)</p> <p>Utiliser un générateur de nombres aléatoires (utiliser une fonction avec paramètre avec retour) x2</p> <p>Modifier la valeur d'une variable (affectation)</p> <p>Modifier la valeur d'une variable (ajout)</p> <p>Utiliser la valeur d'une variable x3</p>	
<p><b>Étape 6 : Programmer un lutin (arbitre) qui gère la fin de jeu</b></p> <p>« Une dernière étape consiste à ajouter un quatrième lutin, l'arbitre, qui apparaît quand le score devient négatif (après des rencontres du chat et du chien) ou dépasse une valeur fixée (le chat a réussi à manger suffisamment de poissons). Là encore, on met en évidence la nécessité d'une boucle infinie pour scruter la valeur d'une variable globale, le score »</p>	<p><i>Script de l'arbitre</i></p> 	



	<p><u>Ingrédients élémentaires :</u></p> <p>Attendre la survenue d'un évènement</p> <p>Manipuler une variable de type entier</p> <p>Utiliser une boucle non for</p> <p>Utiliser une structure conditionnelle à deux branches</p> <p>Utiliser un connecteur logique « ou »</p> <p>Tester une inégalité sur des nombres x3</p> <p>Utiliser la valeur d'une variable x3</p> <p>Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour) x2</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x2</p>	
<p><b><u>Projet 4 : programmer le jeu « Pong »</u></b></p>		<p>Permet de travailler les angles, les instructions conditionnelles, les variables et la gestion des évènements</p>



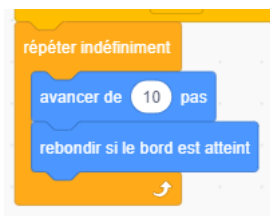
« Le professeur peut choisir ce type d'activité avec différents objectifs : travailler la notion de direction angulaire relative ou absolue, découvrir et utiliser les constructions conditionnelles, découvrir et utiliser des variables, programmer une gestion d'événements. » p. 16

**Étape 1 : Programmer le déplacement de la balle**

« On peut commencer par une activité concernant le rebond sur les bords d'une balle. [...]. Une répétition infinie de « avance de 10 », « rebondir lorsque le bord est atteint » permet alors l'observation de ce qui se passe aux bords : un objectif est que les élèves conjecturent les règles de changement de direction. » p. 16

« On peut alors proposer un défi, qui est de réaliser le même processus, mais sans utiliser le bloc « rebondir si le bord est atteint ». Il faut toutefois prendre garde à repérer l'approche des bords, en

**Technique 1 :**



**Ingrédients élémentaires :**

Utiliser une boucle non for

Permet d'observer le comportement de la balle au bord de la scène et de conjecturer les règles de changement de direction

ajoutant des répétitions infinies de test sur l'abscisse (inférieure à -215 ou supérieure à 215) et sur l'ordonnée. Ce défi n'est pas si simple pour des élèves inexpérimentés. »p 16

Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour)

Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour)

Technique 2 :

```
s'orienter à nombre aléatoire entre 1 et 360
répéter indéfiniment
  avancer de 10 pas
  si absceisse x < -208 alors
    s'orienter à 360 - direction
  sinon
    si absceisse x > 208 alors
      s'orienter à 360 - direction
    sinon
      si ordonnée y > 153 alors
        s'orienter à 180 - direction
      sinon
        si ordonnée y < -153 alors
          s'orienter à 180 - direction
```

Ingrédients élémentaires :

Proposer de la différenciation pédagogique sous la forme d'un défi

	<p>Utiliser une boucle non for</p> <p>Manipuler une variable de type entier x3</p> <p>Utiliser une structure conditionnelle à deux branches x3</p> <p>Utiliser une structure conditionnelle à une branches</p> <p>Tester une inégalité sur nombres x4</p> <p>Soustraire des nombres x4</p> <p>Utiliser la valeur d'une variable x8</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x6</p> <p>Utiliser un générateur de nombres aléatoires (utiliser une fonction avec paramètre avec retour)</p>	
<p><b>Étape 2 : Programmer le pilotage au clavier du déplacement des raquettes</b></p> <p>« On ajoute deux raquettes (que les élèves peuvent dessiner, un simple rectangle suffit). On pensera à les nommer de façon explicite, raquette gauche et raquette droite, par exemple. Il n'est pas très difficile d'associer à chaque raquette un script qui permet à</p>	<p><i>Script raquette (2 raquettes)</i></p>	<p>Façon simple de mettre en œuvre la programmation évènementielle</p>

*l'utilisateur de la faire monter ou descendre, à l'appui d'une touche ou d'une autre. Il faudra donc choisir quatre touches différentes du clavier, pour le joueur gauche et le joueur droit. Il s'agit là d'une programmation événementielle très simple à mettre en œuvre.* » p.

16



#### Ingrédients élémentaires :

Manipuler une variable de type entier


Utiliser une structure conditionnelle à une branche x2

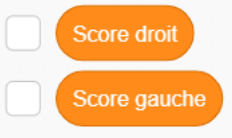
Tester une inégalité sur des nombres x2

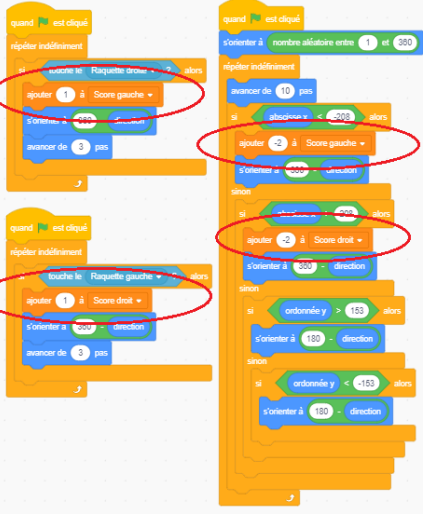
Attendre la survenue d'un évènement x2

Utiliser la valeur d'une variable x2

Modifier la valeur d'une variable (ajout) x2

	<p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour)</p>	
<p><b>Étape 3 : Programmer l'interaction entre la balle et les raquettes</b></p> <p>« Il convient maintenant de gérer le rebond de la balle sur chaque raquette. Pour cela on crée une boucle infinie qui teste le contact à l'aide du premier bloc de la catégorie « Capteurs ». Si la raquette gauche est touchée, il suffit de changer la direction de la balle en l'opposé de la direction actuelle. [...] C'est un objectif important que les élèves comprennent bien quelles commandes sont à écrire dans le script de la balle et quelles autres dans celui de chaque raquette : on introduit ici des concepts liés à la programmation-objet, sans avoir pour autant besoin du moindre formalisme. Une autre remarque s'impose ici : [...] il est sans doute plus simple de créer plusieurs scripts associés chacun à un comportement particulier : le déplacement perpétuel sur le terrain de jeu, la gestion du rebond sur les bords (si on ne veut pas utiliser la commande « rebondir si bord est atteint »), le contact avec la raquette gauche, le contact avec la raquette droite. Cela introduit à la fois la notion de programmation d'actions en parallèle et permet une mise au point beaucoup plus facile des programmes. »p 16-17</p>	<p><i>Script de la balle</i></p>  <p><u>Ingrédients élémentaires :</u></p> <ul style="list-style-type: none"> <li>Manipuler une variable de type entier</li> <li>Attendre la survenue d'un évènement x2</li> </ul>	<p>La présence de plusieurs lutin en interaction permet aux élèves de faire la distinction entre les scripts des différents lutins (introduction programmation-objet)</p> <p>La séparation des traitement (déplacement, contact avec les raquettes) pour la balle permet une mise au point plus facile du programme et d'introduire la programmation parallèle.</p>

	<p>Tester la survenue d'un évènement x2</p> <p>Soustraire des nombres x2</p> <p>Utiliser une boucle non for x2</p> <p>Utiliser une structure conditionnelle à une branche x2</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x2</p> <p>Utiliser la valeur d'une variable x2</p>	
<p><b>Étape 4 : Programmer le gestion du score</b></p> <p>« Plusieurs prolongements sont envisageables. En premier lieu, on peut ajouter un score pour chacun des deux joueurs, en créant deux variables score gauche et score droit (toujours utiliser des identifiants explicites et non abrégés), qui sont incrémentées à chaque rebond réussi sur la raquette correspondante. [...] On peut également ajouter des pénalités au score adéquat dans ce cas de balle ratée. » p. 17</p>	<p><b>Variables</b></p>  <p><b>Ingrédients élémentaires :</b></p> <p>Déclarer une variable globale x2</p>	<p>Permet des prolongements</p>

	 <p><u>Ingrédients élémentaires :</u>      Manipuler une variable de type entier x2      Modifier la valeur d'une variable (ajout) x4</p>	
<p><b>Projet 5 : Programmer un jeu de retournement de pièces</b></p>		<p>Permet de travailler les structures conditionnelles, les variables, les listes simples, les évènements déclenchés à la souris, la programmation parallèle, et les échanges de messages.</p>





« On dispose de quatre pièces, disposées initialement respectivement du côté face, pile, face, pile. Chaque clic sur une pièce la retourne, ainsi que son éventuelle voisine de droite. Le joueur gagne quand les quatre pièces sont du même côté. » p. 17

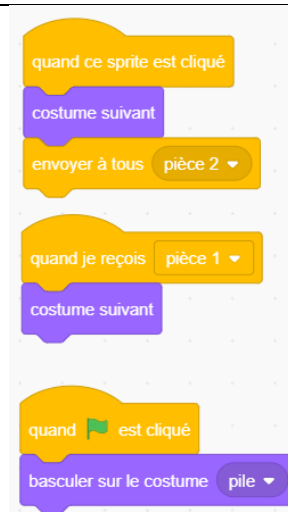
« L'objectif de cette activité est de travailler les notions suivantes :

- instructions conditionnelles ;
- utilisation d'une variable ;
- utilisation simple d'une liste ;
- gestion d'événements déclenchés par la souris ;
- déclenchement de plusieurs actions en parallèle ;
- échange de messages entre objets. » p. 17

**Étape 1 : Programmer les interactions avec les pièces**

*Script pièce 2 (4 pièces)*

« Chaque pièce est représentée par un lutin, avec deux costumes, pour le côté pile et le côté face [...] Une difficulté de l'activité tient à ce qu'un clic sur une pièce doit entraîner une action sur sa voisine : c'est une occasion naturelle d'introduire la notion de message entre lutins. » p. 17



#### Ingrédients élémentaires :

Attendre la survenue d'un évènement x2

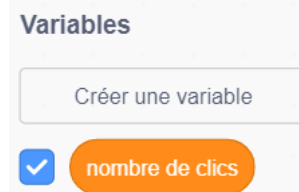
Envoyer un message

Attendre la réception d'un message

Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour)

#### Étape 2 : Programmer le compteur du nombre de clics

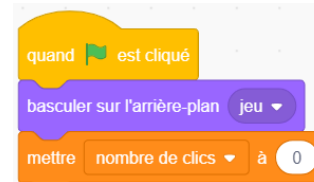
« L'initialisation de la variable qui compte le nombre de clics [...] est réalisée, à l'appui du drapeau vert, par un script lié à l'arrière-plan » p. 18



Ingrédients élémentaires :

Déclarer une variable globale

*Script de l'arrière-plan*



Ingrédients élémentaires :

Manipuler une variable de type entier


Attendre la survenue d'un évènement

Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour)

Modifier la valeur une variable globale (affectation)

*Script pièce 2 (4 pièces)*

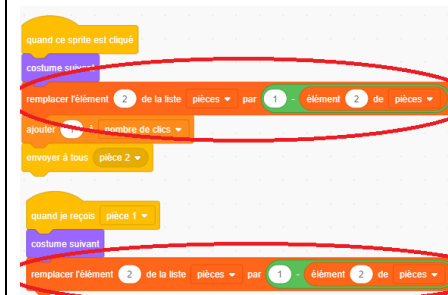


	<p><u>Ingrédients élémentaires :</u></p> <p>Manipuler une variable de type entier</p> <p>Modifier la valeur d'une variable (ajout)</p>	
<p><b>Étape 3 : Programmer la détection de la victoire</b></p> <p>« L'initialisation [...] de la liste qui retient les faces courantes des pièces est réalisée, à l'appui du drapeau vert, par un script lié à l'arrière-plan. Cette méthode permet de distinguer ce qui relève du comportement global de l'application et ce qui est spécifique de chaque pièce. On notera qu'on a choisi également de tester la victoire dans le même script, lié à l'arrière-plan. L'utilisation d'une liste paraît indispensable, car chaque lutin est le seul à avoir accès au numéro de son costume courant : aucun lutin ne peut donc tester l'égalité des côtés des quatre pièces en utilisant seulement le bloc « costume n° ». Cette activité ne demande qu'une utilisation très élémentaire d'une liste, et peut donc servir d'introduction à cette notion. Le test de victoire peut être réalisé de plusieurs façons : par une imbrication de plusieurs conditionnelles, comme dans la production jointe, mais aussi en utilisant les connecteurs logiques. Ce peut être intéressant de ne pas guider les élèves sur telle ou telle voie, mais de les laisser choisir. On peut s'attendre à ce que les élèves cherchent un bloc pour tester des égalités simultanées, et expliquer qu'on ne peut tester que l'égalité de deux quantités. On a choisi de représenter les valeurs pile et face par les nombres 1 et 0, ce qui permet d'utiliser « remplacer l'élément 2 de la liste pièce par 1 - élément 2 de pièce » qui devra naturellement faire l'objet d'un travail spécifique : une fois qu'on aura trouvé comment modifier un élément</p>	<p>Créer une liste</p> <p><input type="checkbox"/> pièces</p> <p><u>Ingrédients élémentaires :</u></p> <p>Déclarer une liste globale</p> <p>Script de l'arrière-plan</p>  <p><u>Ingrédients élémentaires :</u></p> <p>Manipuler une variable de type liste</p> <p>Modifier la valeur d'un élément d'une liste x4</p>	<p>Initialiser les variables globales dans le script de l'arrière-plan permet de bien distinguer les éléments globaux et locaux.</p> <p>L'utilisation d'une liste globale contenant la position de toutes les pièces est rendu nécessaire par le fait qu'un lutin n'a pas accès au costume des autres.</p> <p>C'est une utilisation simple permettant de faire office d'introduction</p> <p>Représenter pile/face par 1/0 permet d'enregistrer le changement de face par un calcul (1-face précédente)</p>

d'une liste, il faudra encore trouver comment une seule expression arithmétique simple permet de passer de pile à face et vice versa » p.

18

### Script de la pièce 2 (4 pièces)



#### Ingrédients élémentaires :

Manipuler une variable de type liste

Accéder à un élément d'une liste  
x2

Modifier la valeur d'un élément d'une liste  
x2

Soustraire des nombres x2

### Script de l'arrière-plan

#### Technique 1



#### Ingrédients élémentaires :

Utiliser une boucle non for

Utiliser une structure conditionnelle à une branche x3

Tester une égalité sur des nombres x3

Accéder à un élément d'une liste x6

Agir sur l'arrière-plan avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour)

## Technique 2



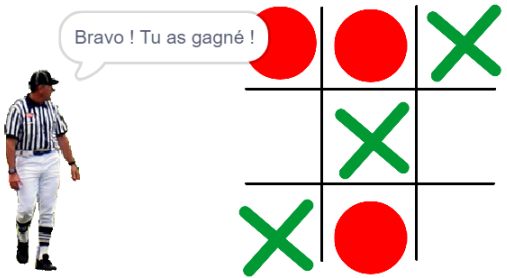
### Ingrédients élémentaires :

Utiliser une boucle non for

Utiliser une structure conditionnelle à une branche x1

Utiliser un connecteur logique

« et » x2

	<p>Tester une égalité sur des nombres x3</p> <p>Agir sur l'arrière-plan avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour)</p> <p>Accéder à un élément d'une liste x6</p>	
<p><b>Projet 6 : programmer le jeu Tic-tac-toe</b></p> <div data-bbox="324 619 974 1114" style="border: 1px solid black; padding: 10px; text-align: center;"> <p>Le tic-tac-toe</p>  </div> <p>« Cette activité est plus ambitieuse, et le fichier joint n'est qu'un exemple de production qui demande un niveau de maîtrise important. Mais on pourra s'en inspirer, car elle se découpe aisément en sous-problèmes dont la résolution permet d'approcher de nombreuses notions différentes » p. 18</p>	<p>(Cette tâche étant beaucoup plus complexe que les autres : 10 lutins, 32 scripts, nous ne précisons pas les étapes et ne reproduisons pas les scripts. Nous nous contentons de donner la liste des ingrédients élémentaires)</p> <div data-bbox="1164 901 1400 1220" style="border: 1px solid gray; padding: 5px;"> <p>Créer une variable</p> <p><input type="checkbox"/> case</p> <p><input type="checkbox"/> coups</p> <p><input type="checkbox"/> s</p> <p>Créer une liste</p> <p><input type="checkbox"/> valeurs</p> </div> <p><u>Ingrédients élémentaires :</u></p> <p>Déclarer une variable globale x3</p>	

	<p><b>Déclarer une liste globale</b></p> <p><i>Scripts de l'arbitre</i></p> <p><u>Ingrédients élémentaires :</u></p> <p>Manipuler une variable de type entier x3</p> <p>Manipuler une variable de type liste</p> <p>Modifier la valeur d'une variable (affectation) x11</p> <p>Utiliser la valeur d'une variable x10</p> <p>Utiliser une structure conditionnelle à une branche x3</p> <p>Utiliser une boucle non-for</p> <p>Tester une égalité sur des nombres x3</p> <p>Attendre la survenue d'un évènement</p> <p>Additionner des nombres x16</p> <p>Agir sur un lutin sans paramètre (utiliser une méthode d'un objet sans paramètre sans retour) x6</p>	
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--



	<p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x6</p> <p>Utiliser un générateur de nombres aléatoires (utiliser une fonction avec paramètre avec retour) x2</p> <p>Envoyer un message</p> <p>Attendre la réception d'un message x3</p> <p>Accéder à un élément d'une liste x25</p> <p>Créer un bloc construit avec une entrée (définir une fonction avec un paramètre sans retour)</p> <p>Utiliser un bloc construit avec une entrée (appeler une fonction avec un paramètre sans retour) x8</p> <p><i>Scripts de la case 2 (9 cases) :</i></p> <p><u>Ingrédients élémentaires :</u></p> <p>Manipuler une variable de type entier x2</p> <p>Manipuler une variable de type liste</p>	
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

	<p>Modifier la valeur d'une variable (affectation)</p> <p>Modifier la valeur d'une variable (ajout) x2</p> <p>Utiliser la valeur d'une variable x4</p> <p>Utiliser une structure conditionnelle à une branche x2</p> <p>Tester une égalité sur des nombres x2</p> <p>Attendre la survenue d'un évènement x2</p> <p>Agir sur un lutin avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x3</p> <p>Envoyer un message x4</p> <p>Attendre la réception d'un message</p> <p>Accéder à un élément d'une liste</p> <p>Modifier la valeur d'un élément d'une liste x3</p>	
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

## A.2 Cycle 4 – Technologie

### A.2.1 Programme d'enseignement PT4 (MEN, 2020)

#### A.2.1.1 Organisations didactiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Traiter les notions d'algorithmique conjointement avec les cours de mathématiques (contenu)</b></p> <p>« Les notions d'algorithmique sont traitées conjointement en mathématiques et en technologie » p. 123</p>		
<p><b>Faire travailler les élèves dans une démarche de projet en répondant à un besoin/problème. (dispositif/activité)</b></p> <p>« Il est également l'occasion de mettre en place des modalités d'enseignement fondées sur une pédagogie de projet » p. 117</p> <p>« Dans le cadre des projets, les élèves utilisent des outils numériques adaptés (organiser, rechercher, concevoir, produire, planifier, simuler) et conçoivent tout ou partie d'un programme, le compilent et l'exécutent pour répondre au besoin du système et des fonctions à réaliser. » p. 123</p> <p>« Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème simple. » p 117</p> <p>« Design, innovation et créativité : imaginer des solutions pour produire des objets et des éléments de programmes informatiques en réponse au besoin : [...] Représentation de solutions (croquis, schémas, algorithmes).[...] » p 119</p>	<p>Les élèves conçoivent un programme pour répondre au besoin du système et des fonctions à réaliser.</p> <p>Résoudre un problème</p> <p>Répondre à un besoin</p>	

<p><b>Proposer une pédagogie active (pédagogique)</b></p> <p>« Il est également l'occasion de mettre en place des modalités d'enseignement fondées sur une pédagogie de projet, active et collaborative. Pour donner du sens aux apprentissages et valoriser le travail des élèves, cet enseignement doit se traduire par la réalisation de productions collectives (programme, application, animation, sites, etc.) dans le cadre d'activités de création numérique, au cours desquelles les élèves développent leur autonomie, mais aussi le sens du travail collaboratif. » p. 117</p>	<p>Proposer des activités de création numérique</p>	<p>Donne du sens aux apprentissages.</p> <p>Valorise le travail des élèves</p> <p>Développe l'autonomie des élèves et le sens du travail collaboratif</p>
<p><b>Faire collaborer les élèves (dispositif/activité)</b></p> <p>« Il est également l'occasion de mettre en place des modalités d'enseignement fondées sur une pédagogie de projet, active et collaborative. Pour donner du sens aux apprentissages et valoriser le travail des élèves, cet enseignement doit se traduire par la réalisation de productions collectives (programme, application, animation, sites, etc.) dans le cadre d'activités de création numérique, au cours desquelles les élèves développent leur autonomie, mais aussi le sens du travail collaboratif. » p. 117</p>	<p>Faire réaliser aux élèves des productions collectives (programme, application, animation, sites, etc.)</p>	<p>Donne du sens aux apprentissages.</p> <p>Valorise le travail des élèves</p> <p>Développe l'autonomie des élèves et le sens du travail collaboratif</p>
<p><b>Faire utiliser aux élèves un langage de programmation graphique (artéfact)</b></p> <p>« Ils peuvent être initiés à programmer avec un langage de programmation couplé à une interface graphique pour en faciliter la lecture. La conception, la lecture et la modification de la programmation sont réalisées au travers de logiciels d'application utilisant la représentation graphique simplifiée des éléments constitutifs de la programmation. » p. 123</p>	<p>Faire concevoir, lire ou modifier un programme dans un logiciel de programmation graphique.</p>	<p>Les langages graphiques facilitent la lecture des programmes.</p> <p>Les représentation graphique des éléments de programmation sont plus simples.</p>

<p><b>Faire exprimer aux élèves leur pensée à l'aide l'algorithmes (contenu/activité)</b></p> <p>« Comparer et commenter les évolutions des objets et systèmes : : [...] <i>Notion d'algorithme.</i> » p. 120</p>		
<p><b>Faire programmer aux élèves des systèmes technologiques réels ou simulés (artéfact)</b></p> <p>« Les moyens utilisés sont des systèmes pluri-technologiques réels didactisés ou non, dont la programmation est pilotée par ordinateur ou une tablette numérique. Ils peuvent être complétés par l'usage de modélisation numérique permettant des simulations et des modifications du comportement. » p. 123</p>	<p>Faire programmer les systèmes réels à l'aide d'ordinateur ou de tablettes</p>	
<p><b>Introduire les notions de manière progressive (contenu)</b></p> <p>« Repères de progressivité.</p> <p><i>En 5<sup>ème</sup> : traitement, mise au point et exécution de programme simple avec un nombre limité de variables d'entrée et de sortie, développement de programmes avec des boucles itératives.</i></p> <p><i>En 4<sup>ème</sup> : traitement, mise au point et exécution de programme avec introduction de plusieurs variables d'entrée et de sortie.</i></p> <p><i>En 3<sup>ème</sup> : introduction du comptage et de plusieurs boucles conditionnelles imbriquées, décomposition en plusieurs sous-problèmes.</i> » p 124</p>	<p>5<sup>ème</sup> : programmes simples, boucles itératives</p> <p>4<sup>ème</sup> : programmes avec plusieurs variables d'entrée et de sortie</p> <p>3<sup>ème</sup> : comptage, boucles conditionnelles imbriquées, décomposition en sous-problèmes.</p>	<p>Les modélisations numériques permettent des simulations et des modifications de comportement</p>
<p><b>Faire modifier le programme d'un système réel pour l'améliorer (activité)</b></p>	<p>Modifier le programme d'un système pour améliorer son comportement, ses performances ou mieux répondre à une problème.</p>	

<p>« Modifier un programme existant dans un système technique, afin d'améliorer son comportement, ses performances pour mieux répondre à une problématique donnée. » p. 123</p>		
<p><b>Faire programmer un système réel en réponse à un cahier des charges (activité/artefact)</b></p> <p>« Attendus de fin de cycle : [...] - Écrire, mettre au point et exécuter un programme. » p 123</p> <p>« Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu. » p. 123</p> <p>« Concevoir, paramétrer, programmer des applications informatiques pour des appareils nomades. » p. 123</p> <p>« Écrire, à partir d'un cahier des charges de fonctionnement, un programme afin de commander un système ou un système programmable de la vie courante, identifier les variables d'entrée et de sortie. » p. 123</p> <p>« Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande. » p 123</p>	<p>Programmer des appareils nomades</p> <p>Programmer un système programmable de la vie courante à partir d'un cahier des charges.</p> <p>Programmer un système réel pour répondre à un problème posé en décomposant en sous-problème</p>	
<p><b>Faire décrire aux élèves les éléments de programmation d'un système réel (activité/artefact)</b></p> <p>« Observer et décrire le comportement d'un robot ou d'un système embarqué. En décrire les éléments de sa programmation. » p. 123</p>	<p>Décrire les éléments de programmation d'un robot ou d'un système embarqué</p>	
<p><b>Faire agencer un robot pour répondre à un programme donnée (activité/artefact)</b></p>	<p>Interconnecter des capteurs et des actionneurs pour construire un robot</p>	


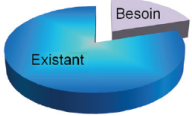


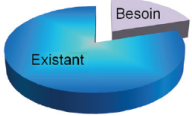


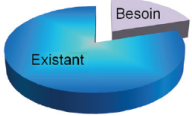

<p>« Agencer un robot (capteurs, actionneurs) pour répondre à une activité et un programme donnés. » p. 123</p>		
-----------------------------------------------------------------------------------------------------------------	--	--

## A.2.2 Ressource d'accompagnement R1T4 (MEN, 2016a)

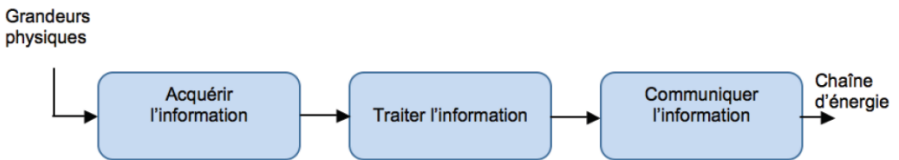
### A.2.2.1 Organisations didactiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>L'enseignement doit être réalisé en coordination avec les professeurs de mathématiques (contenu)</b></p> <p>« Ces trois thématiques s'articulent avec une quatrième qui est liée à l'enseignement d'informatique dont la responsabilité incombe aux professeurs de technologie et de mathématiques » p. 4</p> <p>« L'informatique doit être intégrée aux séquences d'enseignement de la technologie et avoir un prolongement dans le domaine des mathématiques pour ce qui est de la représentation des nombres et de l'algorithmique » p. 17</p> <p>« Une partie des compétences de l'enseignement d'informatique en technologie est commune avec le thème « algorithmique et programmation » du programme de mathématiques. Ces compétences développées dans l'une ou l'autre discipline n'ont pas vocation à être vues deux fois de manière séparée, mais à être réinvesties dans une problématique plus complexe de pilotage de système en technologie. Il appartient aux professeurs de ces deux disciplines de mettre en cohérence l'enseignement de l'informatique par la définition de thématiques communes. Les enseignements pratiques interdisciplinaires offrent de ce point de vue un cadre privilégié. » p. 18</p>	<p>Les enseignements de l'informatique en technologie doivent avoir un prolongement en mathématiques (représentation des nombres et algorithmique)</p> <p>Les compétences en programmation n'ont pas à être vues deux fois de manière séparée.</p> <p>Les professeurs des deux disciplines doivent se coordonner pour mettre en cohérence leurs enseignements.</p> <p>Les EPI offre un cadre privilégié pour ces mises en œuvre.</p>	
<p><b>Proposer des activités de programmation dans un contexte technologique : objets, systèmes (artefact)</b></p>	<p>Proposer des approches concrètes et actives.</p>	



<p>« <i>L'informatique et la programmation ne peuvent être enseignées hors d'un contexte technologique. Les concepts, que devront acquérir les élèves, seront abordés à partir d'activités liées à l'enseignement de la technologie, c'est-à-dire par une approche concrète et active, et autant qu'il est possible de le faire à partir des objets et systèmes présents dans le laboratoire.</i> » p. 4</p> <p>« <i>Ces compétences [en algorithmique et programmation] développées dans l'une ou l'autre discipline [maths ou techno] n'ont pas vocation à être vues deux fois de manière séparée, mais à être réinvesties dans une problématique plus complexe de pilotage de système en technologie.</i> » p. 18</p>	<p>Faire travailler les élèves à partir d'objets et systèmes présents dans le laboratoire.</p> <p>Les compétences en programmation doivent être réinvesties en technologie dans un contexte de pilotage de systèmes</p>																	
<p><b>Faire utiliser aux élèves la programmation graphique (artefact)</b></p> <p>« <i>Les langages de programmation doivent se limiter à de la programmation graphique qui est interprétée et compilée dans une base robotique ou des platines microcontrôleurs.</i> » p. 18</p>	<p>Les programmes sont exécutés par des robots ou des microcontrôleurs</p>																	
<p><b>Proposer trois types de démarche pédagogiques (activité)</b></p> <table border="1" data-bbox="165 959 1055 1326"> <thead> <tr> <th></th> <th>DÉMARCHE D'INVESTIGATION</th> <th>DÉMARCHE DE RÉOLUTION DE PROBLÈMES TECHNIQUES</th> <th>DÉMARCHE DE PROJET</th> </tr> </thead> <tbody> <tr> <td>Objectif de la démarche</td> <td>Découvrir et comprendre</td> <td>Agir</td> <td>Décider et agir</td> </tr> <tr> <td>Activité dans la démarche</td> <td>Analyser et chercher</td> <td>Résoudre</td> <td>Concevoir, développer et agir</td> </tr> <tr> <td>Support ou point de départ de la démarche</td> <td>Système abouti </td> <td>Système perfectible </td> <td>Cahier des charges </td> </tr> </tbody> </table> <p>p. 6</p>		DÉMARCHE D'INVESTIGATION	DÉMARCHE DE RÉOLUTION DE PROBLÈMES TECHNIQUES	DÉMARCHE DE PROJET	Objectif de la démarche	Découvrir et comprendre	Agir	Décider et agir	Activité dans la démarche	Analyser et chercher	Résoudre	Concevoir, développer et agir	Support ou point de départ de la démarche	Système abouti 	Système perfectible 	Cahier des charges 	<p>1) Démarche d'instigation : étudier un système existant dans le but de découvrir et comprendre</p> <p>2) Démarche de résolution de problèmes techniques : résoudre un problème exprimé sous forme d'un besoins en agissant sur un système existant</p> <p>3) Démarche de projet : concevoir et développer un</p>	
	DÉMARCHE D'INVESTIGATION	DÉMARCHE DE RÉOLUTION DE PROBLÈMES TECHNIQUES	DÉMARCHE DE PROJET															
Objectif de la démarche	Découvrir et comprendre	Agir	Décider et agir															
Activité dans la démarche	Analyser et chercher	Résoudre	Concevoir, développer et agir															
Support ou point de départ de la démarche	Système abouti 	Système perfectible 	Cahier des charges 															

	système en réponse à un cahier des charges	
<p><b>Mettre en place une pédagogie collaborative (dispositif)</b></p> <p>« Les chercheurs qui étudient le comportement des jeunes générations pointent des traits qui leur sont spécifiques : l'impatience [...] ; le multitâche [...] ; la communauté virtuelle [...] ; le flux continu [...] ; la nécessité d'être acteurs de leurs apprentissages ; une méfiance à l'égard de l'autorité et de l'information descendante. [...] Pour cela, une pédagogie collaborative doit être mise en place. Cette pédagogie collaborative nécessite d'aménager les salles en îlots, chacun accueillant une équipe d'élèves. Quatre ou cinq élèves, qui travaillent sur un îlot, ne doivent pas constituer un groupe, mais bien une équipe, dans laquelle chacun aura un rôle complémentaire à celui des autres membres, pour réaliser les tâches correspondant aux objectifs fixés par le professeur. [...] Il est important que le professeur propose des activités collaboratives (mises en commun de réalisations complémentaires, information de l'équipe sur l'avancement des travaux de chacun, préparation de documents de synthèse, etc.), tout en intégrant du travail personnel aux différents membres de l'équipe. Des phases de travail autonome viennent compléter des phases de travail avec l'équipe au complet. L'hétérogénéité au sein d'une même équipe ne doit pas être considérée par le professeur comme un handicap mais bien au contraire comme un point fort sur lequel il pourra s'appuyer pour favoriser la mutualisation des connaissances et des compétences au sein du groupe classe. [...] La possibilité de disposer d'une salle banalisée ou d'un « pôle réunion » annexé ou intégré au laboratoire de technologie organisé en îlots, permet au professeur de mixer les modes d'apprentissage sur une même séance. Ce « pôle réunion »</p>	<p>Mettre en place des îlots de 4 ou 5 élèves.</p> <p>Chaque îlot forme une équipe dans laquelle chacun aura un rôle complémentaire de celui des autres.</p> <p>Les travaux proposés doivent mélanger tâches collaboratives à réaliser en groupe et tâches personnelles à réaliser en autonomie.</p> <p>L'hétérogénéité au sein des îlots est à privilégier</p> <p>Possibilité de disposer d'un espace « pôle de réunion » pour les moments de mise en commun ou de créativité collectives.</p>	<p>Les chercheurs ont montré que les jeunes générations ont des traits spécifiques qui correspondent bien à la pédagogie collaborative.</p> <p>Permet de favoriser la mutualisation des connaissances et des compétences.</p> <p>Permet de mixer les modes d'apprentissage sur une même séance</p>

<p>peut aussi être envisagé comme un espace consacré aux moments de créativité (brain-storming, réalisation de cartes mentales, etc.) durant les projets. » p. 11</p>		
<p><b>Présenter les systèmes en chaîne d'énergie et chaîne d'information (contenu)</b></p> <p>« La décision d'une action résulte toujours de la prise en compte d'informations extérieures et de leur analyse. L'organisation des systèmes en chaîne d'énergie et chaîne d'information fait apparaître un bloc de traitement de l'information qui, la plupart du temps, a comme structure un module programmable. » p. 17</p>  <pre> graph LR     A[Acquérir l'information] --&gt; B[Traiter l'information]     B --&gt; C[Communiquer l'information]     GP[Grandeurs physiques] --&gt; A     C --&gt; CE[Chaîne d'énergie]   </pre>		<p>La prise de décision s'appuie toujours sur la prise en compte d'informations extérieures et de leur analyse.</p>
<p><b>Toujours mettre la chaîne d'information complète à disposition des élèves (contenu)</b></p> <p>« Si les éléments du chapitre d'informatique s'inscrivent naturellement dans le bloc de traitement de l'information, il est primordial de mettre à disposition des élèves, à tout moment, la chaîne d'information complète. C'est un moyen de matérialiser l'action d'un programme par l'observation d'un phénomène concret. » p. 18</p>		<p>Les éléments de programmation se retrouvent que dans le bloc de travail de l'information.</p> <p>Permet de matérialiser l'action d'un programme par d'observation d'un phénomène concret</p>
<p><b>Faire apparaître les écart entre simulation et réalité dans les blocs d'acquisition de l'information (contenu)</b></p> <p>« L'acquisition d'une information s'obtient par la transformation d'une grandeur physique en une information numérique, qui aussi</p>	<p>Dans un premier temps, faire simuler les entrées du système par des saisies numériques directes ou à l'aide d'interrupteurs ou de</p>	<p>Les modules d'acquisition d'informations transforment imparfaitement les grandeurs</p>

<p><i>fidèle soit-elle, comporte des imperfections. Durant les phases de simulation ou de prototypage, cette acquisition est simplifiée soit par la saisie directe d'une donnée numérique dans les logiciels de simulation, soit par le raccordement d'interrupteurs ou de potentiomètres à une carte microcontrôleur. La mise en évidence de différences de comportement entre celui obtenu par simulation et celui obtenu sur l'objet réel est un moyen de faire apparaître la notion d'écart et l'impact de la formulation d'hypothèses simplificatrices</i> » p. 18</p>	<p>potentiomètres. Puis mettre en évidence la différence de comportement du système lors d'acquisition de données réelles.</p>	<p>physiques en grandeur numérique. Permet de faire apparaître la notion d'écart. Permet de voir l'impact de la formulation d'hypothèses simplificatrices.</p>
<p><b>Mettre en évidence l'action de l'information communiquée vers la chaîne d'énergie (contenu)</b></p> <p><i>« De la même manière, le bloc « communiquer l'information » se résume quelquefois à la visualisation de l'information. L'action de cette information sur la chaîne d'énergie doit aussi être mise en évidence afin d'en mesurer sa traduction »</i> p. 18</p>		<p>Le bloc communiquer l'information se résume quelquefois à de la visualisation (sans traduction en action)</p> <p>Permet de mesurer la traduction de l'information en énergie</p>
<p><b>Organiser les activités de manière progressive (contenu/activité)</b></p> <p><i>« Les activités liées au traitement de l'information au cycle 4 doivent être progressives. Dans un premier temps, les problèmes doivent se limiter à montrer qu'une action peut résulter uniquement d'une combinaison des informations d'entrées. Les objets de la domotique, tels que les éclairages automatiques, les systèmes de surveillance et de sécurité qui intègrent les fonctions logiques de base, peuvent être aisément étudiés au collège. Il convient par la suite de complexifier la programmation en montrant qu'une action dépend d'un enchaînement de décisions. Les structures de programmation telles que « si...alors...sinon », « faire...tant que... » permettent d'initier les</i></p>	<p>Deux temps :</p> <p>1) l'action dépend uniquement d'une combinaison des informations d'entrées : objets domotiques (éclairage automatiques, systèmes de surveillance et de sécurité</p> <p>2) l'action dépend d'un enchaînement de décisions articulées par les structures conditionnelles et les boucles non fors : systèmes</p>	<p>Les systèmes domotiques intègrent les fonctions logiques de base et peuvent être étudiés aisément au collège</p> <p>Permet de complexifier la programmation</p>

<p>élèves aux comportements des systèmes automatiques. De nombreux systèmes de la vie courante et les supports utilisés dans la progression pédagogique peuvent servir d'exemples. » p. 18</p>	<p>automatiques (systèmes de la vie courante)</p>	
<p><b>Proposer aux élèves de travailler en équipe sur des programmes plus importants (dispositif/activité)</b></p> <p>« Dans le cadre d'un projet, un programme informatique est l'association de plusieurs sous-programmes écrits par des personnes différentes. Cela impose une décomposition fonctionnelle des tâches à programmer et un partage des variables du programme. Cette décomposition doit être proposée par le professeur afin de limiter les dispersions qui poseraient des difficultés lors de l'assemblage final. Le travail en équipes avec une distribution des tâches à programmer est indispensable pour mettre en œuvre ce principe de travail collaboratif de programmation. Cette décomposition fonctionnelle des programmes doit faire écho à celle vue dans l'analyse des systèmes. » p. 18</p>	<p>Décomposer les tâches à programmer de manière fonctionnelle et partager les variables</p>	<p>Lors d'un projet, un programme informatique est composé de plusieurs sous-programmes écrits par des personnes différentes.</p>
	<p>La décomposition est proposée par le professeur</p>	<p>Permet de limiter les dispersions qui peuvent poser des problèmes lors de l'assemblage</p>
	<p>Proposer un travail en équipe avec une distribution des tâches à programmer</p>	<p>Permet de mettre en œuvre le principe du travail collaboratif</p>
<p><b>Faire travailler les élèves dans le laboratoire de technologie (dispositif)</b></p> <p>« intégré au laboratoire de technologie organisé en îlots » p11</p> <p>« Un mini réseau indépendant du réseau du collège peut être assemblé dans le laboratoire de technologie »p. 21</p>		

## A.2.3 Ressource d'accompagnement R2T4 (MEN, 2016b)

### A.2.3.1 Organisations informatiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Séquence 1 : créer une carte interactive permettant de présenter sa ville</b></p> <p>« <i>Présentation/objectifs</i> : Comment rendre une vue aérienne interactive ? <b>But de l'EPI : faire réinvestir aux élèves leurs lieux d'habitation et leur faire s'approprier la ville par les mots, en sublimant un univers qu'ils connaissent finalement mal par le français (compétences d'écriture et d'étude du langage poétique), et la technologie (réalisation d'un programme en ligne sous Scratch cartographiant les quartiers et lieux de la ville avec une interactivité avec la souris (petit bonhomme mobile faisant au gré de la souris découvrir la ville de manière interactive à partir d'une vue aérienne (exploration et description des lieux par des photos et des poèmes saisis et/ou enregistrés (son ou vidéo)))</b> »</p> <p>« <i>Question directrice 2</i> : Comment mettre en forme notre travail de manière attractive et ludique afin de le diffuser ? »</p> <p>« <i>Activités</i> : Proposition par les élèves de solutions courantes à cette question (diaporama, exposition de photos, articles dans le journal du collège...). Présentation par le professeur d'applications créées sous Scratch et échanges sur l'intérêt qu'aurait ce logiciel dans la mise en valeur et la diffusion de leurs travaux. <b>Création par les groupes d'élèves de scénarii simples permettant à partir d'une vue aérienne de la ville, de photos prises et des poèmes réalisés de mettre en scène l'ensemble. Réalisation d'une application qui permettrait de</b></p>	<p>Créer de façon interdisciplinaire (français technologie) une animation interactive en scratch : à partir d'une photo aérienne, mettre en scène des photos et de poèmes (texte, son ou vidéos) apparaissant au gré des déplacements de la souris.</p> <p><b>Phases :</b></p> <p>Prendre des photos de la ville et réaliser des poèmes. (français)</p> <p>Création des scénarios sur Scratch par groupes : mettre en scène photos et poèmes à partir d'une vue aérienne</p> <p>Test et ajustements</p> <p>Présentation finale devant le professeur de français et les parents</p> <p><b>Ingrédients élémentaires :</b></p> <p>Utiliser une boucle</p>	<p>Ce type de tâches permet aux élèves de réinvestir leurs lieux d'habitation et de s'approprier la ville par les mots et la technologie.</p> <p>Le logiciel scratch permet de créer des projets interactifs permettant de mettre en valeur et de diffuser les travaux des élèves.</p>

<p><i> naviguer de manière interactive dans la ville en sublimant les lieux choisis par les élèves. »</i></p> <p><i>« Question directrice 3 : Comment saisir notre scénario dans le logiciel Scratch ? »</i></p> <p><i>« Activités : Les élèves n'ayant jamais utilisé ce genre de programme, présentation par le professeur des fonctions nécessaires à la saisie de leurs scénarios (costumes, boucles, instructions conditionnelles liées au clic souris...). Saisie des scénarios par les groupes, test et ajustement. Présentation finale de ceux-ci devant le professeur de français et les parents. »</i></p> <p><i>« Bilan : Bilan lié aux notions de boucles, d'instructions conditionnelles, d'algorithme et de programme. » Onglet P1_9</i></p>	<p>Utiliser une structure conditionnelle</p> <p>Déclencher une action par un évènement</p>	
<p><b>Séquence 2 : programmer un éclairage automatique</b></p> <p><i>« Présentation/objectifs : Comment programmer un éclairage automatique ? À partir du système d'éclairage automatique de chemin piétonnier d'accès au parking. Cette séquence a pour but de faire découvrir la notion de système automatique. [...] La définition des E/S d'un système automatique est une étape préliminaire pour rédiger un algorithme ou un algorithme, ce qui permettra d'écrire le programme informatique du système. »</i></p> <p><i>« Question directrice 3 : Comment écrire le programme d'un système automatique ? »</i></p> <p><i>« Activités : Observer et décrire les différentes étapes du fonctionnement d'un système automatisé. Définir les variables entrées et les variables de sorties du futur programme et écrire ce</i></p>	<p><b>Phases :</b></p> <p>Observer et décrire les étapes de fonctionnement du système</p> <p>Définir les variables d'entrées et de sorties du futur programme</p> <p>Écrire le programme sous forme d'algorithme.</p> <p>Programmation Arduino ou Ardublock</p> <p><b>Ingrédients élémentaires :</b></p> <p>Utiliser une boucle</p> <p>Utiliser les fonctions logiques (et/ou)</p>	<p>Cette séquence permet de découvrir la notion de système automatique</p> <p>La définition des entrées-sorties d'un système est un étape préliminaire pour rédiger un algorithme</p> <p>Écrire un algorithme ou un algorithme permet d'écrire le programme informatique</p>



<p>dernier sous forme d'algorigramme. Programmation arduino et/ou ardublock. »</p> <p>« <u>Bilan</u> : Définir la notion d'algorithme, de programme et de variables E/S. Fonctions logiques et boucles itératives. »</p> <p>« <u>Ressources</u> : Vidéo et maquette d'un système d'éclairage (détection mouvement et luminosité). Fiches ressources : symboles électriques, algorigramme, le logiciel ardublock. » Onglet P2-1</p>		
<p><b>Séquence 3 : piloter un panneau solaire pour optimiser l'apport solaire</b></p> <p>« <u>Présentation/objectifs</u> : Comment optimiser les apports solaires sur le toit d'un immeuble ? Les élèves ont, lors d'une phase d'initiation, appris à programmer avec le logiciel Scratch. Ils savent programmer une interface programmable (type Groomy ou PICAXE 28X2) avec le logiciel Scratch. Un tracker solaire radiocommandé a été réalisé et est présenté aux élèves. »</p> <p>« <u>Question directrice 2</u> : Comment réaliser notre solution ? »</p> <p>« <u>Activités</u> : Modification et/ou réalisation d'un programme pour rendre un tracker solaire autonome afin qu'il maintienne en position optimale une cellule photovoltaïque tout au long de la journée. Simulation, puis programmation d'une interface pour piloter le tracker. »</p> <p>« <u>Bilan</u> : Revue de projet : présentation de la solution finale (croquis, modélisation, photos de la solution finie) plus explications sur "comment l'avons-nous réalisée ?" et "comment l'avons-nous programmée ?" »</p>	<p><u>Phases</u> :</p> <p>Modifier ou réaliser un programme en scratch pour qu'un tracker solaire maintienne en position optimale une cellule photovoltaïque tout au long de la journée.</p> <p>Simulation puis programmation d'une interface pour piloter le tracker.</p> <p><u>Ingrédients élémentaires</u> :</p>	



<p>« <u>Ressources</u> : Machine du laboratoire de technologie (Imprimante 3D, CFAO, cisaille...), logiciel de modélisation (Sketchup Make), <b>logiciel Scratch</b>. » Onglet P6-5</p>		
<p><b>Séquence 4 : programmer un mini-robot pour qu'il évite un obstacle puis qu'il suive une ligne</b></p> <p>« <u>Présentation/objectif</u> : Par quoi et comment programmer un objet technique ? Un mini-robot doit évoluer dans une enceinte fermée. Cette séquence permet de travailler sur logiciel et d'étudier les capteurs du mini-robot. Pour évoluer dans son environnement, le mini-robot doit acquérir des informations grâce à des capteurs. Ceux-ci permettent au mini-robot de connaître la position d'obstacles et d'agir pour les éviter. »</p> <p>« <u>Question directrice 1</u> : <b>Comment éviter un obstacle ?</b> »</p> <p>« <u>Activités</u> : <b>Écriture d'organigrammes simples</b>. »</p> <p>« <u>Bilan</u> : Définir les capteurs utiles pour éviter un obstacle. Indiquer comment <b>utiliser des variables et des boucles</b>. »</p> <p>« <u>Ressources</u> : Situation déclenchante robot sans détecteur. Tutoriel sur le logiciel de programmation. »</p> <p>« <u>Question directrice 2</u> : <b>Comment suivre une ligne ?</b> »</p> <p>« <u>Activités</u> : <b>Rechercher le fonctionnement des capteurs de détection. Écrire un programme de suivi de ligne</b>. »</p> <p>« <u>Bilan</u> : Les différents types de capteurs. <b>Utilisation des boucles, variables et des ports</b>. »</p> <p>« <u>Ressources</u> : Vidéo chariot filoguidé dans une entreprise. <b>Architecture du programme</b>. » Onglet P7_1</p>	<p><b>Phases</b> :</p> <p>Écrire un organigramme simple permettant à un mini-robot d'éviter un obstacle</p> <p>Faire des recherches sur les capteurs de détection.</p> <p>Écrire un programme de suivi de ligne (l'architecture du programme est fournie)/</p> <p><u>Ingrédients élémentaires</u> :</p> <p>Utiliser une variable</p> <p>Utiliser une boucle</p>	<p>Ce type de tâches permet de travailler sur logiciel et de d'étudier les capteurs d'un mini-robot</p> <p>Un mini-robot doit utiliser ses capteurs pour pouvoir évoluer dans son environnement</p> <p>Les capteurs permettent de connaître la position des obstacles et de les éviter.</p>

<p><b>Séquence 5 : programmer les effecteurs d'une serre (arrosage, ventilation, chauffage) en fonction de la température intérieure.</b></p> <p>« <u>Présentation/objectif</u> : Comment rendre automatique le fonctionnement d'un système ? À partir d'une serre aménagée et des effecteurs conçus et réalisés dans la séquence P2_2 [arrosage, ventilation, chauffage], les élèves doivent écrire les algorithmes et les programmes. Objectif : déclenchement d'une action par un événement, séquences d'instructions, boucles, instructions conditionnelles. Systèmes embarqués »</p> <p>« <u>Question directrice 1</u> : Comment programmer les différents effecteurs (indépendamment de la température) pour qu'ils assurent la régulation du milieu de la serre ? »</p> <p>« <u>Activités</u> : Mettre en place les effecteurs, réaliser les sous programmes de pilotage des effecteurs aération, ventilation et arrosage. »</p> <p>« <u>Bilan</u> : Sous-programme. »</p> <p>« <u>Ressources</u> : Logiciel de programmation. »</p> <p>« <u>Question directrice 2</u> : Comment programmer indépendamment les différents effecteurs en intégrant les contraintes de température du cahier des charges fonctionnel ? »</p> <p>« <u>Activités</u> : Réaliser les sous programmes de pilotage des effecteurs aération, ventilation et arrosage en intégrant le paramètre de la température. »</p> <p>« <u>Bilan</u> : Variable, conditions. »</p> <p>« <u>Ressources</u> : Logiciel de programmation. »</p>	<p><b>Phases :</b></p> <p>Mettre en place des effecteurs (aération, ventilation et arrosage)</p> <p>Réaliser les sous programmes de pilotage (indépendamment de la température)</p> <p>Intégrer le paramètre température dans les sous-programmes</p> <p>Coordonner les sous-programmes dans un programme global</p> <p>Tester le fonctionnement dans des conditions réelles</p> <p>Modifier éventuellement le programme afin de valider le cahier des charges.</p> <p><b>Ingrédients élémentaires :</b></p> <p>Utiliser une variable</p> <p>Utiliser une structure conditionnelle</p> <p>Utiliser une boucle</p> <p>Déclencher une action par un évènement</p>	
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

<p>« <u>Question directrice 3</u> : Comment synchroniser les différents effecteurs en vue de répondre au fonctionnement demandé ? Test et validation du fonctionnement. »</p> <p>« <u>Activités</u> : <b>Coordonner les différents sous-programmes en vue de générer le programme global. Mise en œuvre du système dans les conditions réelles</b> »</p> <p>« <u>Bilan</u> : Le prototype permet, après vérification du comportement attendu, de modifier le programme afin de valider le cahier des charges. Lecture d'une carte heuristique objet, <b>évènement, algorithme, variable</b>. »</p> <p>« <u>Ressources</u> : Maquette fonctionnelle de la serre, <b>logiciel de programmation</b>. » Onglet P7_3</p>		
<p><b>Séquence 6 : programmer un éclairage urbain pour l'adapter à son contexte</b></p> <p>« <u>Présentation/objectif</u> : Comment adapter un éclairage urbain dans son contexte ? [...] <b>À partir d'un courrier de la mairie (fictif) demandant de tester des scénarii permettant de réduire la consommation de l'éclairage d'une rue de la ville, les élèves vont étudier ces scénarii et repérer les leviers d'actions. Ensuite ils devront choisir parmi différents capteurs permettant la captation des éléments extérieurs (piétons, luminosité, véhicules...). Puis, ils mettent en place ces scénarii sous le logiciel Scratch connecté à une interface elle-même reliée à la maquette d'éclairage de la rue concernée. [...]. Objectifs : Mise en avant des composantes d'un programme commandé par des éléments extérieurs et de la rigueur nécessaire à la saisie. Intérêt de modéliser le fonctionnement ou le</b></p>	<p><u>Phases</u> :</p> <p>Échanger sur les différents scénarios proposés par la mairie</p> <p>Prendre connaissance du squelette du programme à compléter (blocs construits, interface graphique et habillages)</p> <p>Préparer les programmes permettant de tester les différents scénarios</p> <p>Saisir les programmes permettant de tester les différents scénarios</p>	<p>Ce type de tâches permet de mettre en avant les composantes d'un programme commandé par des éléments extérieurs.</p> <p>Permet de mettre en avant la rigueur nécessaire à la saisie d'un programme.</p> <p>Il est intéressant de bien préparer un programme complexe avant de la saisir.</p> <p>Le découpage d'un programme en fonctions permet de mieux gérer la complexité</p>

<p>rendu d'un objet pour le tester, le partager et mener une investigation. »</p> <p>« <u>Question directrice 2</u> : Comment saisir ces scénarios à l'aide du logiciel Scratch ? »</p> <p>« <u>Activités</u> : Rappel des travaux déjà réalisés sous Scratch et remémorisation des grands principes. <b>Échange sur les scénarios proposés</b> et de l'intérêt de bien préparer le programme avant de le saisir vu sa complexité. <b>Présentation du programme de base et des sous- programmes (Block sous Scratch). Préparation et saisie des scénarios sous le logiciel Scratch (interface graphique et habillages des ampoules déjà saisis). Simulation de l'état des capteurs par clic souris pour test virtuel du fonctionnement. Une fois validé, intégration des capteurs et effecteurs réels dans le programme et test sur la maquette de la salle. Ajustements si nécessaire et vérification du respect du cahier des charges. Présentation à la classe des travaux et comparaison des scénarios.</b> »</p> <p>« <u>Bilan</u> : Mise en avant de la complexité du programme comparé à l'année précédente (nombre de boucles, de variables et d'instructions) et donc de l'intérêt de créer des séquences d'instructions distinctes "s'appelant" les unes les autres et ce qui suppose une préparation sérieuse avant la saisie. Mise en avant de l'intérêt de simuler le fonctionnement avant de le tester sur le système réel (facilitation de recherche des erreurs de programmation) »</p> <p>« <u>Ressources</u> : <b>PC avec Scratch</b>, capteurs choisis. » Onglet P7_5</p>	<p>Tester virtuellement les programmes par simulation de l'état des capteurs</p> <p>Intégration des capteurs et effecteurs réels d'une maquette dans le programme</p> <p>Tester réellement le programme sur la maquette</p> <p>Ajuster si nécessaire et vérifier le respect du cahier des charges.</p> <p>Présenter à la classe et comparer des scénarios</p> <p><u>Ingrédients élémentaires</u> :</p> <p>Utiliser une boucle</p> <p>Utiliser une variable</p> <p>Créer un bloc construit (créer une fonction)</p> <p>Utiliser un bloc construit (appeler une fonction)</p>	<p>Simuler un système réel avant de la tester réellement permet de faciliter la recherche d'erreurs dans les programmes.</p>
<p><b>Séquence 7 : réaliser une application simple pour smartphone</b></p>	<p><u>Phases</u> :</p>	

<p>« <u>Présentation/objectif</u> : Comment réaliser une application Android ? Qu'est-ce qu'une application smartphone ? Réaliser une carte mentale afin d'identifier les informations importantes à communiquer et les actions intéressantes à retrouver dans cette application. Description graphique de l'application, programmation sur Appinventor de l'application, test et mise à disposition des familles. Objectifs : Comment exprimer sa pensée, définir un algorithme ? Notion de boucle et condition. »</p> <p>« <u>Question directrice 2</u> : Comment traduire nos idées sous la forme d'un programme ? »</p> <p>« <u>Activités</u> : Qu'est-ce qu'un algorithme ? Renseigner le questionnaire après avoir visionné le film. Pour chaque objet présent, sur la copie d'écran écrire ce qui est attendu (chaîne d'actions). Chercher dans une bibliothèque la boucle de programme adaptée. »</p> <p>« <u>Bilan</u> : L'algorithme, les boucles, les conditions »</p> <p>« <u>Ressources</u> : Tutoriel App inventor » Onglet P7_7</p>	<p>Réaliser une carte mentale pour identifier les éléments importants de l'application.</p> <p>Décrire graphiquement l'application</p> <p>Programmer l'application à l'aide du logiciel AppInventor (développement d'applications Android en programmation par blocs) en s'aidant de la bibliothèque.</p> <p>Tester l'application</p> <p>Mettre l'application à disposition des familles</p> <p><u>Ingrédients élémentaires</u> :</p> <p>Utiliser une boucle</p> <p>Utiliser une structure conditionnelle</p>	
<p><b>Séquence 8 : Programmer l'ouverture automatique d'un portail grâce à la technologie RFID</b></p> <p>« <u>Présentation/objectif</u> : comment utiliser la technologie RFID pour ouvrir automatiquement un portail ? [...] La finalité de la séquence étant de programmer le fonctionnement automatique d'un portail. Objectifs : Les élèves doivent être capables de compléter un programme permettant de mettre en œuvre un système automatique simple comportant plusieurs variables d'entrée et de sortie. »</p>	<p><u>Phases</u> :</p> <p>Compléter un programme (actions à réaliser au niveau du moteur et récupération d'informations venant d'un capteur) pour permettre l'ouverture automatique d'une barrière sur une maquette</p>	

<p>« <u>Question directrice 3</u> : Réalisation du programme de commande du portail. »</p> <p>« <u>Activités</u> : Réalisation de tout ou partie (ouverture) du programme de commande en définissant les actions à réaliser au niveau du moteur et en récupérant et utilisant les informations venant des capteurs de la maquette. »</p> <p>« <u>Bilan</u> : Programme réalisé puis testé sur la maquette après modification. »</p> <p>« <u>Ressources</u> : Ordinateur, réseau. Carte programmable. Maquette de portail + capteurs et actionneurs. » Onglet P7_8</p>	<p><u>Ingrédients élémentaires</u> :</p>	
<p><b>Séquence 9 : programmer une webcam pour réaliser un système de vidéosurveillance</b></p> <p>« <u>Présentation/objectif</u> : Comment surveiller une zone dans une habitation ? La séquence consiste à programmer une webcam (usb ou intégrée) pour en faire un système de vidéosurveillance. Objectifs : À la fin de la séquence, les élèves doivent être capables de mettre au point un programme simple avec un nombre limité de variables et des boucles itératives. Notions d'algorithme et de programme. Déclenchement d'une action par un événement. »</p> <p>« <u>Question directrice 1</u> : Comment fonctionne un système réel de vidéosurveillance par Webcam ? »</p> <p>« <u>Activités</u> : Les élèves (en groupe) analysent une capsule vidéo illustrant le fonctionnement d'une vidéosurveillance par Webcam et produisent un texte qui explique le fonctionnement. Les textes sont ensuite mis en commun pour montrer les différentes instructions que le « programme » doit faire. Des illustrations « graphiques »</p>	<p><u>Phases</u> :</p> <p>Analyser une capsule vidéo illustrant le fonctionnement d'un système de vidéosurveillance par webcam et produire un texte explicatif</p> <p>Mettre en commun des textes pour faire émerger les instructions du programme</p> <p>Illustrer graphiquement ces instructions afin de structurer le programme.</p> <p>Réaliser éventuellement une ébauche du programme.</p> <p>Création du programme dans scratch</p>	

<p>permettent, ensuite, de structurer différemment ce « programme ». Une ébauche de programme peut être réalisée. [...] »</p> <p>« <u>Bilan</u> : Notion d'instructions. Notion d'évènements. »</p> <p>« <u>Ressources</u> : Capsule vidéo. Vidéo Projecteur Interactif. Tableau collaboratif (pour la mise en commun des textes). »</p> <p>« <u>Question directrice 2</u> : Comment réaliser un programme de vidéosurveillance sur un PC ? »</p> <p>« <u>Activités</u> : Le programme réalisé à la séance précédente est repris. Ce programme est ensuite créé avec le logiciel Scratch (suivi d'un tutoriel de prise en main). Un problème est alors posé : comment modifier le programme pour surveiller 2 zones ? »</p> <p>« <u>Bilan</u> : Notion d'algorithme. Notion de programme. »</p> <p>« <u>Ressources</u> : Micro-ordinateur et Webcam. Logiciel Scratch. » Onglet P7_10</p>	<p>Chercher comment modifier le programme pour surveiller deux zones.</p> <p><u>Ingrédients élémentaires</u> :</p> <p>Utiliser une variable</p> <p>Utiliser une boucle</p> <p>Déclencher une action par un évènement</p>	
<p><b>Séquence 10 : programmer le déverrouillage d'une porte par un bouton poussoir ou un capteur extérieur</b></p> <p>« <u>Présentation/objectif</u> : comment commander l'ouverture d'une gâche électrique d'une serrure ? Cette séquence permet de commander une gâche électrique de serrure à partir de plusieurs technologies : d'un clavier, d'une télécommande, d'un badge RFID, d'une application sur un smartphone depuis l'extérieur de l'habitation ou d'un interrupteur situé à l'intérieur de l'habitation. Elle permet d'identifier les composants et de compléter un algorithme. Les élèves travaillent en îlot avec des composants différents. Objectifs : Capteur, actionneur, algorithme, organigramme, programme, fonction logique (ou) »</p>	<p><u>Phases</u> :</p> <p>Compléter un organigramme partiel pour permettre à un bouton poussoir de commander le déverrouillage d'une porte.</p> <p>Vérifier ses hypothèses en programmant par blocs un microcontrôleur.</p> <p>Compléter un organigramme partiel pour permettre à un bouton poussoir ou à un capteur extérieur (selon les groupes :</p>	

<p>« <u>Question directrice 2</u> : Comment commander la gâche à partir du bouton poussoir? »</p> <p>« <u>Activités</u> : À partir d'un organigramme partiel, sur Ardublock faire des hypothèses en le complétant puis les vérifier en programmant le microcontrôleur. »</p> <p>« <u>Bilan</u> : Lien algorithme, organigramme, programme. »</p> <p>« <u>Ressources</u> : Ordinateur avec Ardublock augmenté, Educaduino ou Arduino, câble USB, fils avec connecteurs Grove, carte Grove Shield, bouton-poussoir Grove, afficheur I2C Grove, LED Grove, servomoteur angulaire. »</p> <p>« <u>Question directrice 3</u> : Comment commander la gâche à partir des informations reçues par un capteur à l'extérieur ou un bouton poussoir à l'intérieur ? »</p> <p>« <u>Activités</u> : À partir d'un organigramme partiel sur Ardublock, faire des hypothèses pour le compléter puis les vérifier en programmant le microcontrôleur. »</p> <p>« <u>Bilan</u> : Lien algorithme, organigramme, programme, fonction OU. »</p> <p>« <u>Ressources</u> : Idem séance 2 et suivant les groupes ajouter un clavier, une télécommande, un capteur RFID ou un récepteur Bluetooth. » Onglet P7_12</p>	<p>carte RFID, télécommande, clavier, smartphone) de commander le déverrouillage d'une porte.</p> <p>Vérifier ses hypothèses en programmant par blocs un microcontrôleur.</p> <p><u>Ingrédients élémentaires</u> :</p> <p>Utiliser la fonction logique « ou »</p>	
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--



## A.3 Seconde – Mathématiques

### A.3.1 Programme d'enseignement PMS (MEN, 2019b)

#### A.3.1.1 Organisations didactiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Faire utiliser aux élèves des logiciels de visualisation, de calcul de simulation et de programmation (artefact/dispositif)</b></p> <p>« Utilisation de logiciels : L'utilisation de logiciels (calculatrice ou ordinateur), d'outils de visualisation et de représentation, de calcul (numérique ou formel), de simulation, de programmation développe la possibilité d'expérimenter, ouvre largement le dialogue entre l'observation et la démonstration et change profondément la nature de l'enseignement. L'utilisation régulière de ces outils peut intervenir selon trois modalités : par le professeur, en classe, avec un dispositif de visualisation collective adapté ; par les élèves, en classe, à l'occasion de la résolution d'exercices ou de problèmes ; dans le cadre du travail personnel des élèves hors du temps de classe (par exemple au CDI ou à un autre point d'accès au réseau local). » p. 3</p>	<p>Utilisation par le professeur avec un dispositif de visualisation collective</p> <p>Utilisation en classe par les élèves en classe dans le cadre de la résolution de problèmes ou d'exercices</p> <p>Utilisation par les élèves lors du travail personnel hors de la classe</p>	<p>Permet d'expérimenter</p> <p>Ouvre le dialogue entre l'observation et la démonstration</p> <p>Change profondément la nature de l'enseignement</p>
<p><b>S'appuyer sur des textes historiques pour programmer des algorithmes classiques (activité)</b></p> <p>« Histoire des mathématiques : [...] Il est possible d'étudier des textes anciens d'auteurs tels que Diophante, Euclide, Al-Khwarizmi, Fibonacci, Viète, Fermat, Descartes et mettre en évidence leurs aspects algorithmiques. » p. 6</p> <p>« Histoire des mathématiques : Les textes évoqués dans la thématique « Nombres et calculs » indiquent une préoccupation</p>	<p><u>Ingrédients</u> :</p> <p>Faire étudier des textes anciens</p> <p>Mettre en évidence leurs aspects algorithmiques</p> <p>Transformer les méthodes qu'il présente en un algorithme</p> <p>Programmer cet algorithme</p>	<p>Permet de travailler les changements de registre.</p> <p>Donne du sens au formalisme mathématique</p>

<p>algorithmique tout au long de l'Histoire. <i>Lorsqu'un texte historique a une visée algorithmique, transformer les méthodes qu'il présente en un algorithme, voire en un programme, ou inversement, est l'occasion de travailler des changements de registre qui donnent du sens au formalisme mathématique</i> » p. 15</p>		
<p><b>Traiter les notions liées à l'échantillonnage en liaison avec l'algorithmique et la programmation. (contenu)</b></p> <p>« Échantillonnage : <i>En liaison avec la partie « Algorithmique et programmation », on définit la notion d'échantillon. L'objectif est de faire percevoir, sous une forme expérimentale, la loi des grands nombres, la fluctuation d'échantillonnage et le principe de l'estimation d'une probabilité par une fréquence observée sur un échantillon.</i> » p. 14</p>	<p>Définir les notions de :</p> <ul style="list-style-type: none"> <li>- échantillon ;</li> <li>- loi des grands nombres ;</li> <li>- fluctuation d'échantillonnage ;</li> <li>- Intervalle de confiance ;</li> </ul> <p>à travers la programmation.</p>	<p>Permet de percevoir les notions sous une forme expérimentale.</p>
<p><b>Consolider les acquis du cycle 4 et introduire les notions de fonctions et de programmation textuelle (contenu)</b></p> <p>« <i>Au cycle 4, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Une consolidation des acquis du cycle 4 est proposée autour de deux idées essentielles : la notion de fonction ; la programmation comme production d'un texte dans un langage informatique</i> » p 14</p>	<p>Consolider les acquis autour de la notion de fonction et de la programmation comme production d'un texte dans un langage informatique.</p>	
<p><b>Faire utiliser le langage de programmation Python (artéfact)</b></p> <p>« <i>Un langage de programmation simple d'usage est nécessaire pour l'écriture des programmes informatiques. Le langage choisi est Python, langage interprété, concis, largement répandu et pouvant fonctionner dans une diversité d'environnements.</i> » p. 15</p>		<p>Python est simple, interprété, concis, très répandu et fonctionne dans des environnements divers.</p>

<p><b>Les problèmes traités doivent être en relation avec les mathématiques (contenu)</b></p> <p>« <i>L'algorithmique a une place naturelle dans tous les champs des mathématiques et les problèmes ainsi traités doivent être en relation avec les autres parties du programme (fonctions, géométrie, statistiques et probabilité, logique)</i> » p .15</p>		<p>L'algorithmique a une place naturelle dans tous les champs des mathématiques</p>
<p><b>Les problèmes traités doivent être en relation avec les autres disciplines ou la vie courante (contenu)</b></p> <p>« <i>[L]es problèmes ainsi traités doivent être en relation avec [...] les autres disciplines ou la vie courante.</i> » p .15</p>		
<p><b>Transmettre aux élèves l'exigence d'exactitude et de rigueur (contenu)</b></p> <p>« <i>À l'occasion de l'écriture d'algorithmes et de petits programmes, il convient de transmettre aux élèves l'exigence d'exactitude et de rigueur, et de les entraîner aux pratiques systématiques de vérification et de contrôle.</i> » p. 15</p>	<p>Entraîner les élèves aux pratiques systématiques de vérification et de contrôle lors de l'écriture d'algorithmes et de programmes</p>	
<p><b>Revisiter les notions de variable et de fonction (contenu)</b></p> <p>« <i>En programmant, les élèves revisitent les notions de variables et de fonctions sous une forme différente</i> »</p>	<p>Faire programmer les élèves.</p>	<p>Les variables et les fonctions ont les formes différentes en mathématiques et en programmation.</p>
<p><b>Faire décrire un algorithme en langage naturel (activité/artéfact)</b></p> <p>« <i>Dans le cadre de cette activité [l'algorithmique et la programmation], les élèves s'exercent à : décrire des algorithmes en langage naturel.</i> » p. 15</p>		

<p><b>Faire décrire un algorithme en langage de programmation (activité/artéfact)</b></p> <p>« Dans le cadre de cette activité [l'algorithmique et la programmation], les élèves s'exercent à : décrire des algorithmes [...] dans un langage de programmation. » p. 15</p>		
<p><b>Réaliser un algorithme à l'aide d'un programme simple écrit dans un langage de programmation textuel (activité/artéfact)</b></p> <p>« Dans le cadre de cette activité [l'algorithmique et la programmation], les élèves s'exercent à : [...] en réaliser quelques-uns [des algorithmes] à l'aide d'un programme simple écrit dans un langage de programmation textuel. » p. 15</p>		
<p><b>Faire interpréter un algorithme plus complexe (activité)</b></p> <p>« Dans le cadre de cette activité [l'algorithmique et la programmation], les élèves s'exercent à : [...] interpréter [...] des algorithmes plus complexes. » p. 15</p> <p>« Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme. » p. 15</p>		
<p><b>Faire compléter un algorithme plus complexe (activité)</b></p> <p>« Dans le cadre de cette activité [l'algorithmique et la programmation], les élèves s'exercent à : [...] compléter [...] des algorithmes plus complexes. » p. 15</p> <p>« Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme. » p. 15</p>		
<p><b>Faire modifier un algorithme plus complexe (activité)</b></p>		

<p>« Dans le cadre de cette activité [l'algorithmique et la programmation], les élèves s'exercent à : [...] modifier des algorithmes plus complexes. » p. 15</p> <p>« Dans des cas plus complexes : lire, comprendre, modifier ou compléter un algorithme ou un programme. » p. 15</p>		
<p><b>Faire passer du langage naturel ou Python et inversement</b> (activité)</p> <p>« Les élèves sont entraînés à passer du langage naturel à Python et inversement. » p. 15</p>	En s'entraînant	
<p><b>Faire choisir ou déterminer le type d'une variable</b> (contenu/activité)</p> <p>« Capacités attendues : Choisir ou déterminer le type d'une variable (entier, flottant ou chaîne de caractères). » p. 15</p>		
<p><b>Faire concevoir et écrire une instruction d'affectation</b> (contenu/activité)</p> <p>« Concevoir et écrire une instruction d'affectation » p. 15</p>		
<p><b>Faire concevoir et écrire une séquence d'instructions</b> (contenu/activité)</p> <p>« Concevoir et écrire [...], une séquence d'instructions. » p. 15</p>		
<p><b>Faire concevoir et écrire une instruction conditionnelle</b> (contenu/activité)</p> <p>« Concevoir et écrire [...] une instruction conditionnelle. » p. 15</p>		
<p><b>Écrire une formule combinant des variables</b> (contenu/activité)</p>		

« <b>Écrire une formule permettant un calcul combinant des variables.</b> » p. 15		
<b>Faire programmer une boucle for ou non-for dans un cas simple (contenu)</b> « <b>Programmer, dans des cas simples, une boucle for, une boucle non for.</b> » p. 15		
<b>Faire écrire des fonctions simples (contenu/activité)</b> « Capacités attendues : <b>Écrire des fonctions simples</b> » p. 16		
<b>Faire lire, comprendre, modifier, compléter des fonctions complexes (contenu/activité)</b> « <b>Lire, comprendre, modifier, compléter des fonctions plus complexes.</b> » p. 16		
<b>Faire appeler une fonction (contenu/activité)</b> « <b>Appeler une fonction.</b> » p. 16		
<b>Faire déterminer un encadrement de <math>\sqrt{2}</math> (contenu/activité)</b> « [Partie nombres et calculs] Exemple d'algorithme : <b>Déterminer par balayage un encadrement de <math>\sqrt{2}</math> d'amplitude inférieure ou égale à <math>10^{-n}</math></b> » p. 6	Procéder par balayage	
<b>Faire déterminer la première puissance d'un nombre supérieure à une valeur. (contenu/activité)</b> « [Partie nombres et calculs] Exemple d'algorithme : <b>Déterminer la première puissance d'un nombre positif donné supérieure ou inférieure à une valeur donnée.</b> » p. 8		

<p><b>Faire étudier l'alignement de trois points du plan (contenu/activité)</b></p> <p>« [Partie géométrie] Exemples d'algorithme : Étudier l'alignement de trois points dans le plan. » p. 10</p>		
<p><b>Faire déterminer une équation de droite passant par deux points (contenu/activité)</b></p> <p>« [Partie Géométrie] Exemples d'algorithme : [...] Déterminer une équation de droite passant par deux points donnés. » p 10</p>		
<p><b>Faire programmer des fonctions mathématiques en Python (contenu/activité)</b></p> <p>« [Partie fonctions] Les outils numériques sont mis à profit : [...] Python, le tableur ou la calculatrice, pour mettre en évidence l'aspect de programme de calcul [des fonctions]. » p. 11</p>		<p>Permet de mettre en évidence l'aspect programme de calcul des fonctions</p>
<p><b>Faire approximer un extremum de fonction (contenu/activité)</b></p> <p>« [Partie fonctions] Exemples d'algorithme : Pour une fonction dont le tableau de variations est donné, algorithmes d'approximation numérique d'un extremum (balayage, dichotomie). » p. 12</p>	<p>Méthode par balayage</p> <p>Méthode par dichotomie</p>	
<p><b>Faire calculer la longueur d'une portion de courbe représentative (contenu/activité)</b></p> <p>« [Partie fonctions] Exemples d'algorithme : Algorithme de calcul approché de longueur d'une portion de courbe représentative de fonction. » p. 12</p>		
<p><b>Faire comprendre une fonction Python renvoyant les indicateurs d'une série statistique (contenu/activité)</b></p>	<p>À partir de données réels</p> <p>À partir de données issues d'une simulation</p>	

<p>« [Partie statistiques et probabilités] Capacités attendues : <b>Pour des données réelles ou issues d'une simulation</b>, lire et comprendre une fonction écrite en Python renvoyant la moyenne <math>m</math>, l'écart type <math>s</math>, et la proportion d'éléments appartenant à <math>[m - 2s ; m + 2s]</math>. p. 13</p> <p>« Lire et comprendre une fonction renvoyant une moyenne, un écart type. <b>Aucune connaissance sur les listes n'est exigée.</b> » p. 16</p>		
<p><b>Faire comprendre une fonction Python renvoyant le nombre de succès dans un échantillon pour une expérience à deux issues.</b> (contenu/activité)</p> <p>« [Partie statistiques et probabilités] Capacités attendues : Lire et comprendre une fonction Python renvoyant le nombre ou la fréquence de succès dans un échantillon de taille <math>n</math> pour une expérience aléatoire à deux issues. » p. 14</p>		
<p><b>Faire observer la loi des grands nombres à l'aide d'une simulation</b> (contenu/activité)</p> <p>« [Partie statistiques et probabilités] Capacités attendues : [...] Observer la loi des grands nombres à l'aide d'une simulation <b>sur Python ou tableur.</b> » p. 14</p> <p>« <b>Écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire, d'une répétition d'expériences aléatoires indépendantes.</b> » p. 16</p>	<p>Simulation sur Python. En écrivant une fonction renvoyant le résultat d'une expérience aléatoire.</p> <p>Simulation sur tableur</p>	
<p><b>Faire simuler plusieurs échantillons et calculer la proportion de cas où la proportion est dans l'intervalle de confiance</b> (contenu/activité)</p> <p>« [Partie statistiques et probabilités] Capacités attendues : [...] Simuler <math>N</math> échantillons de taille <math>n</math> d'une expérience aléatoire à deux</p>	<p>En écrivant une fonction renvoyant le résultat d'une répétition d'expériences aléatoires indépendantes.</p>	



<p>issues. Si <math>p</math> est la probabilité d'une issue et <math>f</math> sa fréquence observée dans un échantillon, calculer la proportion des cas où l'écart entre <math>p</math> et <math>f</math> est inférieur ou égal à <math>\frac{1}{\sqrt{n}}</math>. » p. 14</p>		
<p>« <i>Écrire des fonctions renvoyant le résultat numérique d'une expérience aléatoire, d'une répétition d'expériences aléatoires indépendantes.</i> » p. 16</p>		

## A.3.2 Ressource d'accompagnement R1MS (MEN, 2019a)

### A.3.2.1 Organisations didactiques

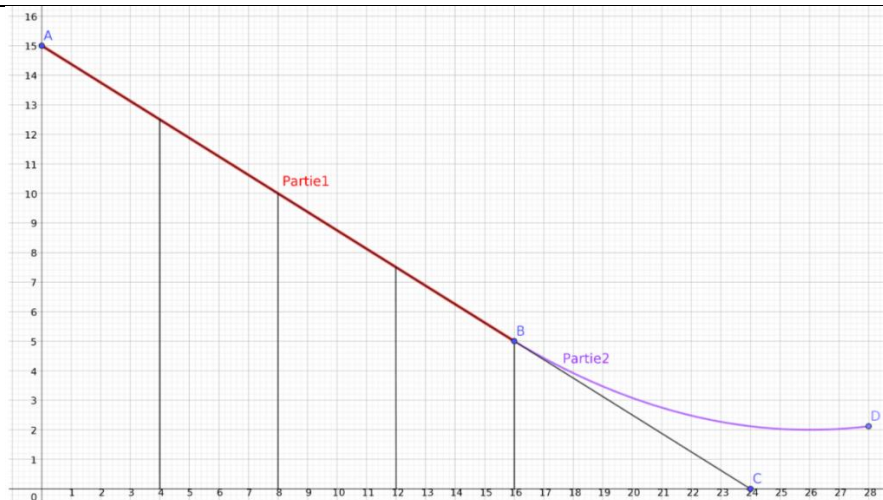
Type de tâches	Technique(s)	Technologie(s)
<p><b>Faire utiliser le langage Python aux élèves (artéfact)</b></p> <p>« <i>Le langage de programmation choisi au lycée (voies générale et technologique) est le langage Python. Sa syntaxe est claire et facilite le passage de la programmation par blocs à la programmation dans un langage textuel. Il est libre et gratuit et fonctionne sur des systèmes d'exploitation variés. De nombreuses bibliothèques sont disponibles (traitement de données numériques, d'images, de sons, etc.), qui peuvent être utiles aux mathématiques, mais aussi à d'autres disciplines.</i> » p. 2</p>		<p>Syntaxe claire</p> <p>Facilite le passage de la programmation par blocs à la programmation textuelle.</p> <p>Est libre et gratuit</p> <p>Est multiplateforme</p> <p>Possède de nombreuses bibliothèques</p>
<p><b>Ne pas exiger de technicité au niveau de la syntaxe Python (contenu)</b></p> <p>« <i>La mise en place d'activités relevant de l'algorithmique et la programmation ne doit pas faire perdre de vue leur objectif de formation mathématique. C'est pourquoi aucune technicité au niveau de la syntaxe du langage Python n'est attendu. La compréhension est privilégiée par rapport à l'optimisation du code des programmes.</i> » p. 2</p>		<p>L'objectif principal est la formation mathématique</p> <p>La compréhension est privilégiée par rapport à l'optimisation du code.</p>
<p><b>Enseigner la notion de fonction (contenu/activité)</b></p> <p>« <i>L'apprentissage de l'algorithmique et de la programmation a débuté au cycle 4 sur le logiciel Scratch de manière à faire acquérir aux élèves les premiers concepts de programmation. [...]. En revanche, la notion de fonction informatique, qui ne figure pas au</i></p>	<p>Présenter les fonctions comme un sous-programme indépendant</p> <p>Ne pas faire appel à des saisies au clavier ni à des affichages à l'écran mais gérer les entrées-</p>	<p>Permet de traiter séparément les différentes parties d'un programme.</p> <p>Permet de prendre de bonnes habitudes de programmation</p>

<p><i>programme du collège, est une nouveauté du programme de seconde.</i> » p. 2</p> <p>« <i>Une fonction informatique est une portion de code représentant un sous-programme effectuant une tâche ou un calcul relativement indépendant du reste du programme. Le découpage d'un programme en sous-programmes plus petits et réutilisables permet d'en traiter séparément différentes parties. [...] L'écriture et l'utilisation de fonctions informatiques permettent également de prendre très tôt de bonnes habitudes de programmation. Ainsi, au niveau des entrées/ sorties, l'usage des fonctions évite de faire appel à des instructions du type ( int(input("a ?")) ) qui obscurcissent le code. Les entrées apparaissent comme paramètres de la fonction et l'exécution de celle-ci fournit les sorties souhaitées sans avoir recours à l'instruction print dont l'utilisation n'est nécessaire que si on souhaite afficher un résultat à l'intérieur d'une fonction ou dans un script.</i> » p. 2</p>	<p>sorties à l'aide des paramètres et du retour d'une fonction</p>	
<p><b>Faire le lien avec la notion de fonction mathématique (contenu)</b></p> <p>« <i>En classe de seconde, l'introduction de la notion de fonction informatique est mise en lien avec la notion de fonction mathématique</i> introduite dès le collège et dont l'étude se prolonge » p. 2</p>		
<p><b>Confier l'écriture de sous-programme à des groupes différents (dispositif)</b></p> <p>« <i>L'écriture de sous-programmes peut être confiée à différents groupes d'élèves, favorisant le travail collaboratif et la mise en place d'activités de projet.</i> » p. 2</p>		<p>Favorise le travail collaboratif</p> <p>Favorise la mise en place d'activités de projet.</p>

<p><b>Expliquer une programme (activité)</b></p> <p>« l'enseignant peut, s'il le souhaite, soumettre à ses élèves, chacune d'elles [des suggestions] correspondant à un savoir-faire de l'un des types suivants : <b>Expliquer</b> : il est alors demandé d'interpréter une ou plusieurs lignes d'un programme informatique fourni [...] » p. 1</p>	<p>Interpréter une ou plusieurs lignes d'un programme fourni.</p>	
<p><b>Modifier un programme (activité)</b></p> <p>« <b>Modifier</b> : il est alors demandé de changer quelques lignes de code pour répondre à une nouvelle question, de corriger une erreur ou d'améliorer un programme. » p. 1</p>	<p>Changer quelques lignes de code pour répondre à une nouvelle question</p>	
	<p>Corriger une erreur</p>	
	<p>Améliorer un programme</p>	
<p><b>Tester un programme (activité)</b></p> <p>« <b>Tester</b> : il est alors demandé d'exécuter un programme et d'en interpréter les sorties. » p. 1</p>	<p>Exécuter un programme et en interpréter les sorties.</p>	
<p><b>Compléter un programme (activité)</b></p> <p>« <b>Compléter</b> : il est alors demandé de compléter le programme pour manifester sa compréhension des notions mathématiques ou informatiques mises en jeu. » p. 2</p>		<p>Permet de manifester sa compréhension des notions (mathématiques ou informatiques) mises en jeu</p>
<p><b>Écrire un programme (activité)</b></p> <p>« <b>Écrire</b> : il est alors demandé de programmer une fonction ou une suite d'instructions répondant à un problème donné. » p. 2</p>	<p>Programmer une fonction ou une suite d'instruction répondant à un problème donné.</p>	

### A.3.2.2 Organisations informatiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Activité 1 : étudier un tremplin de ski</b></p> <p>(MEN 2019 - Activité 1 : Étude d'un tremplin de ski – Éduscol - <a href="https://bit.ly/3kphsfj">https://bit.ly/3kphsfj</a>)</p> <p>« <i>L'étude de la structure d'un tremplin de saut à ski permet de réexploiter les notions d'alignement, d'équations droites et de fonction. On étudie la forme du tremplin, la hauteur des supports et la longueur de la piste.</i> »</p> <p>« <i>Situation : Une station de sports d'hiver souhaite construire un tremplin de saut à ski dont la structure est en acier. Le profil de la piste est représenté dans un plan rapporté à un repère orthonormal <math>(O, I, J)</math>. L'unité est le mètre. La piste d'envol du tremplin est constituée de 2 parties : une partie représentée par un segment <math>[AB]</math> ; une partie représentée par un arc de parabole <math>BD</math>. Il faut qu'il y ait raccordement au point <math>B</math>.</i> »</p>		<p>Permet réexploiter les notions d'alignement, d'équations de droites et de fonction mathématique.</p>



### Partie 1 : étude la partie [AB]

« La première partie est portée par un support rectiligne passant par le point A de coordonnées (0; 15) et le point C de coordonnées (24; 0). Le raccordement entre les deux parties doit se faire au point B de coordonnées (16; 5). Alignement de trois points. La fonction `alignement` permet de vérifier si trois points de coordonnées entières sont alignés. Elle prend les coordonnées entières des trois points en paramètres et renvoie un booléen. »

#### Question 1 :

« Compléter un programme : Le programme précédent étant fourni [...] demander aux élèves de compléter les lignes 2,3,4 et 6 de la fonction. »

### Question 1 : Technique 1

```
def alignement(x1,y1,x2,y2,x3,y3)
    determinant=(x2-x1)*(y3-y1)
                -(y2-y1)*(x3-x1)
    if determinant==0:
        rep=True
    else:
        rep=False
    return rep
print(alignement(0,15,24,0,16,5))
```

#### Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x3

Utiliser la valeur d'une variable x2

Manipuler une variable de type « int » (entier)

```
def alignement(x1,y1,x2,y2,x3,y3):
    determinant= ...
    if ...
        rep= ...
    else:
        rep= ...
    return rep
```

Tester la fonction précédente pour vérifier l'alignement des points A,B et C. »

Manipuler une variable de type « bool » (booléen)

Soustraire des nombres x5

Multiplier des nombres x2

Utiliser une structure conditionnelle à deux branches

Tester une égalité sur des nombres

Définir une fonction avec paramètres et retour

Utiliser une fonction avec paramètres et retour

Utiliser une fonction avec paramètres et sans retour

Question 1 : Technique 2

```
def alignement(x1,y1,x2,y2,x3,y3):
    determinant=(x2-x1)*(y3-y1)\
                -(y2-y1)*(x3-x1)
    rep=(determinant==0)
    return rep
print(alignement(0,15,24,0,16,5))
```

Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x2

Utiliser la valeur d'une variable x2

	<p>Manipuler une variable de type « int » (entier)</p> <p>Manipuler une variable de type « bool » (booléen)</p> <p>Soustraire des nombres x5</p> <p>Multiplier des nombres x2</p> <p>Tester une égalité sur des nombres</p> <p>Définir une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et sans retour</p>	
<p><b>Partie 2 : Hauteur des poteaux</b></p> <p>« La première partie du tremplin est supportée par des poteaux placés tous les quatre mètres (abscisses : 4, 8 et 12). <b>On souhaite déterminer la hauteur de ses poteaux.</b> »</p> <p>« Équation de droite. La fonction <code>equation</code> donne les coefficients <math>a</math> et <math>b</math> de l'équation réduite <math>y = ax + b</math> d'une droite passant par deux points d'abscisses différentes. Elle prend les coordonnées des deux points en paramètres et renvoie les valeurs de <math>a</math> et <math>b</math>. »</p> <p><u>Question 1 :</u></p>	<p><b>Question 1 :</b></p> <pre>def equation(xA, yA, xB, yB):     assert xB!=xA, "il faut "\         +"donner deux points"\         +"d'abscisses distinctes"     a=(yB-yA)/(xB-xA)     b=yA-a*xA     return a,b  a,b = equation(0,15,16,5) print(a,b)</pre> <p><b>Ingrédients élémentaires :</b></p>	



« Compléter un programme : [...] demander aux élèves de compléter les lignes 3 et 4 de la fonction equation.

```
def equation(xA, yA, xB, yB):  
    assert xB!=xA, "il faut \  
    +\"donner deux points\"\  
    +\"d'abscisses distinctes\"  
    a= ...  
    b= ...  
    return a,b
```

```
a,b = equation(0,15,16,5)  
print(a,b)
```

Tester la fonction equation afin d'obtenir le coefficient directeur et l'ordonnée à l'origine de la droite (AB). »

« Image par une fonction affine. La fonction suivante détermine l'image d'un réel  $x$  par une fonction affine d'équation  $g(x) = ax + b$ . Elle prend les valeurs de  $a$  et  $b$  et celle de  $x$  en paramètres et renvoie l'image de  $x$ . On l'appelle ensuite pour  $x = 4$ , pour  $x = 8$ , et pour  $x = 12$  avec les paramètres correspondant à l'équation de la droite de la première partie du tremplin pour déterminer la hauteur des poteaux. »

Question 2 :

Écrire un programme : Écrire une fonction fonctionAffine qui prend  $a, b, x$  en paramètres et qui renvoie  $ax + b$ .

Tester la fonction fonctionAffine afin de calculer les hauteurs des poteaux. »

Modifier la valeur d'une variable (affectation) x2

Utiliser la valeur d'une variable x2

Manipuler une variable de type « int » (entier) x2

Tester une égalité sur des nombres

Soustraire des nombres x3

Multiplier des nombres

Diviser des nombres

Définir une fonction avec paramètres et retours

Utiliser une fonction avec paramètres et retours

Utiliser une fonction avec paramètres et sans retour

Question 2 :

```
def fonctionAffine(a,b,x):  
    y=a*x+b  
    return(y)
```

```
a,b = equation(0,15,16,5)  
print("Hauteur du premier poteau =",\  
      fonctionAffine(a,b,4), "m")  
print("Hauteur du seconde poteau =",\  
      fonctionAffine(a,b,8), "m")  
print("Hauteur du troisième poteau =",\  
      fonctionAffine(a,b,12), "m")
```

	<p><u>Ingrédients élémentaires :</u></p> <p>Modifier la valeur d'une variable (affectation) x2</p> <p>Utiliser la valeur d'une variable x7</p> <p>Manipuler une variable de type « int » (entier) x3</p> <p>Additionner des nombres</p> <p>Multiplier des nombres</p> <p>Définir une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et retours x4</p> <p>Utiliser une fonction avec paramètres et sans retour</p>	
<p><b><u>Partie 3 : Longueur de la piste [AB]</u></b></p> <p>« Afin de placer le revêtement sur la piste, on souhaite connaître sa longueur totale. La fonction suivante calcule la longueur d'un segment [AB] dans un repère orthonormal (O,I,J). Elle prend les coordonnées des points A et B en paramètres et renvoie la longueur de la piste en unités de longueur. »</p> <p><u>Question 1 :</u></p>	<p><u>Question 1 :</u></p> <pre>import math def longueur(x1,y1,x2,y2):     d=math.sqrt((x1-x2)**2 \                 +(y1-y2)**2)     return d print("La longueur AB est égale à" \       ,longueur(0,15,16,5),"mètres.")</pre> <p><u>Ingrédients élémentaires :</u></p> <p>Importer une bibliothèque</p>	

<p>« <i>Écrire un programme. Écrire une fonction longueur qui prend en paramètres les coordonnées <math>x_1, y_1, x_2, y_2</math> des extrémités d'un segment et renvoie sa longueur.</i></p> <p><i>Tester la fonction longueur afin d'estimer la longueur de la piste [AB].</i> »</p>	<p>Modifier la valeur d'une variable (affectation)</p> <p>Utiliser la valeur d'une variable</p> <p>Manipuler une variable de type « int » (entier)</p> <p>Additionner des nombres</p> <p>Soustraire des nombres x2</p> <p>Élever des nombres au carré x2</p> <p>Définir une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et retours x2</p> <p>Utiliser une fonction avec paramètres et sans retour</p>	
<p><b>Partie 4 : Étude de la partie BD</b></p> <p>« <i>La deuxième partie du tremplin est un arc de parabole d'équation <math>f(x) = 0,03x^2 - 1,56x + 22,28</math>. L'abscisse du point D est égale à 28. Coordonnées de D. La fonction suivante calcule l'image d'un réel x par la fonction f. Celle-ci permet de déterminer l'ordonnée de D.</i> »</p> <p><u>Question 1 :</u></p> <p>« <i>Écrire un programme : Écrire une fonction f prenant en paramètre un nombre x et renvoyant son image par la fonction (mathématique) f.</i></p>	<p><u>Question 1 :</u></p> <pre>def f(x):     return 0.03*x**2-1.56*x+22.28 print(f(28))</pre> <p><u>Ingrédients élémentaires :</u></p> <p>Définir une fonction avec paramètres et retour</p> <p>Additionner des nombres</p> <p>Soustraire des nombres</p> <p>Multiplier des nombres x2</p>	

Tester la fonction pour calculer l'ordonnée de D. Interpréter le résultat (expliquer le nombre de décimales). »

« Longueur de l'arc BD. Pour calculer la longueur totale du revêtement, on souhaite évaluer également la longueur de la deuxième partie de la piste. Pour cela, on divise l'intervalle [16 ; 28] en  $n$  intervalles de longueur  $(28 - 16)n = 12n$ . On place alors les points  $A_0, A_1, A_2, \dots, A_n$  de la courbe d'abscisses respectives : 16 ;  $16 + 12n$  ;  $16 + 2 \times 12n$  ; ... ; 28. Pour calculer une valeur approchée de notre portion de courbe, il suffit alors de faire la somme des longueurs des segments  $[A_i; A_{i+1}]$ . »

**Question 2 :**

```
def LongueurArc(n, borneInf, borneSup) :  
    longueurTotale=0  
    x1=borneInf  
    pas=(borneSup-borneInf)/n  
    x2=x1+pas  
    for i in range(n):  
        y1=f(x1)  
        y2=f(x2)  
        longueurTotale= ...  
        x1= ...  
        x2= ...  
    return longueurTotale
```

« Expliquer un programme Expliquer les lignes 2 à 8.

Compléter un programme Le programme précédent étant fourni [...]  
demander aux élèves de compléter la ligne 9.

Compléter un programme. Le programme précédent étant fourni [...],  
demander aux élèves de compléter les lignes 10 et 11 pour passer au segment suivant.

Élever des nombres au carré

Utiliser une fonction avec paramètres et retours

Utiliser une fonction avec paramètres et sans retour

**Question 2 :**

```
def LongueurArc(n, borneInf, borneSup) :  
    longueurTotale=0  
    x1=borneInf  
    pas=(borneSup-borneInf)/n  
    x2=x1+pas  
    for i in range(n):  
        y1=f(x1)  
        y2=f(x2)  
        longueurTotale=longueurTotale\  
        +longueur(x1, y1, x2, y2)  
        x1=x1+pas  
        x2=x2+pas  
    return longueurTotale  
print(LongueurArc(100, 16, 28))
```

**Ingrédients élémentaires :**

Modifier la valeur d'une variable (affectation) x9

Utiliser la valeur d'une variable x14

Manipuler une variable de type « float » (flottant) x4

Additionner des nombres x4

Soustraire des nombres

<p>Tester la fonction longueurArc pour calculer la longueur de l'arc BD avec différentes valeurs de n (nombre d'intervalles). »</p>	<p>Diviser des nombres</p> <p>Utiliser une boucle for sans utilisation de la variable de boucle</p> <p>Définir une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et retour x4</p> <p>Utiliser une fonction avec paramètres et sans retour</p>	
<p><b>Activité 2 : encadrer <math>\sqrt{2}</math> par balayage</b></p> <p>(MEN 2019 - Activité 2 : Encadrement de <math>\sqrt{2}</math> par balayage – Éduscol - <a href="https://bit.ly/3m6XmXE">https://bit.ly/3m6XmXE</a>)</p> <p>« Présentation de l'activité. <b>Références au programme : Seconde : déterminer par balayage un encadrement de <math>\sqrt{2}</math> d'amplitude inférieure ou égale à <math>10^{-n}</math>.</b>[...] Description : <b>approximation de <math>\sqrt{2}</math>, par l'étude de la fonction <math>x \rightarrow x^2 - 2</math> et la recherche d'un zéro de cette fonction par balayage.</b> »</p>		<p>Permet de travailler l'élément du programme : déterminer par balayage un encadrement de <math>\sqrt{2}</math> d'amplitude inférieure ou égale à <math>10^{-n}</math></p>
<p><b>Partie 1 : Valeur approchée par défaut</b></p> <p>« Représentation de la fonction <math>x \rightarrow x^2 - 2</math> sur l'intervalle <math>[0 ; 2]</math> .</p> <p>La fonction représentation trace la représentation graphique d'une fonction f. Elle utilise une subdivision de <math>[a,b]</math> en N intervalles de même longueur. Elle prend comme paramètres : les bornes a et b de l'intervalle <math>[a; b]</math>, le nombre N d'intervalles de la subdivision. »</p>	<p><u>Question 1 :</u></p>	

### Question 1 :

« Expliquer un programme. À quoi correspond la variable pas ? Comment est-elle calculée ? Expliquer la ligne 7. Quelle est la première valeur de x dans la boucle de la ligne 7 ? Quelle est la dernière valeur de x dans la boucle de la ligne 7 ? Comment obtenir une valeur approchée de  $\sqrt{2}$  à l'aide du graphique ?

```
import matplotlib.pyplot as plt
def f(x):
    return ...
def representation(a,b,N):
    pas = ...
    for i in range(N+1):
        x = ...
        plt.plot(x, f(x), 'b.')
representation(0,2,100)
plt.plot([0,2],[0,0], 'r--')
plt.show()
```

Compléter un programme. Le programme précédent étant fourni [...] demander aux élèves de compléter les lignes 2, 5 et 7.

Tester la fonction representation pour l'intervalle [0;2] divisé en 100 parties. »

« Approximation de  $\sqrt{2}$ . Le programme suivant donne une approximation par défaut de  $\sqrt{2}$  avec une erreur inférieure à  $10^{-2}$ . »

### Question 2 :

« Expliquer un programme. Expliquer la valeur de N ligne 3.

```
import matplotlib.pyplot as plt
def f(x):
    return x**2-2
def representation(a,b,N):
    pas = (b-a)/N
    for i in range(N+1):
        x = a+i*pas
        plt.plot(x, f(x), 'b.')
representation(0,2,100)
plt.plot([0,2],[0,0], 'r--')
plt.show()
```

### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x2

Utiliser la valeur d'une variable x3

Manipuler une variable de type « float » (flottant) x2

Additionner des nombres x2

Soustraire des nombres

Multiplier des nombres

Diviser des nombres

Élever des nombres au carré

Utiliser une boucle for avec utilisation de la variable de boucle

```

def f(x):
    return x**2-2
a = 0
b = 2
N = 200
pas = (b-a)/N
estimation = ...
while ...
    estimation = ...
print("Estimation de racine de"\
      +"racine de 2:",estimation)
print("Estimation de racine de"\
      +"2 au carré:",estimation**2)

```

**Compléter un programme. Le programme précédent étant fourni [...], demander aux élèves de compléter les lignes 6, 7 et 8. »**

Définir une fonction avec paramètres et retour

Définir une fonction avec paramètres et sans retour

Utiliser une fonction avec paramètres et retour

Utiliser une fonction avec paramètres sans retour x4

Utiliser une fonction sans paramètres et sans retour

Question 2 :

```

def f(x):
    return x**2-2
a = 0
b = 2
N = 200
pas = (b-a)/N
estimation = a
while f(estimation)<0:
    estimation = estimation + pas
print("Estimation de racine de"\
      +"racine de 2:",estimation)
print("Estimation de racine de"\
      +"2 au carré:",estimation**2)

```

Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x6

Utiliser la valeur d'une variable x9

	<p>Manipuler une variable de type « int » (entier) x3</p> <p>Manipuler une variable de type « float » (flottant) x2</p> <p>Additionner des nombres</p> <p>Soustraire des nombres x2</p> <p>Diviser des nombres</p> <p>Élever des nombres au carré x2</p> <p>Utiliser une boucle non-for</p> <p>Tester une inégalité sur des nombres</p> <p>Utiliser une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et sans retour x2</p>	
<p><b>Partie 2 : Encadrement d'amplitude <math>10^{-2}</math></b></p> <p>« On cherche à déterminer un encadrement de <math>\sqrt{2}</math> à <math>10^{-2}</math> près par balayage. La valeur approchée précédemment obtenue étant la plus grande valeur de <math>x</math> pour laquelle <math>f(x) \leq 0</math>, si on ajoute à <math>x</math> la valeur du pas (ici <math>10^{-2}</math>), on a <math>f(x + 10^{-2}) &gt; 0</math> et donc <math>x \leq \sqrt{2} &lt; x + 10^{-2}</math>, ce qui donne un encadrement de <math>\sqrt{2}</math>. »</p> <p>Question 1 :</p> <p>« Expliquer un programme. Que fait l'instruction de la ligne 12 ?</p>	<p>Question 1 :</p>	



```

def f(x):
    return x**2-2
a = 0
b = 2
N = 200
pas = (b-a)/N
x = a
while f(x)<0:
    x = x + pas
estimationDroite = ...
estimationGauche = x - pas
print("Encadrement de racine"\
      +"de 2 à 10**{-2} près:"\
      ,estimationGauche\
      ,estimationDroite)

```

**Compléter un programme. Le programme précédent étant fourni [...], demander aux élèves de compléter la ligne 12 »**

```

def f(x):
    return x**2-2
a = 0
b = 2
N = 200
pas = (b-a)/N
x = a
while f(x)<0:
    x = x + pas
estimationDroite = x
estimationGauche = x - pas
print("Encadrement de racine"\
      +"de 2 à 10**{-2} près:"\
      ,estimationGauche\
      ,estimationDroite)

```

#### Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x8

Utiliser la valeur d'une variable x11

Manipuler une variable de type « int » (entier) x3

Manipuler une variable de type « float » (flottant) x4

Additionner des nombres

Soustraire des nombres x2

Diviser des nombres

Élever des nombres au carré

	<p>Utiliser une boucle non-for</p> <p>Tester une inégalité sur des nombres</p> <p>Utiliser une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et sans retour</p>	
<p><b>Activité 3 : trouver le nombre de chiffres d'un nombre donné</b></p> <p>(MEN 2019 - Activité 3 - Nombre de chiffres d'un nombre donné - Éduscol - <a href="https://bit.ly/37wmpiQ">https://bit.ly/37wmpiQ</a>)</p> <p>« <i>Présentation de l'activité. Référence au programme : Seconde: Déterminer la première puissance d'un nombre positif donné supérieure ou inférieure à une valeur donnée. Description : activité de décompte du nombre de chiffres d'un nombre fixé permettant de faire travailler la notion de puissance et de faire réfléchir sur la notion de base.</i> »</p>		<p>Permet de travailler un élément du programme : déterminer la première puissance d'un nombre positif donnée supérieur ou inférieur à une valeur donnée.</p> <p>Permet de travailler la notion de puissance et celle de base.</p>
<p><b>Partie 1 : cas général</b></p> <p>« <i>Dans cette partie, on cherche à écrire un programme qui détermine le nombre de chiffres de l'écriture décimale d'un entier naturel. Pour cela, on remarque que <math>10^n</math> a exactement <math>n + 1</math> chiffres.[...] La fonction nombreChiffres prend en paramètre un entier naturel <math>a</math> non nul et renvoie l'exposant de la plus petite puissance de 10 (<math>\geq 1</math>) strictement supérieure au nombre <math>a</math> (cela correspond au nombre de chiffres de <math>a</math>).</i> »</p> <p><u>Question 1 :</u></p>	<p><u>Question 1 :</u></p> <pre>def nombreChiffres(a):     n = 0     while 10**n &lt;= a:         n = n+1     return n print(nombreChiffres(109))</pre> <p><u>Ingrédients élémentaires :</u></p>	

<p>« Expliquer un programme. À quelle condition s'arrête la boucle while ? À quoi servent les lignes 2 et 4 ? »</p> <p>Compléter un programme. Le programme précédent étant fourni [...] demander aux élèves de compléter les lignes 2, 3 et 4.</p> <pre>def nombreChiffres(a):     n = ...     while ...         n = ...     return n print(nombreChiffres(109))</pre> <p>Écrire un programme. Écrire une fonction prenant en entrée un entier a et renvoyant l'exposant de la plus petite puissance de 10 strictement supérieure à a. »</p>	<p>Modifier la valeur d'une variable (affectation) x2</p> <p>Utiliser la valeur d'une variable x2</p> <p>Additionner des nombres</p> <p>Élever des nombres à la puissance n</p> <p>Manipuler une variable de type « int » (entier)</p> <p>Utiliser une boucle non-for</p> <p>Tester une inégalité sur des nombres</p> <p>Définir une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et sans retour</p>	
<p><b>Partie 2 : cas des puissance de 2</b></p> <p>« Comme application, on cherche maintenant à connaître le nombre de chiffres des puissances de 2 . »</p> <p><u>Question 1 :</u></p>	<p><u>Question 1 :</u></p> <pre>print(nombreChiffres(2**10), \       nombreChiffres(2**64), \       nombreChiffres(2**100))</pre> <p><u>Ingrédients élémentaires :</u></p>	

<p>« Tester un programme. En utilisant le programme <code>nombreChiffres</code>, déterminer le nombre de chiffres des nombres <math>2^{10}</math>, <math>2^{64}</math> et <math>2^{100}</math>. »</p>	<p>Utiliser une fonction avec paramètres et a avec retour x3</p> <p>Utiliser une fonction avec paramètres et sans retour</p> <p>Élever des nombres à la puissance n x3</p>	
<p><b>Activité 4 : programmer un jeu de dé</b></p> <p>(MEN 2019 - Activité 4 - Jeu de dés - Éduscol - <a href="https://bit.ly/2Thq4c1">https://bit.ly/2Thq4c1</a>)</p> <p>« Présentation de l'activité. Références au programme :</p> <p>Seconde : Lire et comprendre une fonction Python renvoyant le nombre ou la fréquence de succès dans un échantillon de taille n pour une expérience aléatoire à deux issues.</p> <p>Observer la loi des grands nombres à l'aide d'une simulation sur Python.</p> <p>Simuler N échantillons de taille n d'une expérience aléatoire à deux issues. Si p est la probabilité d'une issue et f sa fréquence observée dans un échantillon, calculer la proportion des cas où l'écart entre p et f est inférieur ou égal à <math>\sqrt{n}</math>.</p> <p>Description de l'activité : cette activité propose un jeu de dés qui débouche sur deux issues (gagner ou perdre). Dans un premier temps, on écrit un programme permettant de dénombrer tous les cas (favorables et possibles) et d'en déduire la probabilité p de gagner au cours du jeu. Cela sensibilise les élèves à l'intérêt de la programmation lorsqu'en l'absence de raisonnement combinatoire, on est obligé de dénombrer tous les cas. Dans un deuxième temps, on</p>		<p>Permet de travailler les éléments du programme :</p> <p>Lire et comprendre une fonction Python renvoyant le nombre ou la fréquence de succès dans un échantillon de taille n pour une expérience aléatoire à deux issues.</p> <p>Observer la loi des grands nombres à l'aide d'une simulation sur Python.</p> <p>Simuler N échantillons de taille n d'une expérience aléatoire à deux issues. Si p est la probabilité d'une issue et f sa fréquence observée dans un échantillon, calculer la proportion des cas où l'écart entre p et f est inférieur ou égal à <math>\sqrt{n}</math></p>

<p><i>simule des échantillons de différentes tailles de manière à visualiser la stabilisation des fréquences lorsque la taille des échantillons croît. Dans un troisième temps, on simule N échantillons de même taille n et on calcule la proportion des cas où l'écart entre la probabilité p et la fréquence de succès dans l'échantillon est inférieure à <math>1/\sqrt{n}</math>.</i> »</p>		<p>Permet de sensibiliser à l'intérêt de la programmation pour dénombrer tous le cas</p> <p>Permet de visualiser la stabilisation des fréquences</p>
<p><b>Partie 1 : Calculer la probabilité de gagner</b></p> <p>« Situation : Le jour de la Fête de la science, des étudiants organisent un jeu : le joueur lance 3 dés cubiques (un rouge, un vert, un bleu) bien équilibrés dont les faces sont numérotées de 1 à 6. S'il arrive à construire un triangle (non aplati) dont les côtés ont pour mesure les valeurs des faces obtenues, il gagne, sinon il perd. [...] La fonction suivante prend en paramètres trois nombres entiers correspondant aux valeurs des faces des trois dés et renvoie 1 si la partie est gagnée et 0 sinon. »</p> <p><b>Question 1 :</b></p> <p><i>Compléter un programme. Le programme précédent étant fourni [...] demander aux élèves de compléter les lignes 2, 3, 4 et 5.</i> »</p> <pre>def partie(d1, d2, d3) :     sommeDesTroisCotes = ...     grand = ...     sommeDesDeuxAutresCotes = ...     if ...         return 1     else :         return 0</pre> <p>« Calcul de la probabilité de gagner. À l'aide de boucles imbriquées, on calcule le nombre de toutes les issues possibles (total) lors du jeu et le nombre de celles qui permettent de gagner (favorable). Le</p>	<p><b>Question 1 :</b></p> <pre>def partie(d1, d2, d3) :     sommeDesTroisCotes = d1\         + d2 + d3     grand = max(d1, d2, d3)     sommeDesDeuxAutresCotes = \         sommeDesTroisCotes - grand     if grand &lt; sommeDesDeuxAutresCotes:         return 1     else :         return 0 # Tests print(partie(2,3,4)) print(partie(1,2,3)) print(partie(1,1,4))</pre> <p><b>Ingrédients élémentaires :</b></p> <ul style="list-style-type: none"> <li>Modifier la valeur d'une variable (affectation) x3</li> <li>Utiliser la valeur d'une variable x4</li> <li>Manipuler une variable de type « int » (entier) x3</li> <li>Additionner des nombres</li> <li>Soustraire des nombres</li> </ul>	

quotient favorable/total donne alors la probabilité de gagner lors d'une partie. »

Question 2 :

« Écrire un programme. Écrire un programme qui permet de calculer le nombre exact de cas favorables et le nombre de cas possibles.

Compléter un programme. Le programme précédent étant fourni [...] demander aux élèves de compléter les lignes 5, 6 et 7. »

```
favorable, total = 0, 0
for d1 in range(1,7) :
    for d2 in range(1,7) :
        for d3 in range(1,7) :
            if ...
                favorable = ...
            total = ...
print(favorable, total)
print(favorable/total)
```

Utiliser une structure conditionnelle à deux branches

Tester une inégalité sur des nombres

Définir une fonction avec paramètres et retour

Utiliser une fonction avec paramètres et avec retour x4

Utiliser une fonction avec paramètres et sans retour

Question 2 :

```
favorable, total = 0, 0
for d1 in range(1,7) :
    for d2 in range(1,7) :
        for d3 in range(1,7) :
            if partie(d1, d2, d3) == 1:
                favorable = favorable + 1
            total = total + 1
print(favorable, total)
print(favorable/total)
```

Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x4

Utiliser la valeur d'une variable x9

Manipuler une variable de type « int » (entier) x2

	<p>Utiliser une boucle for avec utilisation de la variable de boucle x3</p> <p>Utiliser une structure conditionnelle à deux branches</p> <p>Tester une inégalité sur des nombres</p> <p>Additionner des nombres x2</p> <p>Utiliser une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et sans retour x2</p>	
<p><b>Partie 2 : Calculer la fréquence des parties gagnées dans un échantillon simulé</b></p> <p>« On calcule la fréquence des parties gagnées sur un échantillon de <math>n</math> parties simulées. La fonction <code>echantillon</code> prend en paramètre l'entier <math>n</math> (taille de l'échantillon) et renvoie la proportion de parties gagnées sur cet échantillon. »</p> <p><u>Question 1 :</u></p> <p>« Compléter un programme. Le programme précédent étant fourni [], demander aux élèves de compléter les lignes 5 à 8 de manière à simuler <math>n</math> parties et à compter dans la variable <code>gagne</code> le nombre de parties gagnées (en utilisant la fonction <code>partie</code>).</p>	<p><u>Question 1 :</u></p> <pre>def echantillon(n) :     gagne = 0     for i in range(n) :         d1 = randint(1,6)         d2 = randint(1,6)         d3 = randint(1,6)         gagne = gagne + \             partie(d1, d2, d3)     return gagne / n  print(echantillon(10)) print(echantillon(100)) print(echantillon(1000))</pre> <p><u>Ingrédients élémentaires :</u></p>	

<pre>def echantillon(n) :     gagne = 0     for i in range(n) :         d1 = ...         d2 = ...         d3 = ...         gagne = ...     return gagne / n</pre> <p><b>Expliquer un programme. Expliquer la ligne 8. »</b></p>	<p>Modifier la valeur d'une variable (affectation) x5</p> <p>Utiliser la valeur d'une variable x4</p> <p>Manipuler une variable de type « int » (entier) x4</p> <p>Utiliser une boucle for sans utilisation de la variable de boucle.</p> <p>Additionner des nombres</p> <p>Diviser des nombres</p> <p>Définir une fonction avec paramètres et retour</p> <p>Utiliser une fonction avec paramètres et retour x7</p> <p>Utiliser une fonction avec paramètres et sans retour x3</p>	
<p><b>Partie 3 : Fluctuation de la fréquence des parties gagnées dans plusieurs échantillons simulés.</b></p> <p>« Le programme suivant affiche des listes contenant les proportions (ou fréquences) de parties gagnées pour 10 échantillons de taille 10, puis 10 échantillons de taille 100 et enfin 10 échantillons de taille 1000. »</p> <pre>print([echantillon(10) for i in range(10)]) print([echantillon(100) for i in range(10)]) print([echantillon(1000) for i in range(10)])</pre>		



### Question 1 :

« **Observer. Que constate-t-on dans les listes précédentes ?** »

« La fonction ci-dessous permet de réaliser  $N = 100$  simulations différentes d'échantillons de  $n = 1000$  parties. On a représenté par des points les fréquences obtenues : en vert les fréquences qui sont dans l'intervalle  $\left[p - \frac{1}{\sqrt{n}}; p + \frac{1}{\sqrt{n}}\right]$ , en rouge celles qui ne le sont pas. »

### Question 2 :

« **Tester le programme pour différentes valeurs de  $n$ .**

```
from matplotlib import pyplot as plt
from math import sqrt
n, p = 1000, 111/216
N = 100
mini, maxi = p-1/sqrt(n), p+1/sqrt(n)
for repet in range(N) :
    f = echantillon(n)
    if mini < f < maxi :
        plt.plot(repet, f, "go")
    else :
        plt.plot(repet, f, "ro")
plt.plot([0, repet], [p, p], "m--")
plt.plot([0, repet], [mini, mini], "k--")
plt.plot([0, repet], [maxi, maxi], "k--")
plt.show()
```

**Compléter le programme précédent pour calculer le nombre de cas où l'écart entre la fréquence de gagner au cours de  $n$  parties et la probabilité  $p$  est inférieur ou égal à  $\frac{1}{\sqrt{n}}$  (on prendra toujours  $N = 100$  échantillons).**

Écrire une fonction qui prend en argument le nombre  $n$  de parties et qui renvoie le nombre de cas où l'écart entre la fréquence de gagner au cours des  $n$  parties et la probabilité  $p$  est inférieur ou égal à  $\frac{1}{\sqrt{n}}$ . Le nombre d'échantillons est fixé, par exemple à  $N = 100$  »

### A.3.3 Ressource d'accompagnement R2MS (MEN, 2017)

#### A.3.3.1 Organisations didactiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Adosser les activités algorithmiques aux mathématiques</b> (contenu)</p> <p>« Elle prévoit une partie dédiée à l'algorithmique et la programmation dont la rédaction a été remaniée et rappelle que le travail correspondant doit être réinvesti dans les trois autres parties (fonctions, géométrie, statistiques et probabilités). À la différence du programme de mathématiques du cycle 4 du collège, il s'agit donc d'adosser explicitement les activités de la partie algorithmique et programmation aux mathématiques. » p. 1</p>		
<p><b>Proposer des modalités d'apprentissage variées</b> (dispositif/artefact)</p> <p>« Les modalités de l'apprentissage correspondant peuvent être variées : travail individuel ou en groupe, en salle informatique ou en salle banale, au tableau ou sur papier, sur tablette ou sur ordinateur. » p. 1</p>	<p>Individuel ou en groupe</p> <p>Salle informatique ou salle banale</p> <p>Au tableau ou sur papier</p> <p>Sur tablette ou ordinateur</p>	
<p><b>Faire utiliser un langage textuel aux élèves</b> (artéfact)</p> <p>« Le choix d'un langage textuel, comme Python, au lieu d'un langage par blocs, comme Scratch, permet aux élèves de se confronter à la précision et la rigidité d'une syntaxe proche de celle des expressions mathématiques, avec l'avantage de pouvoir bénéficier du contrôle apporté par l'analyseur syntaxique. » p. 1</p>	<p>Utiliser par exemple Python</p>	<p>Permet aux élèves de se confronter à la précision et la rigidité d'une syntaxe proche des expressions mathématiques.</p>

		Permet de bénéficier de l'avantage du contrôle syntaxique.
<p><b>Faire utiliser le langage Python aux élèves (artéfact)</b></p> <p>« le langage Python, choisi pour la concision et la simplicité de sa syntaxe, la taille de la communauté d'utilisateurs (en particulier dans le cadre éducatif), ainsi que la richesse des ressources disponibles » p. 1</p>		<p>Le langage Python possède une syntaxe simple et concise</p> <p>Il possède une grande communauté d'utilisateurs.</p> <p>Les ressources disponibles sont riches.</p>
<p><b>Ne pas former les élèves à devenir des experts d'un langage de programmation (contenu)</b></p> <p>« Le professeur gardera à l'esprit que l'enseignement de la partie algorithmique et programmation n'a pas pour objectif de former des experts dans tel ou tel langage de programmation ou dans la connaissance détaillée de telle ou telle bibliothèque de programme » p. 1</p>		
<p><b>Présenter la notion d'algorithme (contenu)</b></p> <p>« Un algorithme est une procédure de résolution de problème [...]. Un algorithme s'applique donc à une famille d'instances d'un problème et produit, en un nombre fini d'étapes constructives, effectives, non-ambigües et organisées, la réponse au problème pour toute instance de cette famille. De la même façon qu'un script Scratch se construit en accolant des briques élémentaires, un algorithme s'appuie sur un ensemble très réduit de constructions : l'affectation d'une variable, la séquence d'instructions, l'instruction conditionnelle, les boucles (fors ou non fors), les fonctions. » p.2</p>	Un algorithme est un ensemble de constructions élémentaires permettant de résoudre un problème	

<p><b>Ne pas présenter les notions algorithmiques de manière magistrale (dispositif)</b></p> <p>« Ci-après sont décrites ces différentes constructions. Toutefois, <b>il n'est pas recommandé de les présenter de façon magistrale aux élèves, qui les ont déjà rencontrées au cycle 4</b> » p. 2</p>	<p>Mobiliser les notions algorithmiques</p>	<p>Les notions ont déjà été rencontrées au cycle 4</p>
<p><b>Introduire les notions dans des activités (contenu/activité)</b></p> <p>« Toutefois, il n'est pas recommandé de les présenter de façon magistrale aux élèves, [...], <b>mais de les mobiliser de façon naturelle dans les activités travaillées</b> » p. 2</p>	<p>Mobiliser les notions de façon naturelle</p>	
<p><b>Introduire les notions en les imbriquant de manière progressive (contenu/activité)</b></p> <p>« Ci-après sont décrites ces différentes constructions. Toutefois, il n'est pas recommandé de les présenter de façon magistrale aux élèves [...], <b>mais de les mobiliser [...] en introduisant progressivement leur imbrication</b> » p. 2</p>		
<p><b>Utiliser la métaphore de la boîte pour introduire les variables (contenu)</b></p> <p>« Un modèle rigoureux de la variable informatique consiste à <b>dire qu'une variable est une étiquette collée sur une boîte qui peut contenir différentes valeurs</b>. Le contenu de chaque boîte varie au cours de l'exécution d'un programme » p. 2</p>		
<p><b>Être attentif à la différence entre affectation et test d'égalité (contenu)</b></p> <p>« <b>On observe également que le symbole = a été choisi pour l'opération d'affectation en Python. Le symbole = n'est donc pas disponible pour le test d'égalité qui s'écrit ==. On sera attentif à cette</b></p>		<p>L'affectation est omniprésente en informatique et absente en mathématiques. Le test de l'égalité est moins fréquent en informatique.</p>

<p><b>différence</b> : l'affectation est absente en mathématiques et omniprésente en informatique, le test d'égalité d'un usage moins fréquent en informatique » p. 3</p>		
<p><b>Distinguer la conception et l'utilisation des algorithmes (activité)</b></p> <p>« Il est important de distinguer très clairement la conception des algorithmes et leur utilisation. Les élèves écriront des fonctions qui pourront être enregistrées dans des fichiers, appelés scripts ou modules, qu'on peut développer, enrichir et réutiliser lors de séances successives. Ces modules peuvent être exécutés dans un mode console, afin de tester et mettre au point les algorithmes et les programmes. L'exécution d'un module comportant la définition d'une fonction <math>f</math> ne produit aucun affichage particulier dans la console, même si la définition a été prise en compte : on peut maintenant, dans la console, faire des appels du type <math>f(2)</math> et la console affiche la valeur renvoyée par la fonction. On notera que les notions d'entrées-sorties (fonctions input et print) ne sont pas développées dans ce document : elles ne relèvent pas de la pensée algorithmique et l'accent mis par le programme sur la notion de fonction permet de s'en libérer complètement. » p. 5</p>	<p>Les fonctions sont écrites dans des fichiers.</p> <p>Les fonctions sont exécutés par appels dans la console.</p>	<p>Les entrées-sorties input et print ne relèvent pas de la pensée algorithmique.</p>
<p><b>Faire la transition de Scratch à Python (contenu)</b></p> <p>« En classe de seconde, le passage de Scratch à Python peut être immédiat ou progressif, suivant les choix pédagogiques de l'enseignant. Les deux langages comportent, au-delà des différences évidentes de forme, des similitudes qui facilitent la transition. » p. 6</p>	<p>Immédiat ou progressif</p>	<p>Les deux langages comportent des similitudes qui facilitent la transition.</p>

## A.4 Seconde – Sciences numériques et Technologie

### A.4.1 Programme d'enseignement PSS (MEN, 2019c)

#### A.4.1.1 Organisations didactiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Présenter la notion d’algorithme (contenu)</b></p> <p>« <i>les algorithmes, qui spécifient de façon abstraite et précise des traitements à effectuer sur les données à partir d’opérations élémentaires</i> » p.2</p>	Un algorithme spécifie un traitement de manière abstraite	
<p><b>Présenter la notion de langage (contenu)</b></p> <p>« <i>les langages, qui permettent de traduire les algorithmes abstraits en programmes textuels ou graphiques de façon à ce qu’ils soient exécutables par les machines</i> » p.2</p>	Un langage de programmation permet d’implémenter les algorithmes sur des machines	
<p><b>Proposer des activités de programmation liées aux thèmes du programme (activité)</b></p> <p>« <i>On approfondit les connaissances et compétences des élèves, notamment la pratique de la programmation à travers les activités liées aux thèmes du programme : internet ; le Web ; les réseaux sociaux ; les données structurées et leur traitement ; localisation, cartographie et mobilité ; informatique embarquée et objets connectés ; la photographie numérique.</i> » p. 2-3</p> <p>« <i>Notions transversales de programmation</i> » p. 3</p> <p>« <i>Écrire et développer des programmes pour répondre à des problèmes et modéliser des phénomènes</i> » p. 3</p>	Répondre à des problèmes et modéliser des phénomènes	

<p>« <i>Illustrer ces notions par des activités liées aux différents thèmes du programme</i> » p. 3</p>		
<p><b>Collaborer avec les collègues d'autre matières (contenu)</b></p> <p>« <i>La réflexion sur les sciences numériques et sur leur relation à la technologie peut être conduite dans le cadre d'autres enseignements, que ce soit au travers de l'étude d'œuvres littéraires ou artistiques, de la réflexion sur les enjeux éthiques et politiques, d'analyses des conséquences de la révolution numérique sur l'évolution des métiers. Ces perspectives incitent le professeur en charge de l'enseignement de sciences numériques et technologie à collaborer avec ses collègues.</i> » p. 2</p>	<p>Étudier des œuvres littéraires ou artistiques</p> <p>Réfléchir sur les enjeux éthiques et politiques</p> <p>Analyser les conséquences de la révolution numérique sur les métiers.</p>	<p>La réflexion sur les sciences numériques peut être conduite dans le cadre d'autres enseignements</p>
<p><b>Faire travailler les élèves sous des formes variés (dispositif)</b></p> <p>« <i>Cet enseignement a vocation à multiplier les occasions de mise en activité des élèves, sous des formes variées (exposés, travaux en groupe, mini-projets, productions individuelles ou collectives, etc.) qui permettent de développer des compétences transversales : faire preuve d'autonomie, d'initiative et de créativité ; présenter un problème ou sa solution, développer une argumentation dans le cadre d'un débat ; coopérer au sein d'une équipe ; rechercher de l'information, apprendre à utiliser des sources de qualité, partager des ressources ; faire un usage responsable et critique des sciences et technologies numériques.</i> » p. 3</p>	<p>Exposées</p> <p>Travaux en groupe</p> <p>Mini-projets</p> <p>Productions individuelles</p> <p>Productions collectives</p>	<p>Permet de développer des compétences transversales :</p> <ul style="list-style-type: none"> <li>- autonomie ;</li> <li>- initiative ;</li> <li>- créativité ;</li> <li>- présenter un problème et sa solution ;</li> <li>- développer une argumentation</li> <li>- coopérer</li> <li>- rechercher de l'information</li> <li>- utiliser des sources de qualité</li> <li>- partager des ressources</li> </ul>



		- faire un usage responsable et critique des technologies
<p><b>Faire utiliser aux élèves de langage Python (artéfact)</b></p> <p>« <i>Un langage de programmation est nécessaire pour l'écriture des programmes : un langage simple d'usage, interprété, concis, libre et gratuit, multiplateforme, largement répandu, riche de bibliothèques adaptées aux thématiques étudiées et bénéficiant d'une vaste communauté d'auteurs dans le monde éducatif est nécessaire. Au moment de la conception de ce programme, le langage choisi est Python version 3 (ou supérieure).</i> » p. 3</p>		<p>Python est :</p> <ul style="list-style-type: none"> <li>- simple d'usage</li> <li>- interprété</li> <li>- convis</li> <li>- libre et gratuit</li> <li>- multiplateforme</li> <li>- largement répandu</li> <li>- riche en bibliothèques adaptées aux thématiques étudiées</li> <li>- bénéficiaire d'une vaste communauté d'auteurs dans le monde éducatif</li> </ul>
<p><b>Se coordonner avec le cours de mathématiques (contenu)</b></p> <p>« <i>Une coordination avec le cours de mathématiques est donc nécessaire pour déterminer à quel moment des éléments de programmation peuvent être utilisés en sciences numériques et technologie.</i> » p. 4</p>		Permet de déterminer à quel moment les éléments de programmation peuvent être utilisés en SNT
<p><b>Faire utiliser aux élèves les langages HTML et CSS (artefact)</b></p> <p>« <i>Les pages Web sont écrites dans le langage de balises HTML (HypertextMarkupLanguage). Leur style graphique est défini dans le langage CSS (Cascading Style Sheets)</i> » p.6</p>	Faire la différence entre le contenu et le style d'une page Web.	

<p>« <b>Contenus : Langage HTML et CSS. Distinguer ce qui relève du contenu d'une page et de son style de présentation. Étudier et modifier une page HTML simple</b> » p.8</p>	<p>Étudier et modifier une page HTML</p>	
<p><b>Faire programmer un algorithme de calcul de popularité d'une page Web (contenu/activité)</b></p> <p>« <b>Partie Le Web : Calculer la popularité d'une page à l'aide d'un graphe simple puis programmer l'algorithme</b> » p. 8</p>		
<p><b>Faire télécharger des données ouvertes et observer les traitements possibles avec Python (contenu/activité)</b></p> <p>« <b>Partie Les données structurées et leur traitement : Télécharger des données ouvertes</b> (sous forme d'un fichier au format CSV avec les métadonnées associées), <b>observer les différences de traitements possibles selon le logiciel choisi pour lire le fichier : programme Python, [...]</b> » p. 13</p> <p>« <b>La recherche dans des données structurées a d'abord été effectuée selon une indexation préalable faite par l'homme. Des algorithmes ont ensuite permis d'automatiser l'indexation à partir de textes, d'images ou de sons.</b> » p. 12</p>		<p>Les algorithmes permettent d'automatiser l'indexation des données.</p>
<p><b>Faire identifier les algorithmes de contrôle des comportements physiques (contenu/activité)</b></p> <p>« <b>Partie Informatique embarquée et objets connectés : Systèmes informatiques embarqués. Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.</b> » p. 17</p>	<p>Identifier les algorithmes permettant :</p> <ul style="list-style-type: none"> <li>- le contrôle des freins et du moteur d'une automobile</li> <li>- l'assistance au pédalage d'un vélo électrique</li> </ul>	

<p>« Identifier les évolutions apportées par les algorithmes au contrôle des freins et du moteur d'une automobile, ou à l'assistance au pédalage d'un vélo électrique. » p. 17</p>		
<p><b>Faire réaliser l'IHM simple d'un objet connecté.</b> (contenu/activité)</p> <p>« Partie Informatique embarquée et objets connectés : Interface homme-machine (IHM). Réaliser une IHM simple d'un objet connecté. » p. 17</p> <p>« Réaliser une IHM pouvant piloter deux ou trois actionneurs et acquérir les données d'un ou deux capteurs. » p. 17</p>	<p>Réaliser une IHM pouvant piloter des actionneurs et acquérir les données de capteurs.</p>	
<p><b>Faire programmer l'acquisition de données ou la commande d'un actionneur</b> (contenu/activité)</p> <p>« Partie Informatique embarquée et objets connectés : Commande d'un actionneur, acquisition des données d'un capteur. Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur. » p. 17</p> <p>« Gérer des entrées/sorties à travers les ports utilisés par le système. » p. 17</p> <p>« Utiliser un tableau de correspondance entre caractères envoyés ou reçus et commandes physiques (exemple : le moteur A est piloté à 50 % de sa vitesse maximale lorsque le robot reçoit la chaîne de caractères « A50 »). » p. 17</p> <p>« Le développement des logiciels embarqués est délicat, car il pose souvent des questions de temps-réel, c'est-à-dire de respect de temps de réponse imposé. Ceci conduit à des méthodes de programmation spécifiques. » p. 16</p>	<p>Gérer les entrées/sorties d'un système à travers ses ports.</p> <p>Utiliser un tableau de correspondance entre caractères numériques et commandes physiques.</p>	<p>Les logiciels embarqués pose des questions délicates de temps-réel et cela conduit à l'utilisation de méthodes de programmation spécifiques</p>

<p><b>Faire programmer un traitement de transformation d'image (contenu/activité)</b></p> <p>« <i>Partie La photographie numérique : Traitement d'image. Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels.</i> » p. 19</p> <p>« <i>Programmer un algorithme de passage d'une image couleur à une image en niveaux de gris : par moyenne des pixels RVB ou par changement de modèle de représentation (du RVB au TSL, mise de la saturation à zéro, retour au RVB).</i> » p. 19</p> <p>« <i>Programmer un algorithme de passage au négatif d'une image.</i> » p. 19</p> <p>« <i>Programmer un algorithme d'extraction de contours par comparaison entre pixels voisins et utilisation d'un seuil.</i> » p. 19</p> <p>« <i>Des algorithmes permettent de traiter toutes les lumières, d'effectuer une retouche facile, avec une qualité maintenant bien supérieure à l'argentique.</i> » p. 18</p>	<p>Agir sur les trois composantes des pixels</p> <p>Programmer un algorithme de passage d'une image en couleur en niveaux de gris par moyenne des pixels RVB</p> <p>Programmer un algorithme de passage d'une image en couleur en niveaux de gris par changement de modèle de représentation</p> <p>Programmer un algorithme de passage au négatif d'une image</p> <p>Programmer un algorithme d'extraction de contours par comparaison entre pixels voisins et utilisation d'un seuil.</p>	<p>Les algorithmes permettent d'effectuer des traitements d'une qualité bien supérieure à l'argentique</p>
<p><b>Faire expliciter les algorithmes liés à la prise de vue (contenu/activité)</b></p> <p>« <i>Partie La photographie numérique : Rôle des algorithmes dans les appareils photo numériques. Expliciter des algorithmes associés à la prise de vue. Identifier les étapes de la construction de l'image finale.</i> » p. 19</p> <p>« <i>Avec l'arrivée du téléphone mobile, des algorithmes de fusion d'images permettent de concilier une excellente qualité avec un capteur et un objectif minuscules. De nombreux algorithmes</i></p>		<p>Des nombreux algorithmes sont utilisés dans les appareils photos numériques.</p> <p>Ces algorithmes permettent de concilier qualité et miniaturisation.</p>

<i>sophistiqués sont utilisés dans les appareils de photographie numérique : [...] » p. 18</i>		
------------------------------------------------------------------------------------------------	--	--

## A.4.2 Ressource d'accompagnement R1SS (MEN, 2019d)

### A.4.2.1 Organisations informatiques

Type de tâches	Technique(s)	Technologie(s)
<p><b>Activité 1 : traiter des données à l'aide d'un programme</b></p> <p>(MEN 2019 - Les données structurées et leur traitement - données des salles de cinéma en Île-de-France - <a href="https://bit.ly/35u91cC">https://bit.ly/35u91cC</a>)</p> <p>« Contenus : <b>Traitement de données structurées</b>. Capacités attendus : <b>Réaliser des opérations de recherche, filtre, tri ou calcul sur une ou plusieurs tables.</b> » p. 1</p> <p>« Cette activité peut être découpée en plusieurs parties et proposée à différents moments de l'année scolaire puisqu'elle est <b>transversale à deux thématiques (les données structurées et leur traitement/géolocalisation, cartographie et mobilité)</b>. <b>La première partie consiste à importer des données ouvertes</b>. Il est possible de trouver facilement des données autres que celles des salles de cinéma d'Île de France. [...] La quatrième partie, <b>permet de traiter les données en écrivant des programmes en Python</b>. [...]. <b>Les programmes sont soit donnés et à étudier, soit à compléter.</b> » p. 1</p>	<p>Étudier des programmes</p> <p>Compléter des programmes</p>	<p>Partie du programme travaillée :</p> <p>Traitement de données structurées</p> <p>Géolocalisation, cartographie mobilité</p>
<p><b>Partie 1 : Télécharger des données ouvertes sur les salles de cinéma en Ile-de-France</b></p> <p>« <b>Partie A - importer des données ouvertes</b></p> <p><b>À partir du site data.gouv.fr, télécharger des données ouvertes sur les salles de cinéma en Île-de-France. Nommer ce fichier salles.csv</b> » p. 2</p>	<p>En utilisant le site data.gouv.fr</p>	
<p><b>Partie 2 : Afficher les premières lignes du fichier</b></p>	<p><u>Question 1</u> :</p>	

## « Partie D - traitement automatisé des données »

### 1. Écrire le programme suivant :

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 print(descripteurs)
6 ligne = f.readline()
7 print(ligne)
```

#### 1.1. Expliquer les deux lignes qui s'affichent lors de l'exécution du programme.

### 2. Écrire le programme suivant :

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 for i in range(10):
6     ligne = f.readline()
7     print(ligne)
```

#### 2.1. Expliquer la ligne 5.

#### 2.2. Expliquer l'affichage lors de l'exécution du programme.

### 3. Écrire le programme suivant :

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 for i in range(10):
6     ligne = f.readline()
7     donnees = ligne.split(";")
8     print(donnees[2])
```

#### 3.1. Expliquer les lignes qui s'affichent lors de l'exécution du programme.

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 print(descripteurs)
6 ligne = f.readline()
7 print(ligne)
```

### Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x3

Utiliser la valeur d'une variable x4

Manipuler une variable de type objet « TextIOWrapper » (fichier)

Manipuler une variable de type « str » (chaîne de caractères)

Utiliser une méthode d'un objet sans paramètre avec retour x2

Utiliser une fonction avec paramètres et sans retour x2

Utiliser une fonction avec paramètres et avec retour

### Question 2 :

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 for i in range(10):
6     ligne = f.readline()
7     print(ligne)
```

<p>3.2. Modifier le programme précédent afin d'afficher les adresses des salles de cinéma. » p. 3</p>	<p><u>Ingrédients élémentaires :</u></p> <p>Modifier la valeur d'une variable (affectation) x3</p> <p>Utiliser la valeur d'une variable x3</p> <p>Manipuler une variable de type objet « TextIOWrapper » (fichier)</p> <p>Manipuler une variable de type « str » (chaîne de caractères)</p> <p>Utiliser une boucle for sans utilisation de la variable de boucle</p> <p>Utiliser une méthode d'un objet sans paramètre avec retour x2</p> <p>Utiliser une fonction avec paramètres et sans retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p> <p><u>Question 3 :</u></p> <pre>3 f = open("salles.csv", "r") 4 descripteurs = f.readline() 5 for i in range(10): 6     ligne = f.readline() 7     donnees = ligne.split(";") 8     print(donnees[2])</pre>	
-------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--



	<p><u>Ingrédients élémentaires :</u></p> <p>Modifier la valeur d'une variable (affectation) x4</p> <p>Utiliser la valeur d'une variable x4</p> <p>Manipuler une variable de type objet « TextIOWrapper » (fichier)</p> <p>Manipuler une variable de type « str » (chaîne de caractères) x2</p> <p>Manipuler une variable de type « list » (liste)</p> <p>Accéder à un élément d'une liste</p> <p>Utiliser une boucle for sans utilisation de la variable de boucle</p> <p>Utiliser une méthode d'un objet sans paramètre et avec retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Utiliser une fonction avec paramètres et sans retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p>	
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

### Partie 3 : Exploiter toutes les lignes du fichier

« 4. Pour étudier toutes les données du fichier.

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 ligne = f.readline()
6 compteur = 0
7 while ligne != "":
8     donnees = ligne.split(";")
9     print(donnees[3])
10    ligne = f.readline()
11    compteur = compteur + 1
12 print(compteur)
```

4.1. Expliquer l'utilisation de la boucle non for while ligne?

4.2. Écrire le programme Python ci-dessus.

4.3. Expliquer les lignes qui s'affichent lors de l'exécution du programme.

5. Pour calculer le nombre de salles de cinéma dans le 77.

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 ligne = f.readline()
6 compteur = 0
7 while ligne != "":
8     donnees = ligne.split(";")
9     if donnees[...] == "...":
10        compteur = ...
11    ligne = f.readline()
12 print(compteur)
```

5.1. Compléter le programme ci-dessus afin de calculer ce nombre. »  
p. 3-4

### Question 4 :

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 ligne = f.readline()
6 compteur = 0
7 while ligne != "":
8     donnees = ligne.split(";")
9     print(donnees[3])
10    ligne = f.readline()
11    compteur = compteur + 1
12 print(compteur)
```

### Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x7

Utiliser la valeur d'une variable x7

Manipuler une variable de type objet « TextIOWrapper » (fichier)

Manipuler une variable de type « int » (entier)

Manipuler une variable de type « str » (chaîne de caractères)

Manipuler une variable de type « list » (liste)

Accéder à un élément d'une liste

Utiliser une boucle non-for

Tester une égalité sur des chaînes de caractères

Additionner des nombres

Utiliser une méthode d'un objet sans paramètre et avec retour x3

Utiliser une méthode d'un objet avec paramètre et avec retour

Utiliser une fonction avec paramètres et sans retour x2

Utiliser une fonction avec paramètres et avec retour

Question 5 :

```
3 f = open("salles.csv", "r")
4 descripteurs = f.readline()
5 ligne = f.readline()
6 compteur = 0
7 while ligne != "":
8     donnees = ligne.split(";")
9     if donnees[1] == "77":
10         compteur = compteur + 1
11     ligne = f.readline()
12 print(compteur)
```

Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation) x7

Utiliser la valeur d'une variable x7

	<p>Manipuler une variable de type objet « TextIOWrapper » (fichier)</p> <p>Manipuler une variable de type « str » (chaîne de caractères)</p> <p>Manipuler une variable de type « int » (entier)</p> <p>Manipuler une variable de type « list » (liste)</p> <p>Accéder à un élément d'une liste</p> <p>Utiliser une boucle non-for</p> <p>Utiliser une structure conditionnelle à une branche</p> <p>Tester une égalité sur des chaînes de caractères x2</p> <p>Additionner des nombres</p> <p>Utiliser une méthode d'un objet sans paramètre et avec retour x3</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Utiliser une fonction avec paramètres et sans retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p>	
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

#### Partie 4 : Exploiter les données de géolocalisation du fichier

##### « Partie E - géolocalisation des données

Carla souhaite voir sur une carte la position des salles de cinéma. Pour cela nous allons créer un programme qui utilise la latitude et longitude des salles de cinéma du fichier « Salles.csv » et crée une carte dans le fichier carte\_salles.html.

##### 1. Écrire le programme suivant. Enregistrer le : salle.py.

```
1 import folium
2 carte_salles = folium.Map(\
3     location=[42.97,2.76],zoom_start=10)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     coordonneesGPS = donnees[33].split(",")
10    latitude = float(coordonneesGPS[0])
11    longitude = float(coordonneesGPS[1])
12    folium.Marker([latitude,longitude])\
13        .add_to(carte_salles)
14    ligne = f.readline()
15 carte_salles.save("carte_salles.html")
```

##### 1.1. Exécuter le programme.

##### 2. Ouvrir le fichier carte\_salles.html avec votre navigateur.

2.1. Nommer le logiciel qui permet de visualiser le fichier carte\_salles.html qui se trouve dans le même dossier que votre programme.

2.2. Malheureusement la carte n'est pas centrée sur Paris. Modifier la ligne 2 du programme afin de centrer la carte sur Paris.

#### Question 2 :

```
1 import folium
2 carte_salles = folium.Map(\
3     location=[42.97,2.76],zoom_start=10)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     coordonneesGPS = donnees[33].split(",")
10    latitude = float(coordonneesGPS[0])
11    longitude = float(coordonneesGPS[1])
12    folium.Marker([latitude,longitude])\
13        .add_to(carte_salles)
14    ligne = f.readline()
15 carte_salles.save("carte_salles.html")
```

#### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x9

Utiliser la valeur d'une variable x12

Manipuler une variable de type objet « Map » (carte)

Manipuler une variable de type objet « TextIOWrapper » (fichier)

Manipuler une variable de type « str » (chaîne de caractères) x2

Manipuler une variable de type « list » (liste) x2

2.3. Expliquer les lignes 9, 10, 11 et 12.

3. Carla souhaite ne voir que les salles de cinéma du département 75. Compléter les lignes 12 et 13 du programme pour obtenir la carte ci-dessous. » p. 5-6

```
1 import folium
2 carte_salles = folium.Map(\
3     location=[48.85,2.34],zoom_start=12)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     coordonneesGPS = donnees[33].split(",")
10    latitude = float(coordonneesGPS[0])
11    longitude = float(coordonneesGPS[1])
12    dep = donnees[...]
13    if dep == "...":
14        folium.Marker([latitude,longitude])\
15            .add_to(carte_salles)
16    ligne = f.readline()
17 carte_salles.save("carte_salles75.html")
```



Manipuler une variable de type « float » (flottant) x2

Accéder à un élément d'une liste x3

Utiliser une boucle non-for

Tester une égalité sur des chaînes de caractères

Utiliser une méthode d'un objet sans paramètre et avec retour x2

Utiliser une méthode d'un objet avec paramètre et avec retour x2

Utiliser une méthode d'un objet avec paramètre et sans retour x2

Utiliser une fonction avec paramètres et avec retour x5


Question 3 :

```
1 import folium
2 carte_salles = folium.Map(\
3     location=[48.85,2.34],zoom_start=12)
4 f = open("salles.csv","r")
5 descripteurs = f.readline()
6 ligne = f.readline()
7 while ligne != "":
8     donnees = ligne.split(";")
9     coordonneesGPS = donnees[33].split(",")
10    latitude = float(coordonneesGPS[0])
11    longitude = float(coordonneesGPS[1])
12    dep = donnees[1]
13    if dep == "75":
14        folium.Marker([latitude,longitude])\
15            .add_to(carte_salles)
16    ligne = f.readline()
17 carte_salles.save("carte_salles75.html")
```

	<p><u>Ingrédients élémentaires :</u></p> <p>Importer une bibliothèque</p> <p>Modifier la valeur d'une variable (affectation) x10</p> <p>Utiliser la valeur d'une variable x14</p> <p>Manipuler une variable de type objet « Map » (carte)</p> <p>Manipuler une variable de type objet « TextIOWrapper » (fichier)</p> <p>Manipuler une variable de type « str » (chaîne de caractères) x3</p> <p>Manipuler une variable de type « list » (liste) x2</p> <p>Manipuler une variable de type « float » (flottant) x2</p> <p>Accéder à un élément d'une liste x3</p> <p>Utiliser une boucle non-for</p> <p>Utiliser une structure conditionnelle à une branche</p> <p>Tester une égalité sur des chaînes de caractères x2</p>	
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

	<p>Utiliser une méthode d'un objet sans paramètre et avec retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une fonction avec paramètres et avec retour x5</p>	
<p><b>Activité 2 : programmer un robot</b></p> <p>(MEN 2019 - Informatique embarquée et objets connectés. Activité 4 : Robot thymio - <a href="https://bit.ly/3mbHHGv">https://bit.ly/3mbHHGv</a>)</p> <p>« <i>Contenus et capacités :</i></p> <p><i>Systemes informatiques embarqués : Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.</i></p> <p><i>Commande d'un actionneur, acquisition des données d'un capteur : Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.</i> » p. 1</p> <p>« <i>Note d'intention. Dans ces exercices nous travaillerons la thématique de l'informatique embarquée et objets connectés à partir du robot Thymio. L'idée étant d'aborder cette notion sans rentrer dans des détails techniques sur les réglages de capteurs ou d'actionneurs qui se révélerait contreproductif ; les difficultés inhérentes à ses réglages pouvant prendre le pas sur les difficultés informatiques de traitement de l'information. Pourquoi le robot</i></p>	<p>Utiliser le robot Thymio</p> <p>Programmer Thymio avec Scratch</p>	<p>Permet de travailler les parties du programme :</p> <ul style="list-style-type: none"> <li>-Identifier des algorithmes de contrôle des comportements physiques à travers les données des capteurs, l'IHM et les actions des actionneurs dans des systèmes courants.</li> <li>-Écrire des programmes simples d'acquisition de données ou de commande d'un actionneur.</li> </ul> <p>Le robot Thymio permet de ne pas rentrer dans les détails techniques sur les capteurs et les actionneurs, ce qui est contreproductif</p>



<p><i>Thymio ? Ce robot à but exclusivement pédagogique est déjà utilisé depuis quelques années à tous les niveaux de l'école maternelle jusqu'à l'enseignement supérieur. Le logiciel « Thymio Suite » propose différents modes de programmation donc l'un avec Scratch. Ce langage a l'avantage d'avoir déjà été utilisé par les élèves au collège permettant ainsi d'économiser un temps précieux de prise en main de l'interface. Le robot Thymio embarque nativement de nombreux capteurs et actionneurs, cela nous permet de nous focaliser exclusivement sur la partie traitement de l'information. »</i></p> <p>p. 1</p>		<p>Le robot Thymio est exclusivement pédagogique est utilisé depuis quelques années à tous les niveaux d'enseignement.</p> <p>Le robot Thymio inclut nativement de nombreux capteurs et actionneurs, ce qui permet de nous focaliser sur la partie traitement de l'information.</p> <p>L'utilisation de Scratch permet de gagner du temps car les élèves maîtrisent déjà l'environnement.</p>
<p><b>Partie 1 : Prendre en main le robot Thymio</b></p> <p>« <i>Étape 1 - Prise en main du robot Thymio et de « Thymio Suite » Cette première étape, indispensable, ne travaille pas le lien capteur/actionneur mais monopolise uniquement les actionneurs du robot. L'ensemble des exercices proposés ici ont pu en amont avoir été travaillés avec python et le module turtle.</i></p> <p><i>Exercice 1 – Carré. Après avoir mis un crayon dans l'emplacement prévu à cet effet, tracer un carré sur une feuille placée sous le robot.</i></p> <p><i>Exercice 2 – Carré. Tracer un carré avec ses deux diagonales.</i></p> <p><i>Exercice 3 - Quelques polygones. Tracer un hexagone, un octogone, un décagone et enfin un double décagone.</i></p>	<p><b>Exercices 3 : Pour le décagone</b></p>  <p><b>Ingrédients élémentaires :</b></p> <p>Attendre la survenue d'un évènement</p>	<p>Cette première étape est indispensable, elle permet de prendre en main de robot et l'interface de programmation.</p> <p>Le rendu n'étant pas exceptionnel, cela permet d'illustrer la différence entre algorithme et exécution effective sur une machine.</p>

<p>Exercice 4 – Cercle. <b>Tracer un cercle.</b></p> <p>Commentaires : <b>On s’aperçoit très rapidement que le rendu n’est pas exceptionnel</b> : la feuille peut bouger, le stylo flotter dans l’emplacement, les angles ne sont pas parfaits. <b>Cela permet d’illustrer la différence entre l’algorithme juste et la difficulté dans la mise en application lorsque l’on passe par une machine.</b> <b>Quoi qu’il en soit, il s’agit juste ici de prendre en main le robot et son interface de programmation.</b> La deuxième partie de la séance verra un travail plus approfondi sur la prise d’information par les capteurs qui n’est pas traité ici. » p. 2</p>	<p>Utiliser une boucle-for</p> <p>Agir sur le robot avec paramètre (utiliser une méthode d’un objet avec paramètre sans retour) x2</p>	
<p><b>Partie 2 : Programmer des acquisitions de données et le pilotage des actionneurs</b></p> <p>« Étape 2 - <b>Acquisition de données et commande d’actionneurs</b></p> <p>Exercice 1 - Capteurs/actionneurs</p> <ol style="list-style-type: none"> <li><b>Lister les capteurs du robot Thymio.</b></li> <li><b>Lister les actionneurs du robot Thymio.</b></li> </ol> <p>Exercice 2 - <b>Manipulation de quelques actionneurs</b></p> <ol style="list-style-type: none"> <li><b>Faire avancer le robot continuellement</b> (la vitesse devra être paramétrable). <b>Préciser les actionneurs mis en jeu.</b></li> <li><b>Faire avancer puis s’arrêter le robot au bout de quelques secondes.</b> <b>Préciser les actionneurs mis en jeu.</b></li> </ol> <p>Exercice 3 - Les capteurs entrent en jeu</p> <ol style="list-style-type: none"> <li><b>Faire avancer le robot continuellement.</b></li> <li><b>Modifier le programme précédent pour que le robot s’arrête en cas de détection d’obstacle.</b> Il devra reprendre si l’obstacle est retiré.</li> </ol>	<p><u>Exercice 3.2</u></p>	<p>Permet d’ajouter des capteurs au robot.</p>

3. Modifier le programme précédent pour que le robot soit vert quand il avance et rouge à l'arrêt.

4. Modifier le programme précédent pour que le robot s'arrête en cas de soulèvement.

5. Modifier le programme précédent pour que le robot soit bleu en cas de soulèvement

6. Précisez dans chacun des cas les capteurs et actionneurs mobilisés.

Commentaires : Cet exercice très guidé et linéaire permet d'ajouter divers capteurs au programme. On peut terminer en faisant la comparaison avec un robot aspirateur ou un robot tondeuse, éventuellement questionner sur les capteurs et les actionneurs qu'il suffirait de rajouter pour aboutir à un tel robot. » p. 3-4

```
graph TD
    A[quand est cliqué] --> B[mettre ma variable à 1]
    B --> C[ répéter indéfiniment ]
    C --> D[ si ma variable = 1 alors ]
    D --> E[ moteur tous à vitesse 50 ]
    D --> F[ sinon ]
    F --> G[ moteur tous à vitesse 0 ]
    C --> H[ quand un objet est détecté devant ]
    H --> I[ mettre ma variable à 0 ]
    I --> J[ quand aucun objet n'est détecté devant ]
    J --> K[ mettre ma variable à 1 ]
```

### Ingrédients élémentaires :

Manipuler une variable de type entier

Attendre la survenue d'un évènement x3

Modifier la valeur d'une variable (affectation) x3

Utiliser la valeur d'une variable

Utiliser une boucle non-for

Utiliser une structure conditionnelle à deux branches

Tester une égalité sur des nombres

Agir sur le robot avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x2

### Exercice 3.3



```
quand cliqué
mettre ma variable à 1
répéter indéfiniment
  si ma variable = 1 alors
    moteur tous à vitesse 50
    LED tout R: 0 V: 255 B: 0
  sinon
    moteur tous à vitesse 0
    LED tout R: 255 V: 0 B: 0
quand un objet est détecté devant
mettre ma variable à 0
quand aucun objet n'est détecté devant
mettre ma variable à 1
```

Ingrédients élémentaires :

	<p>Modifier la valeur d'une variable (affectation) x3</p> <p>Attendre la survenue d'un évènement x3</p> <p>Utiliser la valeur d'une variable</p> <p>Utiliser une boucle non-for</p> <p>Utiliser une structure conditionnelle à deux branches</p> <p>Tester une égalité sur des nombres</p> <p>Agir sur le robot avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x4</p> <p><u>Exercice 3.4 :</u></p>	
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

The image shows a Scratch script with the following blocks:

- when green flag clicked
- set variable 'ma variable' to 1
- repeat indefinitely loop:
  - if 'ma variable' = 1 then:
    - motor 'tous' at speed 50
    - LED 'tout' R: 0 V: 255 B: 0
  - otherwise:
    - motor 'tous' at speed 0
    - LED 'tout' R: 255 V: 0 B: 0


  

- when object detected in front:
  - set variable 'ma variable' to 0
- when no object detected in front:
  - set variable 'ma variable' to 1
- when no object detected below:
  - set variable 'ma variable' to 0

Ingrédients élémentaires :

Attendre la survenue d'un évènement x4

Modifier la valeur d'une variable (affectation) x4

	<p>Utiliser la valeur d'une variable</p> <p>Utiliser une boucle non-for</p> <p>Utiliser une structure conditionnelle à deux branches</p> <p>Tester une égalité sur des nombres</p> <p>Agir sur le robot avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x4</p>	
<p><b>Ingrédient 3 : Programmer un projet libre au choix</b></p> <p>« Exercice 4 - Exercices libres non guidés</p> <p>Choisir, au choix, l'un ou plusieurs des projets suivants, préciser à chaque fois les capteurs et actionneurs mobilisés.</p> <ol style="list-style-type: none"> <li><b>Robot aspirateur.</b> Le robot avance continuellement. Il devra reculer en cas de détection d'obstacle puis légèrement tourner et tenter de nouveau d'avancer.</li> <li><b>Robot sauvage.</b> Le robot avance continuellement et reste allumé en vert. Il devra reculer en cas de détection d'une main et se mettre en rouge. Puis après quelques secondes, tenter d'avancer de nouveau. Après plusieurs détections, il devra être apprivoisé : s'arrêter devant la main et passer vert.</li> <li><b>Robot taureau.</b> Le robot est à l'arrêt allumé en rouge. Il devra, en cas de détection d'un obstacle passé en rouge et avancer vers cet obstacle. Si l'obstacle disparaît, il devra repasser en rouge et s'arrêter.</li> </ol>	<p><b>Exercice 4.2 :</b></p>  <p><b>Ingrédients élémentaires :</b></p> <p>Déclarer une variable locale x3</p>	<p>Ces projets sont plus libres</p>

Commentaires : Cet exercice propose **trois projets plus libres** parmi une vaste étendue de projets possibles. Il est possible d'en inventer à l'infini : seule l'imagination impose des limites. » p. 5

```

quand est cliqué
mettre detection à 0
mettre apprivoise à 0
mettre hors-sol à 1
répéter indéfiniment
si hors-sol = 0 alors
  LED tout R: 0 V: 0 B: 255
  moteur tous à vitesse 0
sinon
  si detection = 1 et apprivoise < 10 alors
    moteur tous à vitesse
    distance devant + 10 / 19 + -100
  LED tout R: 255 V: 0 B: 0
  sinon
    moteur tous à vitesse 50
    LED tout R: 0 V: 255 B: 0
quand un objet est détecté devant
mettre detection à 1
ajouter 1 à apprivoise
quand aucun objet n'est détecté devant
mettre detection à 0
quand un objet est détecté dessous
mettre hors-sol à 1
quand aucun objet n'est détecté dessous
mettre hors-sol à 0

```

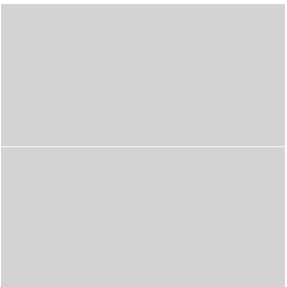
Ingrédients élémentaires :

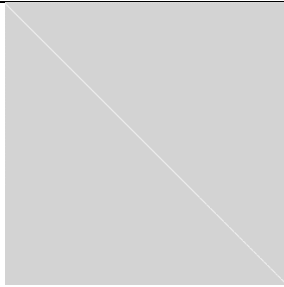


	<p>Manipuler une variable de type entier x3</p> <p>Modifier la valeur d'une variable (affectation) x7</p> <p>Modifier la valeur d'une variable (ajout)</p> <p>Utiliser la valeur d'une variable x3</p> <p>Utiliser une boucle non-for</p> <p>Utiliser une structure conditionnelle à deux branches x2</p> <p>Tester une égalité sur des nombres x2</p> <p>Tester une inégalité sur des nombres</p> <p>Additionner des nombres</p> <p>Multiplier des nombres</p> <p>Diviser des nombres</p> <p>Utiliser un connecteur logique « et »</p> <p>Attendre la survenue d'un évènement x5</p>	
--	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

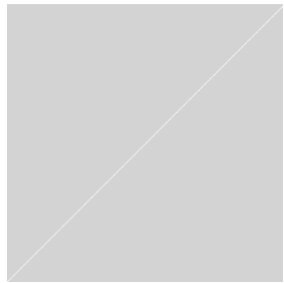
	<p>Agir sur le robot avec paramètre (utiliser une méthode d'un objet avec paramètre sans retour) x6</p>	
<p><b>Activité 3 : construire une image numérique à l'aide d'un programme</b></p> <p>(MEN 2019 - Apprendre à manipuler une image numérique - <a href="https://bit.ly/3jfldBy">https://bit.ly/3jfldBy</a>)</p> <p>« <i>Contenus et capacités</i></p> <p><i>Notions transversales de programmation. Contenus : Affectations, variables ; Séquences ; Instructions conditionnelles ; Boucles fors et non-fors ; Définitions et appels de fonctions. Capacités attendues : Écrire et développer des programmes pour répondre à des problèmes et modéliser des phénomènes physiques, économiques et sociaux.</i></p> <p><i>La photographie numérique. Contenus Traitement d'image Capacités attendues : Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels. » p. 1</i></p> <p>« <i>Note d'intention : [...] la première consiste à se familiariser avec la bibliothèque PIL en présentant les instructions de base tout en revisitant la double boucle. [...] » p 1-2</i></p>		<p>Permet de travailler les éléments du programme :</p> <ul style="list-style-type: none"> <li>- Écrire et développer des programmes pour répondre à des problèmes et modéliser des phénomènes physiques, économiques et sociaux</li> <li>- Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels</li> </ul> <p>Permet de se familiariser avec la bibliothèque de traitement d'image PIL</p> <p>Permet de réinvestir la notion de double boucle</p>
<p><b>Partie 1 : prendre en main l'environnement</b></p> <p>« <i>Exercice 1 - dans l'interpréteur d'EduPython</i></p> <p>1. Ouvrir un nouveau fichier python avec EduPython que vous appellerez <code>depart.py</code>. Sauvegarder le dans votre dossier personnel, puis exécuter le.</p>	<p><b>Exercice 1 :</b></p> <pre>1 from PIL import Image 2 im = Image.new("RGB", (600,300), "grey") 3 print(type(im)) 4 im.save("trace.png", "PNG")</pre> <p><b>Ingrédients élémentaires :</b></p>	

<p>2. Dans l'interpréteur d'EduPython taper :</p> <pre>1 from PIL import Image 2 im = Image.new("RGB", (600,300), "grey") 3 print(type(im)) 4 im.save("trace.png", "PNG")</pre> <p>3. Préciser à quoi sert la fonction <code>Image.new</code> et expliquer comment elle fonctionne. Ne pas hésiter à regarder ce qu'il se passe dans le dossier contenant le fichier <code>depart.py</code>.</p> <p>4. Préciser à quoi sert la méthode <code>.save</code> et expliquer comment elle fonctionne. » p. 2</p> <p>« Exercice 2 - toujours dans l'interpréteur d'EduPython</p> <p>1. Dans l'interpréteur python taper :</p> <pre>1 im.putpixel((10,10), (0,0,0)) 2 im.save("trace.png", "PNG")</pre> <p>2. 2. Préciser à quoi sert la méthode <code>.putpixel</code> et expliquer comment elle fonctionne.</p> <p>3. 3. Indiquer pourquoi <code>.save</code> il indispensable. » p. 2</p>	<p>Importer une bibliothèque</p> <p>Modifier la valeur d'une variable (affectation)</p> <p>Utiliser la valeur d'une variable</p> <p>Manipuler une variable de type objet « Image » (image)</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une fonction avec paramètres et sans retour</p> <p><u>Exercice 2 :</u></p> <pre>1 im.putpixel((10,10), (0,0,0)) 2 im.save("trace.png", "PNG")</pre> <p><u>Ingrédients élémentaires :</u></p> <p>Manipuler une variable de type objet « Image » (image)</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p>	
<p><b>Partie 2 : construire des images simples</b></p> <p>« Exercice 3 - premier programme</p> <p>1. Dans le bloc note d'Edupython, écrire un programme qui créera sur le disque dur une image toute jaune de taille 800 par 800 pixels. Cette image devra s'appeler <code>premiere_image.png</code></p>		<p>Permet de travailler la décomposition RGB</p>

<p>2. Modifier le programme pour que l'image contienne un pixel noir, un bleu, un rouge, un vert et un blanc.</p> <p>3. Modifier votre programme pour que votre image contienne un pixel de couleur « skyblue ».</p> <p>4. Comparer le résultat si nous avons décidé d'appeler notre image <code>premiere_image.jpg</code>. Expliquer pourquoi les résultats sont différents.</p> <p>Commentaires : Ici l'élève est invité à manipuler les commandes qu'il vient de voir. <b>La décomposition d'une couleur selon le codage (RGB) est travaillée [...].</b> » p. 3</p>		
<p><b>Partie 3 : construire des image plus complexes</b></p> <p>« Exercice 4 : À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite. <b>L'objectif est d'obtenir un résultat comme ci-dessous</b> : » p. 3</p>  <p>« Exercice 5 : À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite. <b>L'objectif est d'obtenir un résultat comme ci-dessous</b> : » p. 4</p>	<p><b>Exercice 4 :</b></p> <pre> 1 from PIL import Image 2 im = Image.new("RGB", (800,800)\ 3               , "LightGray") 4 for k in range (800): 5     im.putpixel ((k,400)\ 6                 , (255,255,255)) 7 im.save("image.png", "png") </pre> <p><b>Ingrédients élémentaires :</b></p> <ul style="list-style-type: none"> <li>Importer une bibliothèque</li> <li>Modifier la valeur d'une variable (affectation)</li> <li>Utiliser la valeur d'une variable</li> <li>x3</li> <li>Manipuler une variable de type objet « Image » (image)</li> </ul>	



« Exercice 6 : À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite. **L'objectif est d'obtenir un résultat comme ci-dessous** : » p.4



« Exercice 7 : À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite en pointillé. **L'objectif est d'obtenir un résultat comme ci-dessous** : » p. 5



Utiliser une boucle for avec utilisation de la variable de boucle

Utiliser une méthode d'un objet avec paramètre et sans retour x2

Utiliser une méthode d'un objet avec paramètre et avec retour

Exercice 5 :

```
1 from PIL import Image
2 im = Image.new("RGB", (800,800) \
3               , "LightGray")
4 for k in range (800):
5     im.putpixel((k,k) \
6                , (255,255,255))
7 im.save("image3.png", "png")
```

Ingrédients élémentaires :

Importer une bibliothèque

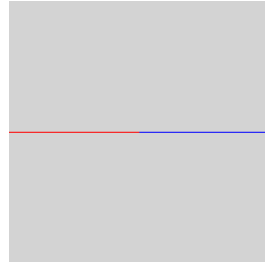
Modifier la valeur d'une variable (affectation)

Utiliser la valeur d'une variable x4

Manipuler une variable de type objet « Image » (image)

Utiliser une boucle for avec utilisation de la variable de boucle

« Exercice 8 : À partir d'une image carrée de taille 800 par 800 dont le fond est gris, tracer un segment de droite bicolore rouge/bleu. L'objectif est d'obtenir un résultat comme ci-dessous : » p. 5



Utiliser une méthode d'un objet avec paramètre et sans retour x2

Utiliser une méthode d'un objet avec paramètre et avec retour

Exercice 6 :

```
1 from PIL import Image
2 im = Image.new("RGB", (800, 800) \
3               , "LightGray")
4 for k in range (800) :
5     im.putpixel ((799-k, k) \
6                 , (255, 255, 255))
7 im.save("image4.png", "png")
```

Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation)

Utiliser la valeur d'une variable x4

Manipuler une variable de type objet « Image » (image)

Soustraire des nombres

Utiliser une boucle for avec utilisation de la variable de boucle

Utiliser une méthode d'un objet avec paramètre et sans retour x2

Utiliser une méthode d'un objet avec paramètre et avec retour

Exercice 7 :

```
1 from PIL import Image
2 im = Image.new("RGB", (800,800)\
3               , "LightGray")
4 for k in range (0,800,2):
5     im.putpixel((k,400)\
6               , (255,255,255))
7 im.save("image5.png", "png")
```

Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation)

Utiliser la valeur d'une variable  
x3

Manipuler une variable de type objet « Image » (image)

Utiliser une boucle for avec utilisation de la variable de boucle

Utiliser une méthode d'un objet avec paramètre et sans retour x2

Utiliser une méthode d'un objet avec paramètre et avec retour

Exercice 8 : Technique 1

```
1 from PIL import Image
2 im = Image.new("RGB", (800,800) \
3               , "LightGray")
4 for k in range (400):
5     im.putpixel((k,400)\
6               , (255,0,0))
7 for k in range (400 ,800) :
8     im.putpixel((k,400)\
9               , (0,0,255))
10 im.save("image6.png", "png")
```

### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable  
(affectation)

Utiliser la valeur d'une variable  
x5

Manipuler une variable de type  
objet « Image » (image)

Utiliser une boucle for avec  
utilisation de la variable de  
boucle x2

Utiliser une méthode d'un objet  
avec paramètre et sans retour x3

Utiliser une méthode d'un objet  
avec paramètre et avec retour

Exercice 8 : Technique 2



```

1 from PIL import Image
2 im = Image.new("RGB", (800,800)\
3               , "LightGray")
4 for k in range (800):
5     if k<400:
6         im.putpixel((k,400)\
7                   , (255,0,0))
8     else :
9         im.putpixel((k,400)\
10                  , (0,0,255))
11 im.save("image6.png", "png")

```

### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable  
(affectation)

Utiliser la valeur d'une variable  
x5


Manipuler une variable de type  
objet « Image » (image)

Utiliser une boucle for avec  
utilisation de la variable de  
boucle

Utiliser une structure  
conditionnelle à deux branches

Tester une inégalité sur des  
nombres

Utiliser une méthode d'un objet  
avec paramètre et sans retour x3

	<p>Utiliser une méthode d'un objet avec paramètre et avec retour</p>	
<p><b>Ingrédient 4 : dessiner des drapeaux</b></p> <p>« Exercice 9 : Drapeaux. Dessiner les drapeaux suivants :</p> <div data-bbox="324 399 974 598" style="text-align: center;">  </div> <p>Commentaires : L'objectif de cet exercice est bien évidemment le drapeau français et la double boucle pour. Il est possible de préciser que le code couleur du bleu et du rouge ont été normalisés et se trouvent facilement sur le web. » p. 6-7</p>	<p><b>Drapeau français :</b></p> <pre> 1 from PIL import Image 2 im =Image.new("RGB", (900 ,450)\ 3               , "grey") 4 for l in range (450): 5     for c in range (900): 6         if c&lt;300: 7             im.putpixel((c,l)\ 8                           , (5,20,64)) 9         elif c&lt;600: 10            im.putpixel((c,l)\ 11                          , (255,255,255)) 12        else : 13            im.putpixel((c,l)\ 14                          , (236,25,32)) 15 im.save("image8.png", "png") </pre> <p><b>Ingrédients élémentaires :</b></p> <ul style="list-style-type: none"> <li>Importer une bibliothèque</li> <li>Modifier la valeur d'une variable (affectation)</li> <li>Utiliser la valeur d'une variable</li> <li>x10</li> <li>Manipuler une variable de type objet « Image » (image)</li> <li>Tester une inégalité sur des nombres x2</li> <li>Utiliser une boucle for avec utilisation de la variable de boucle x2</li> </ul>	<p>L'objectif est la double boucle pour</p>

	<p>Utiliser une structure conditionnelle à trois branches.</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x4</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p>	
<p><b>Activité 4 : programmer le traitement d'une image numérique</b></p> <p>(MEN 2019 – Traitement d'image - <a href="https://bit.ly/3ogN895">https://bit.ly/3ogN895</a>)</p> <p>« <i>Contenus et capacités :</i></p> <p><i>Notions transversales de programmation. Contenus : Affectations, variables ; Séquences ; Instructions conditionnelles ; Boucles fors et non-fors ; Définitions et appels de fonctions. Capacités attendues : Écrire et développer des programmes pour répondre à des problèmes et modéliser des phénomènes physiques, économiques et sociaux.</i></p> <p><i>La photographie numérique. Contenus Traitement d'image Capacités attendues : Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels. » p. 1</i></p> <p>« La seconde partie se focalise sur la retouche d'image : négatif en noir et blanc. Les plus rapides pouvant tout à fait aller plus loin. » p. 1</p>		<p>Permet de travailler les éléments du programme :</p> <ul style="list-style-type: none"> <li>- Écrire et développer des programmes pour répondre à des problèmes et modéliser des phénomènes physiques, économiques et sociaux</li> <li>- Traiter par programme une image pour la transformer en agissant sur les trois composantes de ses pixels</li> </ul>
<p><b>Partie 1 : manipuler un fichier image avec Python</b></p> <p>« <i>Exercice 1 - quelques commandes en plus</i></p>	<p><b>Exercice 1 :</b></p> <pre>1 from PIL import Image 2 img=Image.open("joconde_384.jpg") 3 img.save("resultat.jpg", "JPEG")</pre>	<p>Permet d'apprendre à manipuler une image dans Python</p>

<p>1. Dans votre espace personnel, créer un dossier contenant toutes les images et le fichier <code>retouche.py</code>.</p> <p>2. Ouvrir le fichier <code>retouche.py</code> avec EduPython et enregistrer le dans le même dossier que celui qui contient les quatre images.</p> <pre>1 from PIL import Image 2 img=Image.open("joconde_384.jpg") 3 img.save("resultat.jpg", "JPEG")</pre> <p>3. Après avoir exécuté le fichier <code>retouche.py</code>, déterminer ce que fait la commande <code>.open</code></p> <p>4. Préciser le type de <code>img</code>.</p> <p>Commentaires : Cet exercice présente la méthode <code>.open</code> permettant d'ouvrir une image du disque dur et de l'importer dans une <code>ImageFile</code> manipulable en python. » p. 2</p>	<p><u>Ingrédients élémentaires :</u></p> <p>Importer une bibliothèque</p> <p>Modifier la valeur d'une variable (affectation)</p> <p>Utiliser la valeur d'une variable</p> <p>Manipuler une variable de type objet « <code>ImageFile</code> » (fichier image)</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour</p> <p>Utiliser une fonction avec paramètre et avec retour</p>	
<p><b>Partie 2 : manipuler les pixels d'un fichier image avec Python</b></p> <p>« Exercice 2 - papillon</p> <p>Dans cet exercice nous allons travailler sur l'image <code>papillon.png</code> qui est en noir et blanc.</p> <p>1. Préalable : Modifier le programme <code>retouche.py</code> de manière à importer le papillon dans une <code>ImageFile</code> que nous appellerons <code>pap</code>.</p> <pre>1 from PIL import Image 2 pap=Image.open("papillon.png")</pre> <p>2. Activité : Exécuter le programme <code>retouche.py</code>. Grâce à l'interpréteur, déterminer ce que font les commandes suivantes :</p>	<p><u>Exercice 2.1 :</u></p> <pre>1 from PIL import Image 2 pap=Image.open("papillon.png")</pre> <p><u>Ingrédients élémentaires :</u></p> <p>Importer une bibliothèque</p> <p>Modifier la valeur d'une variable (affectation)</p> <p>Manipuler une variable de type objet « <code>ImageFile</code> » (fichier image)</p> <p>Utiliser une fonction avec paramètre et avec retour</p>	<p>Permet de présenter la méthode <code>getPixel</code> et l'attribut <code>mode</code> relatif à l'encodage des pixels</p>

```

4 print(pap.format)
5 print(pap.size)
6 pap.show()
7 print(pap.getpixel((3,7)))
8 pap.putpixel((3,7),128)
9 print(pap.mode)
10 pap2 = pap.copy()

```

Commentaires : **Présentation de la méthode getpixel et de l'attribut mode permettant de connaître la manière dont sont encodés les pixels** (L : niveau de gris, RGB pour un codage couleur (R,G,B)) » p. 2-3

### Exercice 2.2 :

```

4 print(pap.format)
5 print(pap.size)
6 pap.show()
7 print(pap.getpixel((3,7)))
8 pap.putpixel((3,7),128)
9 print(pap.mode)
10 pap2 = pap.copy()

```

### Ingrédients élémentaires :

Modifier la valeur d'une variable (affectation)

Utiliser la valeur d'une variable x7

Manipuler une variable de type objet « ImageFile » (fichier image)

Utiliser une méthode d'un objet sans paramètre et sans retour

Utiliser une méthode d'un objet sans paramètre et avec retour x3

Utiliser une méthode d'un objet avec paramètre et sans retour

Utiliser une méthode d'un objet avec paramètre et avec retour x2

Utiliser une fonction avec paramètres et sans retour x4

### Partie 3 : programmer le passage en négatif d'une image en noir et blanc

« Exercice 3 - Négatif d'une image en noir et blanc

1. Effectuer le négatif de l'image papillon.png.
2. Effectuer le négatif de l'image Joconde\_10.png.
3. Effectuer le négatif de l'image Joconde\_gris.png. » p. 3

« Commentaires : C'est l'exercice « cœur de séance » : construire le négatif d'une image. La première image est une image 10 par 10 en noir et blanc sans niveau de gris (0 ou 255). Des élèves vont le faire pixel par pixel sans utiliser `getpixel` mais en regardant l'image. Il est possible dans un premier temps de les laisser faire et de les questionner par la suite pour une grosse image (question 3). D'autres élèves utiliseront `getpixel` combiné à un `if` (si `val = 255` alors mettre 0 sinon mettre 255). Il ne faut pas dans la question 1 s'attendre à ce que tous les élèves pensent à faire `255 - val`. C'est l'objectif de la question 2. La deuxième image est une image 10 par 10 de la Joconde en niveau de gris. Il y a ici l'obligation d'utiliser la formule `255 - val`. La double boucle peut alors être rappelée à tous. La troisième image est beaucoup plus grosse. Mais toutefois, le programme est identique. Volontairement dans cet exercice, toutes les images sont carrées afin de ne pas perturber les élèves (pas de différenciation entre hauteur et largeur). L'objectif étant vraiment la double boucle pour et la gestion du négatif. » p. 4

### Exercice 3.1 :

```
1 from PIL import Image
2 pap=Image.open("papillon.png")
3 for l in range(10):
4     for c in range(10):
5         valeur=pap.getpixel((c,l))
6         if valeur==0:
7             pap.putpixel((c,l),255)
8         else :
9             pap.putpixel((c,l),0)
10 pap.save("image.png","png")
```

### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x2

Utiliser la valeur d'une variable x4

Manipuler une variable de type « int » (entier)

Manipuler une variable de type objet « ImageFile » (fichier image)

Utiliser une boucle for avec utilisation de la variable de boucle x2

Utiliser une structure conditionnelle à deux branches.

L'objectif est la double boucle et la gestion du négatif.

Les images sont carrées pour ne pas perturber les élèves.

Dans la question 2, l'image étant en niveau de gris, l'objectif est que les élèves utilisent `255 - val`

Tester une égalité sur des nombres

Utiliser une méthode d'un objet avec paramètre et sans retour x3

Utiliser une méthode d'un objet avec paramètre et avec retour x2

Utiliser une fonction avec paramètres et avec retour

Exercice 3.2 :

```
1 from PIL import Image
2 joco = Image.open("Joconde_10.png")
3 for l in range(10):
4     for c in range(10):
5         valeur=joco.getpixel((c,l))
6         nouvelle_valeur=255-valeur
7         joco.putpixel((c,l)\
8             , nouvelle_valeur)
9 joco.save("image3.png", "png")
```

Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x3

Utiliser la valeur d'une variable x4

Manipuler une variable de type « int » (entier) x2

Manipuler une variable de type objet « ImageFile » (fichier image)

Utiliser une boucle for avec utilisation de la variable de boucle x2

Soustraire des nombres

Utiliser une méthode d'un objet avec paramètre et sans retour x2

Utiliser une méthode d'un objet avec paramètre et avec retour

Utiliser une fonction avec paramètres et avec retour

Exercice 3.3 :

```
1 from PIL import Image
2 joco=Image.open("Joconde_gris.png")
3 for l in range(384):
4     for c in range(384):
5         valeur=joco.getpixel((c,l))
6         nouvelle_valeur=255-valeur
7         joco.putpixel((c,l)\
8             ,nouvelle_valeur)
9 joco.save ("image4.png","png")
```

Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x3



	<p>Utiliser la valeur d'une variable x4</p> <p>Manipuler une variable de type « int » (entier) x2</p> <p>Manipuler une variable de type objet « ImageFile » (fichier image)</p> <p>Utiliser une boucle for avec utilisation de la variable de boucle x2</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Soustraire des nombres</p> <p>Utiliser une fonction avec paramètres et avec retour</p>	
<p><b>Ingrédient 4 : Manipuler les pixels d'une image en couleur avec Python</b></p> <p>« Exercice 4 - comprendre les images couleurs</p> <ol style="list-style-type: none"> <li>1. Importer Joconde_384.jpg dans une ImageFile que nous appellerons joco.</li> <li>2. Exécuter le fichier.</li> <li>3. Tester et comprendre les commandes suivantes dans l'interpréteur :</li> </ol>	<p><b>Exercice 4.3 :</b></p> <pre> 1 from PIL import Image 2 joco=Image.open("Joconde_384.png") 3 valeurs=joco.getpixel((3,7)) 4 print(valeurs) 5 print(valeurs[0]) 6 (R,G,B)=joco.getpixel((3,7)) 7 print(G) </pre> <p><b>Ingrédients élémentaires :</b></p> <p>Importer une bibliothèque</p>	<p>La méthode de récupération des couleurs dans un tuple est plus naturelle pour les élèves.</p>

<pre> 1 from PIL import Image 2 joco=Image.open("Joconde_384.png") 3 valeurs=joco.getpixel((3,7)) 4 print(valeurs) 5 print(valeurs[0]) 6 (R,G,B)=joco.getpixel((3,7)) 7 print(G) </pre> <p>4. <b>Expliquer comment fonctionne <code>joco.getpixel</code>.</b> » p. 4</p> <p>« Commentaires : <b>Sont introduits ici les images couleurs et le <code>.getpixel</code> les manipulant de deux manières différentes.</b> Pour les élèves manipuler la récupération en écrivant <code>(R,G,B) = joco.getpixel((3,7))</code> est plus naturelle. » p. 4</p>	<p>Modifier la valeur d'une variable (affectation) x3</p> <p>Utiliser la valeur d'une variable x5</p> <p>Manipuler une variable de type « int » (entier)</p> <p>Manipuler une variable de type « tuple » (n-uplet)</p> <p>Accéder à un élément d'un « tuple » (n-uplet)</p> <p>Manipuler une variable de type objet « ImageFile » (fichier image)</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour x2</p> <p>Utiliser une fonction avec paramètres et avec retour</p> <p>Utiliser une fonction avec paramètres et sans retour x3</p>	
<p><b>Ingrédient 5 : Programmer la transformation d'une image couleur en noir et blanc.</b></p> <p>« Exercice 5 - transformer une image couleur en noir et blanc.</p> <p>1. Traiter <code>Joconde_384.jpg</code> en remplaçant, pixel par pixel, chacune des valeurs des composantes des couleurs par la moyenne des trois.</p>	<p><u>Exercice 5.1 :</u></p>	

2. Dans *Joconde\_384.pgm*, toujours pixel par pixel, remplacer chaque composante des couleurs par la valeur commune :

$$0,21 \times R + 0,71 \times G + 0,07 \times B$$

3. Répéter la même opération mais en prenant cette fois-ci :

$$0,299 \times R + 0,587 \times G + 0,114 \times B$$

4. Comparer les trois images obtenues. » p. 4

« Commentaires : Exercice classique et basique de manipulation des images couleurs. » p. 5

```
1 from PIL import Image
2 joco = Image.open("Joconde_384.jpg")
3 for l in range(384):
4     for c in range(384):
5         valeurs=joco.getpixel((c,l))
6         nouveaux=(valeurs[0]\
7                 +valeurs[1]\
8                 +valeurs[2])//3
9         joco.putpixel ((c,l)\
10                        ,(nouveaux,nouveaux\
11                          ,nouveaux))
12 joco.save ("image6.png","png")
```

Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x3

Utiliser la valeur d'une variable x9

Manipuler une variable de type « int » (entier)

Manipuler une variable de type objet « ImageFile » (fichier image)

Manipuler une variable de type « tuple » (n-uplet)

Accéder à un élément d'un « tuple » (n-uplet) x3

Utiliser une boucle for avec utilisation de la variable de boucle x2

Utiliser une méthode d'un objet avec paramètre et sans retour x2

Utiliser une méthode d'un objet avec paramètre et avec retour

Utiliser une fonction avec paramètres et avec retour

### Exercice 5.2 :

```
1 from PIL import Image
2 joco = Image.open("Joconde_384.jpg")
3 for l in range(384):
4     for c in range(384):
5         valeurs=joco.getpixel((c,l))
6         nouveaux=0.21* valeurs[0]\
7             +0.71*valeurs[1]\
8             +0.07*valeurs[2]
9         nouveaux=int(nouveaux)
10        joco.putpixel((c,l),\
11            (nouveaux,nouveaux,\
12            nouveaux))
13 joco.save ("image7.png","png")
```

### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x4

Utiliser la valeur d'une variable x10

Manipuler une variable de type « int » (entier)

Manipuler une variable de type « float » (flottant)

	<p>Additionner des nombres x2</p> <p>Multiplier des nombres x3</p> <p>Manipuler une variable de type objet « ImageFile » (fichier image)</p> <p>Manipuler une variable de type « tuple » (n-uplet)</p> <p>Accéder à un élément d'un « tuple » (n-uplet) x3</p> <p>Utiliser une boucle for avec utilisation de la variable de boucle x2</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p> <p><u>Exercice 5.3 :</u></p>	
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

```

1 from PIL import Image
2 joco = Image.open("Joconde_384.jpg")
3 for l in range(384):
4     for c in range(384):
5         (R,G,B)=joco.getpixel((c,l))
6         nouveaux=0.21*R+0.71*G+0.07*B
7         nouveaux=int(nouveaux)
8         joco.putpixel((c,l)\
9             , (nouveaux,nouveaux\
10                ,nouveaux))
11 joco.save ("image8.png", "png")

```

### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable  
(affectation) x4

Utiliser la valeur d'une variable  
x10

Manipuler une variable de type  
« int » (entier)

Manipuler une variable de type  
« float » (flottant)

Additionner des nombres x2

Multiplier des nombres x3

Manipuler une variable de type  
objet « ImageFile » (fichier  
image)

Manipuler une variable de type  
« tuple » (n-uplet)

	<p>Accéder à un élément d'un « tuple » (n-uplet) x3</p> <p>Utiliser une boucle for avec utilisation de la variable de boucle x2</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p>	
<p><b>Ingrédient 6 : Programmer le passage en négatif d'une image en couleur.</b></p> <p>« Exercice 6 - négatif d'une image en couleurs</p> <p><b>Effectuer le négatif de l'image Joconde_384.jpg.</b> » p. 6</p> <p>« Commentaires : Exercice peu difficile car pas de nouvelles réflexions algorithmiques. Les problèmes pour les élèves viendront de la manipulation de la syntaxe en python. » p. 6</p>	<p><b>Exercice 6 :</b></p> <pre> 1 from PIL import Image 2 joco = Image.open("Joconde_384.jpg") 3 for l in range(384): 4     for c in range(384): 5         (R,G,B)=joco.getpixel((c,l)) 6         nouveaux=(255-R,255-G,255-B) 7         joco.putpixel((c,l),nouveaux) 8 joco.save("image9.png","png") </pre> <p><b>Ingrédients élémentaires :</b></p> <p>Importer une bibliothèque</p> <p>Modifier la valeur d'une variable (affectation) x5</p> <p>Utiliser la valeur d'une variable x5</p>	

	<p>Manipuler une variable de type objet « ImageFile » (fichier image)</p> <p>Manipuler une variable de type « tuple » (n-uplet) x2</p> <p>Utiliser une boucle for avec utilisation de la variable de boucle x2</p> <p>Soustraire des nombres x3</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p>	
<p><b>Ingrédient 7 : Programmer la construction du miroir d'une image</b></p> <p>« Exercice 7 - miroir</p> <ol style="list-style-type: none"> <li>1. Effectuer le miroir de l'image papillon.png.</li> <li>2. Effectuer le miroir de l'image Joconde_gris.png.</li> <li>3. Effectuer le miroir de l'image Joconde_384.jpg. » p.6</li> </ol> <p>« Commentaires : La difficulté de cet exercice vient du fait que les élèves modifient leur image de départ. Ainsi, arrivés à la moitié, ils ont perdu l'information de la partie gauche. Il est alors nécessaire de faire une copie de l'image de départ, ce qui constitue une réelle</p>	<p><b>Exercice 7.1 :</b></p> <pre> 1 from PIL import Image 2 pap=Image.open("papillon.png") 3 pap2=pap.copy() 4 for l in range(10): 5     for c in range(10): 6         valeur=pap.getpixel(\ 7             (c,l)) 8         pap2.putpixel((9-c,l)\ 9             ,valeur) 10 pap2.save ("image10.png", "png") </pre> <p><b>Ingrédients élémentaires :</b></p> <p>Importer une bibliothèque</p>	<p>Il faut faire une copie de l'image de départ pour ne pas perdre l'information en cours de traitement.</p>



<p><i>difficulté pour eux, au même titre que travailler avec les indices.</i> » p. 7</p>	<p>Modifier la valeur d'une variable (affectation) x3</p> <p>Utiliser la valeur d'une variable x4</p> <p>Manipuler une variable de type « tuple » (n-uplet)</p> <p>Manipuler une variable de type objet « ImageFile » (fichier image) x2</p> <p>Utiliser une boucle for avec utilisation de la variable de boucle x2</p> <p>Soustraire des nombres</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Utiliser une méthode d'un objet sans paramètre et avec retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p> <p><u>Exercice 7.2 :</u></p>	
------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

```

1 from PIL import Image
2 joco=Image.open("Joconde_gris.png")
3 joco2=joco.copy()
4 for l in range(384):
5     for c in range(384):
6         valeur=joco.getpixel((c,l))
7         joco2.putpixel((383-c,l)\
8             ,valeur)
9 joco2.save ("image11.png","png")

```

### Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable  
(affectation) x3

Utiliser la valeur d'une variable  
x4

Manipuler une variable de type  
« tuple » (n-uplet)

Manipuler une variable de type  
objet « ImageFile » (fichier  
image) x2

Utiliser une boucle for avec  
utilisation de la variable de  
boucle x2

Soustraire des nombres

Utiliser une méthode d'un objet  
avec paramètre et sans retour x2

Utiliser une méthode d'un objet  
avec paramètre et avec retour

Utiliser une méthode d'un objet sans paramètre et avec retour

Utiliser une fonction avec paramètres et avec retour

Exercice 7.3 :

```
1 from PIL import Image
2 joco=Image.open("Joconde_384.jpg")
3 joco2=joco.copy()
4 for l in range(384):
5     for c in range(384):
6         valeur=joco.getpixel(\
7             (c,l))
8         joco2.putpixel((383-c,l)\
9             ,valeur)
10 joco2.save("image12.png","png")
```

Ingrédients élémentaires :

Importer une bibliothèque

Modifier la valeur d'une variable (affectation) x3

Utiliser la valeur d'une variable x4

Manipuler une variable de type « tuple » (n-uplet)

Manipuler une variable de type objet « ImageFile » (fichier image) x2

	<p>Utiliser une boucle for avec utilisation de la variable de boucle x2</p> <p>Soustraire des nombres</p> <p>Utiliser une méthode d'un objet avec paramètre et sans retour x2</p> <p>Utiliser une méthode d'un objet avec paramètre et avec retour</p> <p>Utiliser une méthode d'un objet sans paramètre et avec retour</p> <p>Utiliser une fonction avec paramètres et avec retour</p>	
--	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--

# B Questionnaire enseignants

## B.1 Préambule

### Préambule

Cette enquête est réalisée dans le cadre d'un travail de recherche sur l'enseignement-apprentissage de la programmation informatique dans l'enseignement secondaire. L'objectif est d'établir un état des lieux des pratiques d'enseignement de la programmation au moment spécifique de la transition collège-lycée. Ce travail est mené dans le cadre d'un doctorat effectué à l'Université de Bretagne Occidentale et co-financé par la région Bretagne.

### Quelques précisions

Ce questionnaire s'adresse aux professeurs :

- de mathématiques au niveau troisième
- de technologie au niveau troisième
- de mathématiques au niveau seconde
- de Sciences Numérique et Technologie (SNT) au niveau seconde

Répondre à ce questionnaire vous prendra environ 15 minutes.

Il s'agit ici d'une démarche purement informative au service de la recherche et soumise à aucun jugement ni à aucune évaluation.

### Situation sanitaire

Cette enquête porte sur vos pratiques d'enseignement habituelles, c'est à dire hors situation sanitaire exceptionnelle liée à la COVID19. Il vous est demandé d'essayer, dans la mesure du possible, de faire abstraction des adaptations rendues nécessaires par cette situation particulière et que vous espérez abandonner dès que possible. En revanche, si vous pensez conserver certaines de ces pratiques lorsque la situation reviendra à la normale, vous pouvez les évoquer. Une question en fin de questionnaire portant spécifiquement sur ce thème vous permettra, si vous le souhaitez, de préciser les conséquences de la COVID19 sur votre enseignement.

### Confidentialité

Ce questionnaire est strictement anonyme, il ne recueille donc aucune donnée personnelle (nom, prénom, date de naissance, établissement scolaire, adresse IP, etc.). Par conséquent, il est conforme aux textes encadrant l'utilisation des données personnelles en France :

- la loi n° 2018-493 du 20 juin 2018, qui modifie la loi Informatique et Libertés du 6 janvier 1978 ;
- le Règlement Général sur la Protection des Données (RGPD), voté par le Parlement Européen et le Conseil Européen en 2016, et entré en application le 25 mai 2018.

### Consentement du participant

En cliquant sur le bouton « Suivant », vous indiquez :

- avoir pris connaissance de cette page d'information ;
- être d'accord pour participer

## B.2 Groupe A : informations générales

**QA1\*** : *Quelle est votre ancienneté dans le métier d'enseignant ?*

Liste déroulante choix unique

- Moins d'un an
- Entre 1 an et 5 ans
- Entre 5 ans et 15 ans
- Entre 15 ans et 25 ans
- Entre 25 ans et 35 ans
- Plus de 35 ans

**QA2\*** : *Quelle est votre discipline d'enseignement principale en nombre d'heure ?*

Liste déroulante choix unique

- Arts plastiques
- Éducation Physique et Sportive (EPS)
- Éducation musicale
- Enseignement scientifique
- Français
- Histoire-géographie
- Langues vivantes
- Mathématiques
- Philosophie
- Physique-Chimie
- Sciences de la Vie et de la Terre (SVT)
- Sciences de l'Ingénieur (SI)
- Sciences Économiques et Sociales (SES)
- Sciences Numériques et Technologie (SNT)
- Technologie
- Autre / Préciser

**QA3\*** : *À quel(s) niveau(x) enseignez-vous cette discipline cette année ?*

Cases à cocher choix multiple

- Sixième
- Cinquième
- Quatrième
- Troisième
- Seconde
- Première
- Terminale
- autre / Préciser

**QA4\*** : *Avez-vous une discipline d'enseignement secondaire ?*

*Boutons radio* choix unique

- Oui / Non

[SI QA4 == Oui]

**QA5\*** : *Quelle est votre discipline d'enseignement secondaire ?*

Liste déroulante choix unique

- Arts plastiques
- Éducation Physique et Sportive (EPS)
- Éducation musicale
- Enseignement scientifique
- Français
- Histoire-géographie
- Langues vivantes
- Mathématiques
- Philosophie
- Physique-Chimie
- Sciences de la Vie et de la Terre (SVT)
- Sciences de l'Ingénieur (SI)
- Sciences Économiques et Sociales (SES)
- Sciences Numériques et Technologie (SNT)
- Technologie
- Autre / Préciser

[SI QA4 == Oui]

**QA6\*** : À quel(s) niveau(x) enseignez-vous cette discipline secondaire cette année ?

Cases à cocher choix multiple

- Sixième
- Cinquième
- Quatrième
- Troisième
- Seconde
- Première
- Terminale
- Autre / préciser

**QA7\*** : Concernant votre formation initiale, possédez-vous un ou plusieurs diplômes en lien avec la programmation informatique (au moins un module d'enseignement y était consacré)

Boutons radio choix unique

Oui / non

**QA8\*** : Avez-vous suivi une ou plusieurs formations continues relatives à la programmation informatique ?

Cases à cocher choix multiple

- Non aucune
- Formation académique
- Habilitation ISN
- DIU NSI
- Formation ClassCode
- Formation sur la plateforme M@gistère
- MOOC SNT de l'INRIA
- Autre : préciser

**QA9\*** : Comment qualifieriez-vous les éléments suivants ?

Donner pour chaque élément une estimation allant de « Faible » à « Important » en déplaçant le curseur.

- Votre usage des outils numériques au quotidien

Faible  Important



- Votre intérêt pour le numérique et les technologies en général

Faible  Important

- Vos connaissances en informatique en général

Faibles  Importantes

- Vos connaissances en programmation informatique

Faibles  Importantes

**QA10** : (Facultatif) Vous pouvez ajouter ci-dessous des commentaires ou des remarques sur cette partie « Informations générales »

Champ texte

### B.3 Groupe B : discipline et niveau d'enseignement de la programmation informatique

**QB1\*** : *Dans quelle discipline et à quel niveau enseignez-vous la programmation informatique cette année ? Choisir une possibilité parmi les quatre proposées ci-dessous. Vous répondrez ensuite aux questions qui vont suivre spécifiquement pour cette discipline et ce niveau*

Liste déroulante choix unique

- Mathématiques en troisième (MAT3)
- Technologie en troisième (TEC3)
- Mathématiques en seconde (MAT2)
- Sciences Numériques et Technologie (SNT) en seconde (SNT2)

*Aide : Si vous êtes concernés par plusieurs possibilités, faites le choix qui vous semble le plus pertinent en lien avec la programmation informatique.*

### B.4 Groupe C : contenus et connaissances

**QC1\*** : *Quelle distinction faites-vous avec vos élèves entre la notion d'algorithme et la notion de programme ? Je présente ces notions comme étant :*

Boutons radio choix unique

- Tout à fait similaires
- Proches
- Différentes
- Très différentes

[SI QB1 ==MAT3]

**QC2\*** : *Vous concertez-vous avec vos collègues professeurs de technologie concernant l'introduction ou l'utilisation des notions algorithmiques ?*

*Boutons radio choix unique*

- Non
- Oui et j'ai la charge d'introduire les notions algorithmiques
- Oui et ils ont la charge d'introduire les notions algorithmiques
- Oui et notre organisation est différente de celles citées ci-dessus

*Aide : si votre organisation varie selon vos classes et vos collègues, indiquer la situation la plus courante*

[SI QB1 ==TEC3]

**QC3\*** : *Vous concertez-vous avec vos collègues professeurs de mathématiques concernant l'introduction et l'utilisation des notions algorithmiques ?*

*Boutons radio choix unique*

- Non
- Oui et j'ai la charge d'introduire les notions algorithmiques
- Oui et ils ont la charge d'introduire les notions algorithmiques
- Oui et notre organisation est différente de celles citées ci-dessus

*Aide : si votre organisation varie selon vos classes et vos collègues, indiquer la situation la plus courante*

[SI QB1 ==MAT2]

**QC4\*** : *Vous concertez-vous avec vos collègues professeurs de SNT concernant l'approfondissement et l'utilisation des notions algorithmiques en Python ?*

*Boutons radio choix unique*

- Non
- Oui et j'ai la charge d'introduire la programmation en Python
- Oui et ils ont la charge d'introduire la programmation en Python
- Oui et notre organisation est différente de celles citées ci-dessus

*Aide : si votre organisation varie selon vos classes et vos collègues, indiquer la situation la plus courante*

[SI QB1 ==SNT2]

**QC5\*** : *Vous concertez-vous avec vos collègues professeurs de mathématiques concernant l'approfondissement et l'utilisation des notions algorithmiques en Python ?*

*Boutons radio choix unique*

- Non
- Oui et j'ai la charge d'introduire la programmation en Python
- Oui et ils ont la charge d'introduire la programmation en Python

- Oui et notre organisation est différente de celles citées ci-dessus

*Aide : si votre organisation varie selon vos classes et vos collègues, indiquer la situation la plus courante*

**QC6 :** *(Facultatif) Vous pouvez ajouter ci-dessous des commentaires ou des remarques sur cette partie « Contenus et connaissances »*

Champ texte

## B.5 Groupe D : activité d'enseignement

**QD1\*** : *Combien de séances d'enseignement consacrez-vous à la programmation informatique sur l'année scolaire pour la discipline et le niveau concernés par cette enquête ?*

Champ numérique

*Aide : Compter les séances pendant lesquelles les élèves sont amenés à utiliser la programmation informatique même si cela ne constitue pas l'objectif principal de la séance.*

**QD2\*** : *Sur quelles ressources vous appuyez-vous pour préparer vos activités de programmation informatique ?*

Liste à cocher choix multiples

- Programmes scolaires
- Ressources d'accompagnement proposées par le ministère
- Manuels scolaires
- Cahiers d'activités algorithme et programmation
- Activités proposées par des collègues dans votre établissement
- Activités proposées par d'autres enseignants sur Internet
- Site Internet France IOI
- Site Internet Lumni
- Site Internet VittaScience
- Site Internet Pixees
- Autre / Préciser

[SI QB1 ==MAT3]

**QD3\*** : *Les activités de programmation informatique que vous proposez à vos élèves sont-elles en lien avec les autres thèmes du programme de mathématiques ? Cocher les cases si vous proposez des activités dans les thèmes suivants :*

Liste à cocher choix multiples

- Nombres et calculs

- Organisation et gestion des données, fonctions
- Grandeurs et mesures
- Espace et géométrie
- Activités sans lien direct avec les autres thèmes du programme

[SI QB1 ==TEC3]

**QD4\*** : *Les activités de programmation informatique que vous proposez à vos élèves sont-elles en lien avec les autres thématiques du programme de technologie ? Cocher les cases si vous proposez des activités dans les thématiques suivantes :*

Liste à cocher choix multiples

- Design, innovation et créativité
- Les objets techniques, les services et les changements induits dans la société
- La modélisation et la simulation des objets et systèmes techniques
- Activités sans lien direct avec les autres thématiques du programme

[SI QB1 ==MAT2]

**QD5\*** : *Les activités de programmation informatique que vous proposez à vos élèves sont-elles en lien avec les autres grandes parties du programme de mathématiques ? Cocher les cases si vous proposez des activités dans les grandes parties suivantes :*

Liste à cocher choix multiples

- Nombres et calculs
- Géométrie
- Fonctions
- Statistiques et probabilités
- Activités sans lien direct avec les autres grandes parties du programme

[SI QB1 ==SNT2]

**QD6\*** : *Les activités de programmation informatique que vous proposez à vos élèves sont-elles en lien avec les thèmes du programme de SNT ? Cocher les cases si vous proposez des activités dans les thèmes suivants :*

Liste à cocher choix multiples

- Internet
- Le Web
- Les réseaux sociaux
- Les données structurées et leur traitement
- Localisation, cartographie et mobilité

- Informatique embarquée et objets connectés
- La photographie numérique
- Sans lien direct avec les thèmes du programme

**QD7\*** : *Quel type de démarche proposez-vous à vos élèves lors des activités de programmation ?*

Liste à cocher choix multiples

- Résolution d'exercices
- Résolution de problèmes
- Problème ouvert / tâche complexe
- Mini-projets
- Démarche d'investigation
- Exposées
- Autres / Préciser

**QD8\*** : *Quel type d'activité de programmation proposez-vous à vos élèves ?*

Liste à cocher choix multiples

- Tester un programme donné
- Expliquer un algorithme ou un programme donné
- Corriger un algorithme ou un programme dysfonctionnant
- Modifier un algorithme ou un programme donné
- Compléter un algorithme ou un programme donné
- Écrire un algorithme ou un programme en entier
- Traduire un algorithme en programme
- Autre / Préciser

**QD9\*** : *Quel type d'activités collaboratives proposez-vous à vos élèves dans le cadre de la programmation informatique ?*

Liste à cocher choix multiples

- Établissement d'objectifs communs
- Partage et répartition de tâches
- Assemblage de réalisations complémentaires
- Comparaison de solutions
- Développement collaboratif d'un même algorithme ou programme
- Je n'en propose pas vraiment

- Autre / Préciser

[SI QB1 ==MAT3 ou QB1 ==TEC3]

**QD10\*** : *Quel mode opératoire attendez-vous des élèves lorsqu'ils développent un programme informatique ?*

Liste à cocher choix multiples

- Édition du programme directement en langage de programmation graphique par essais-erreurs successifs
- Élaboration préalable d'un algorithme sous la forme d'un organigramme puis traduction en langage de programmation graphique
- Autre /Préciser

[SI QB1 ==MAT2 ou QB1 ==SNT2]

**QD11\*** : *Quel mode opératoire attendez-vous des élèves lorsqu'ils développent un programme informatique ?*

Liste à cocher choix multiples

- Écriture directement en Python par essais-erreurs successifs
- Écriture préalable d'un algorithme en « pseudo-code » puis traduction en Python
- Élaboration préalable d'un algorithme sous la forme d'un organigramme puis traduction en Python
- Autre / Préciser

**QD12** : *Proposez-vous à vos élèves des activités de manipulation ludiques n'ayant pas recours à l'ordinateur (informatique débranchée) ? Si oui, décrire brièvement ces activités.*

Champ texte

**QD13** : *(Facultatif) Vous pouvez ajouter ci-dessous des commentaires ou des remarques sur cette partie « Activité d'enseignement »*

Champ texte

## B.6 Groupe E : dispositif d'enseignement

**QE1\*** : *Dans quels lieux se déroulent les activités de programmation informatique ?*

Liste à cocher choix multiples

- Salle de classe ordinaire
- Salle informatique
- Laboratoire de technologie

- Laboratoire de sciences
- Autre / Préciser

**QE2\*** : *De combien d'élèves sont constituées vos classes pour le niveau concerné par cette enquête ?*

Champ numérique

*Aide : Si vous avez plusieurs classes à ce niveau, donner une moyenne des effectifs*

**QE3\*** : *Disposez-vous d'heures dédoublées pour le niveau concerné par cette enquête ? Ces heures pouvant être utilisées pour effectuer vos « travaux pratiques » en programmation informatique*

Boutons radio choix unique

Oui / Non

**QE4\*** : *Quel dispositif d'enseignement mettez-vous en œuvre concernant les activités de programmation ?*

Liste à cocher choix multiples

- Élèves en activité individuelle
- Élèves en activité en binômes
- Élèves en activité en groupes (effectif > 2)
- Élèves au tableau devant le groupe classe
- Enseignant au tableau devant le groupe classe (vidéo-projection, écriture)
- Enseignant passe de groupe en groupe
- Enseignant intervient dans les groupes à la demande des élèves
- Autre / Préciser

**QE5\*** : *(Facultatif) Vous pouvez ajouter ci-dessous des commentaires ou des remarques sur cette partie « Dispositif d'enseignement »*

Champ texte

## B.7 Groupe F : aspects matériels et logiciels

[SI QB1 ==MAT2 ou QB1 ==SNT2]

**QF1\*** : *Quel environnement de développement est utilisé par vos élèves ?*

Liste à cocher choix multiples

- Éditeur de texte générique (type bloc-notes)
- IDLE (éditeur par défaut de Python)

- EduPython
- Mu
- Thonny
- Pyzo
- TigerJython
- PyCharm
- Visual Studio code
- Vim
- Sublime Text
- Atom
- Spyder
- Repl.it (en ligne)
- Trinket (en ligne)
- Basthon (en ligne)
- VittaScience (en ligne)
- Jupyter notebook
- Autre / Préciser

[SI QB1 ==MAT2 ou QB1 ==SNT2]

**QF2\*** : Vos élèves utilisent-ils des bibliothèques Python? Si oui, lesquelles?

Liste à cocher choix multiples

- Aucune bibliothèque
- NumPy (calcul numérique)
- Scipy (calcul numérique)
- Matplotlib (visualisation de données)
- SymPy (calcul formel)
- Turtle (tortue graphique)
- PIL (traitement d'image)
- Pandas (traitement de données)
- Pygame (réalisation de jeux)
- Folium (cartes)
- Requests (Requêtes HTTP)
- Tkinter (interfaces graphiques)
- Nanpy (Arduino)



- Autre / préciser

**QF3\*** : *Quel type de machine vos élèves utilisent-ils pour développer leurs programmes informatiques ?*

Case à cocher choix multiples

- Ordinateur fixe
- Ordinateur portable
- Tablette
- Smartphone
- Calculatrice programmable
- Autre - Préciser

[SI QB1 ==TEC3 ou QB1 == SNT2]

Les questions suivantes portent sur les systèmes techniques programmables que vous faites programmer par vos élèves.

[SI QB1 ==TEC3 ou QB1 == SNT2]

**QF4\*** : *Mettez-vous à disposition des élèves des cartes programmables ? Si oui, lesquelles ?*

Case à cocher choix multiples

- Aucune carte programmable
- Micro:bit
- Arduino
- RaspberryPi
- Oxocard
- QuickPi
- Modules grooves
- Autre / Préciser

[SI QB1 ==TEC3 ou QB1 == SNT2]

**QF5\*** : *Mettez-vous à disposition des élèves des robots programmables ? Si oui, lesquels ?*

Case à cocher choix multiples

- Aucun robot programmable
- Thymio

- mBot
- Lego Mindstorms / Boost / WeDo
- Robot basé sur Micro:bit
- Robot basé sur Arduino
- Autre / Préciser

[SI QB1 ==TEC3]

**QF6\*** : *Mettez-vous à disposition des élèves des systèmes domotiques programmables (maquette ou réels)? Si oui, lesquels ?*

Case à cocher choix multiples

- Aucun système domotique
- Portail
- Barrière
- Volet roulant
- Serre
- Éclairage automatique
- Alarme
- Monte-charge
- Autre / Préciser

[SI QB1 ==TEC3 ou QB1 == SNT2]

**QF7\*** : *Utilisez-vous des simulations numériques programmables en remplacement de certains systèmes technologiques physiques ?*

Boutons radio choix unique

Oui/Non

[SI QB1 ==MAT3 ou QB1 ==TEC3]

**QF8\*** : *Quel(s) langage (s) de programmation graphique vos élèves utilisent-ils ?*

Case à cocher choix multiples

- Scratch
- mBlock
- Blockly
- Ardublock
- AppInventor
- Autre / Préciser

[SI QB1 ==MAT2 ou QB1==SNT2 ]

**QF9\*** : *Quel(s) langage(s) de programmation vos élèves utilisent-ils ?*

Case à cocher choix multiples

- Scratch
- mBlock
- Blockly
- Ardublock
- AppInventor
- Python
- Javascript
- Java
- C
- Php
- HTML / CSS
- Langage calculatrice CASIO
- Langage calculatrice TI
- Autre / Préciser

**QF10\*** : *Lesquels de ces navigateurs Internet sont à disposition de vos élèves dans votre établissement ?*

Case à cocher choix multiples

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Microsoft Internet Explorer
- Safari
- Opera
- Autre / préciser

**QF11** : *(Facultatif) Vous pouvez ajouter ci-dessous des commentaires ou des remarques sur cette partie « Aspects matériels et logiciels »*

Champ texte

## B.8 Groupe G : situation sanitaire

**QG1** : *(Facultatif) En quoi la situation sanitaire liée à la COVID19 a-t-elle modifiée vos pratiques d'enseignement de la programmation informatique ?*

Champ texte

## B.9 Remerciements

### **Remerciements**

Merci pour le temps que vous avez consacré à cette enquête et pour l'aide que cela constitue dans l'objectif d'acquérir une vision globale des pratiques d'enseignement de la programmation informatique dans l'enseignement secondaire.

Si vous souhaitez connaître les résultats de ce questionnaire, ou être informé de la conception de ressources numériques conçues pour l'enseignement de la programmation dans le cadre de ma recherche, vous pouvez m'adresser un mail à l'adresse suivante: [matthieu.branthome@univ-brest.fr](mailto:matthieu.branthome@univ-brest.fr)

# C Questionnaire élèves

## C.1 Connaissances préalables

Dans cette partie, il s'agit d'évaluer vos connaissances en programmation avant d'avoir joué au jeu Pyrates.

**Q1** : Comment estimez-vous vos connaissances en Scratch ?

Donner pour chaque élément une estimation allant de « Faibles » à « Importantes » en déplaçant le curseur.

- Scratch en général

Faibles  Importantes

- Notion de variable (Mettre maVariable à ...)

Faibles  Importantes

- Notion de conditionnelle (Si ... alors ... sinon ...)

Faibles  Importantes

- Notion de boucle for (Répéter ... fois)

Faibles  Importantes

- Notion de boucle while (Répéter jusqu'à ce que ...)

Faibles  Importantes

**Q2** : Avant de jouer à Pyrates, aviez-vous déjà utilisé le langage de programmation Python ou un langage similaire (sous la forme d'un texte) ?

- Jamais

- Un peu

- Beaucoup

- Très souvent

## C.2 Avis sur l'application

Dans cette partie, il s'agit de donner votre avis sur l'application Pyrates.

**Q3** : La description des niveaux (objectif, contraintes, fonctions de contrôle) vous semble-elle claire ?

Donner une estimation allant de « Pas claire » à « Très claire » en déplaçant le curseur.

Pas claire  Très claire

**Q4** : Les explications dans le « mémo programmation » (zone bleue) vous semblent-elles claires ?

Donner une estimation allant de « Pas claires » à « Très claires » en déplaçant le curseur.

Pas claire  Très claire

**Q5** : Dans le « mémo programmation » (zone bleue), les comparaisons avec Scratch vous semblent-elles utiles ?

Donner une estimation allant de « Pas utiles » à « Très utiles » en déplaçant le curseur.

Pas utiles  Très utiles

**Q6** : Les messages d'erreurs (en rouge dans la console) vous semblent-ils clairs ?

Donner une estimation allant de « Pas clairs » à « Très claires » en déplaçant le curseur.

Pas clairs  Très clairs

**Q7** : Les aides des professeurs vous semblent-elles suffisantes ?

Donner une estimation allant de « Insuffisantes » à « Tout à fait suffisantes » en déplaçant le curseur.

Insuffisantes  Tout à fait suffisantes

**Q8** : Évaluer chaque aspect de votre expérience :

- Prise en main de l'application

Difficile  Facile

- Difficulté du jeu

Difficile  Facile

- Aspect ludique

Ennuyeux  Divertissant

- Motivation

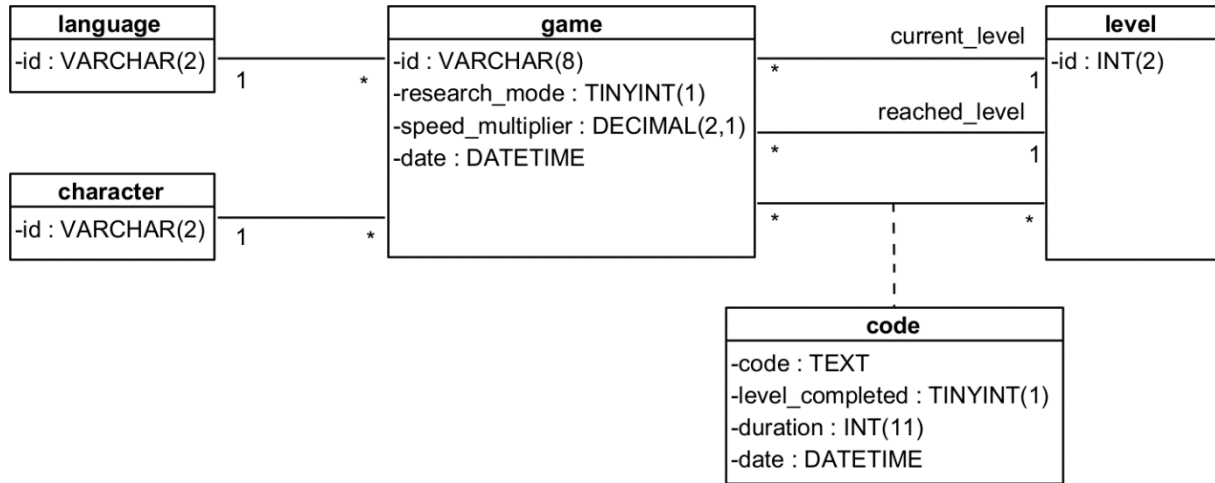
Décourageant  Motivant

### C.3 Remarques et suggestions

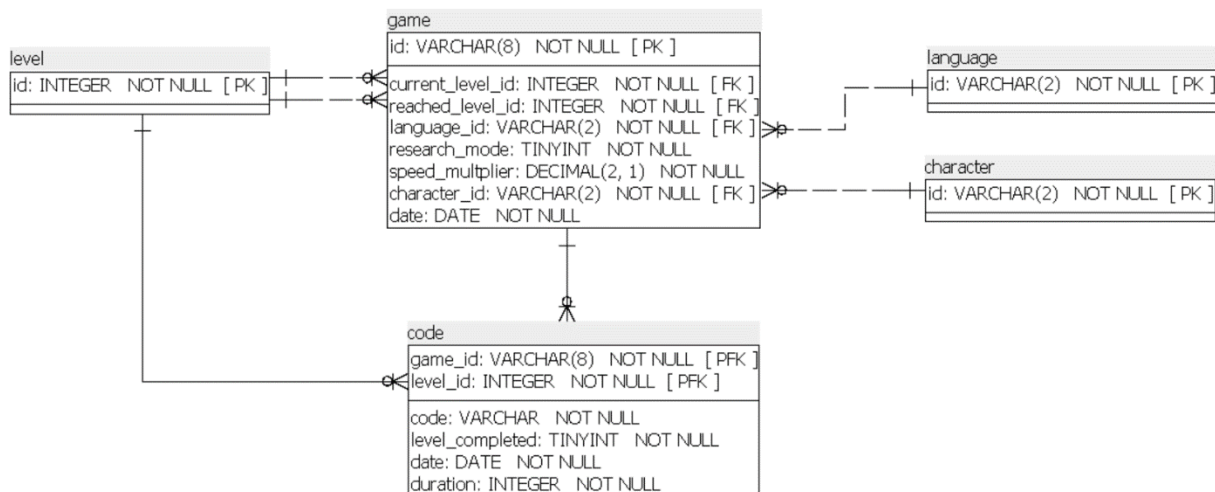
Vous pouvez ajouter ci-dessous vos remarques et suggestions concernant l'application Pyrates :

# D Base de données V1

## D.1 Modèle relationnel



## D.2 Modèle physique



## D.3 Détail des tables

La table « game » permet d'enregistrer l'état courant de la partie, elle comporte les champs suivants :

- id : identifiant unique de la partie généré par l'application web ;
- current\_level : le niveau courant lors de l'enregistrement ;
- reached\_level : plus haut niveau atteint lors de l'enregistrement ;
- language : langue de l'application choisie sur la page d'accueil ;
- reasearch\_mode : valeur booléenne désignant le mode de l'application (0 pour courant, 1 pour recherche) ;

- speed\_multiplier : valeur du curseur la vitesse d'exécution au moment de l'enregistrement ;
- character\_id : identifiant du personnage choisi ;
- date : date de l'enregistrement.

La table « code » contient le code saisi par l'utilisateur pour chaque niveau de la partie, ses champs sont les suivantes :

- game\_id : identifiant de la partie ;
- level\_id : identifiant du niveau ;
- code : code saisi pour le niveau level\_id de la partie game\_id ;
- level\_completed : détermine si le code saisi a permis de terminer le niveau ;
- date : date de l'enregistrement ;
- duration : la durée de jeu dans le niveau.

La table « level » permet de modéliser les différents niveau. Elle contient un unique champ :

- id : identifiant du niveau

La table « language » permet de modéliser les différentes langues disponibles pour le jeu. Elle contient un unique champ :

- id : identifiant de la langue

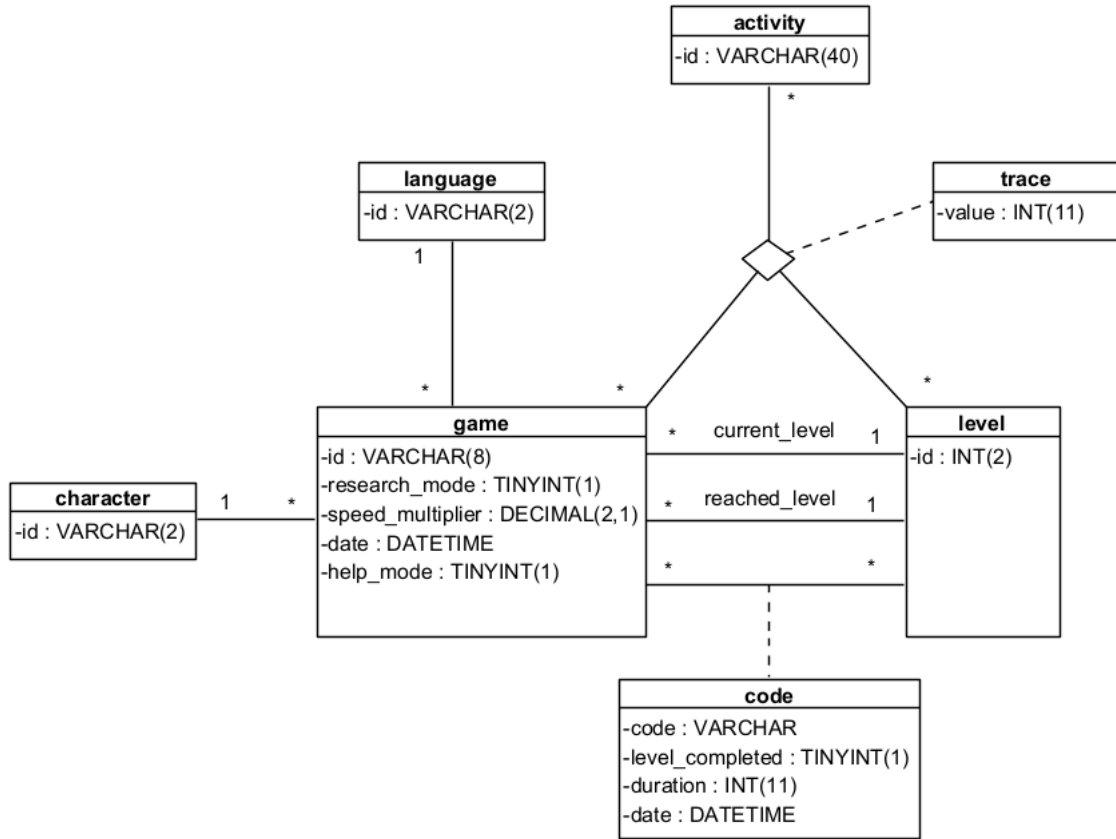
La table « character » permet de modéliser les différents personnages du jeu. Elle contient un unique champs :

- id : identifiant du personnage

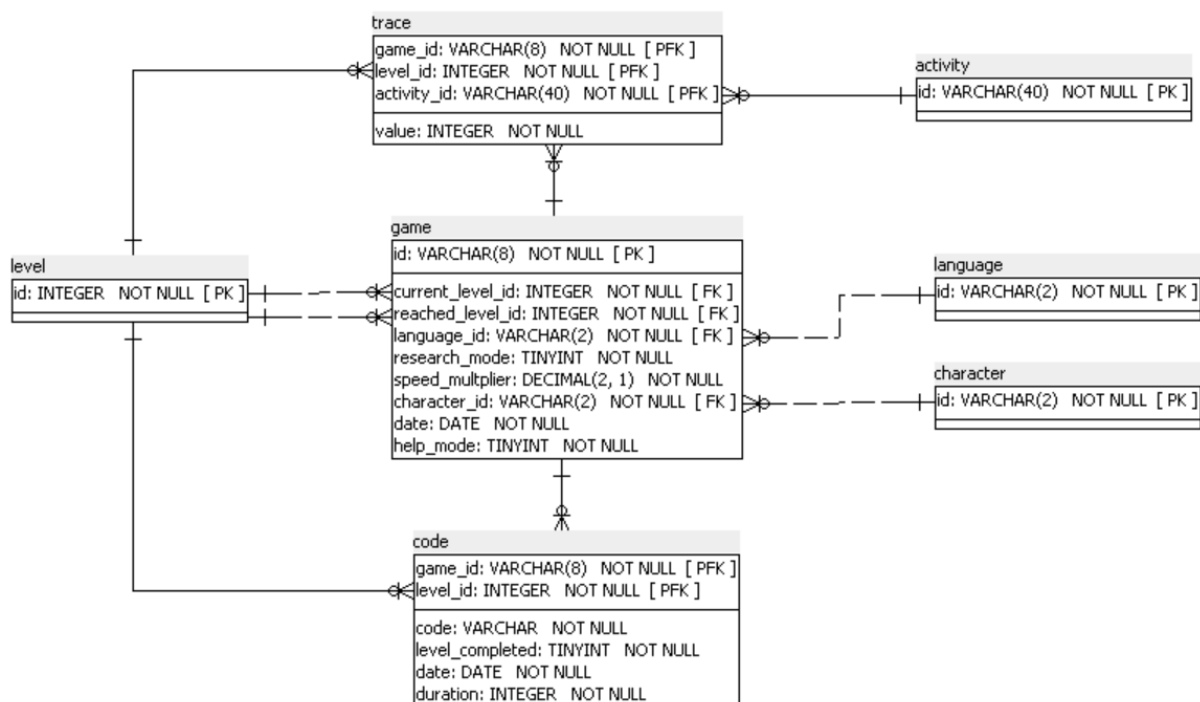


# EBase données V2

## E.1 Modèle relationnel



## E.2 Modèle physique



## E.3 Détail des tables

La table « game » permet d'enregistrer l'état courant de la partie, elle comporte les champs suivants :

- id : identifiant unique de la partie généré par l'application web ;
- current\_level : le niveau courant lors de l'enregistrement ;
- reached\_level : plus haut niveau atteint lors de l'enregistrement ;
- language : langue de l'application choisie sur la page d'accueil ;
- reasearch\_mode : valeur booléenne désignant le mode de l'application (0 pour courant, 1 pour recherche) ;
- speed\_multiplifier : valeur du curseur la vitesse d'exécution au moment de l'enregistrement ;
- character\_id : identifiant du personnage choisi
- date : date de l'enregistrement
- **{V2}** help\_mode : valeur booléenne renseignant sur l'activation du mode d'aide automatiques (0 pour non, 1 pour oui) ;

La table « code » contient le code saisi par l'utilisateur pour chaque niveau de la partie, ses champs sont les suivantes :

- game\_id : identifiant de la partie
- level\_id : identifiant du niveau
- code : code saisi pour le niveau level\_id de la partie game\_id
- level\_completed : détermine si le code saisi a permis de terminer le niveau
- date : date de l'enregistrement
- duration : la durée de jeu dans le niveau

La table « level » permet de modéliser les différents niveau. Elle contient un unique champ :

- id : identifiant du niveau

La table « language » permet de modéliser les différentes langues disponibles pour le jeu. Elle contient un unique champ :

- id : identifiant de la langue

La table « character » permet de modéliser les différents personnages du jeu. Rlle contient un unique champs :

- id : identifiant du personnage

**{V2}** La table « trace » contient les traces d'activités des utilisateurs ayant choisi l'aide automatique. Pour chaque niveau de la partie, ses champs sont les suivants :

- game\_id : identifiant de la partie
- level\_id : identifiant du niveau

- activity\_id : identifiant de l'activité
- value : valeur de la trace pour une partie, un niveau et une activité donnés

{V2} La table « activity » permet de modéliser les différents type d'activité du jeu. Elle contient un unique champ :

- id : identifiant de l'activité

# F Traces d'activités

## F.1 Consultation de contenus

Verbe : <https://py-rates.org/xAPI/verbs/consulted>

L'interface de l'application propose plusieurs zones de contenus. En particulier le guide de démarrage et le mémo de programmation qui sont consultables dans les panneaux latéraux qui apparaissent lors des clics sur les différents boutons. L'application trace la consultation par l'utilisateur de ces zones de contenus.

### Déclenchement

Le déclenchement de l'enregistrement est activé par l'affichage (total ou partiel) à l'écran d'une zone de contenu pour une durée supérieure à trois secondes. La disparition totale de la zone (par défilement, fermeture du panneau, changement d'onglet ou réduction de la fenêtre) entraîne l'arrêt du chronomètre pour cette zone.

### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) identifiant du contenu parmi : *startup-goal / startup-operation / startup-save / base-program / base-error / base-structure / base-trace / base-comment / var-creation / var-modification / var-usage / var-type / condi-1bran / condi-2bran / condi-3bran / for-simple / for-counter-o / for-counter-n / while-simple* ;
- (result.duration) durée de consultation du contenu en secondes ;
- (context.contextActivities.parent[o].id) identifiant du parent du contenu parmi : *startup / base / var / condi / for / while* ;
- (context.contextActivities.other[o].id) niveau du jeu lors de la consultation ;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date de début de la consultation.

### Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "K1TSLwV"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/consulted",
    "display": {
```

```

    "en-US": "consulted",
    "fr": "a consulté"
  }
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/contents/condi-1bran",
  "definition": {
    "name": {
      "en-US": "One branch conditional",
      "fr": "Conditionnelle à une branche"
    },
    "type": "https://py-rates.org/xAPI/activities/content"
  }
},
"result": {
  "duration": "PT9.593S"
},
"context": {
  "contextActivities": {
    "parent": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/contents/condi",
        "definition": {
          "name": {
            "en-US": "Conditional",
            "fr": "Conditionnelle"
          },
          "type": "https://py-rates.org/xAPI/activities/content"
        }
      }
    ]
  },
  "other": [
    {
      "objectType": "Activity",
      "id": "https://py-rates.org/xAPI/activities/levels/level1",
      "definition": {
        "name": {
          "en-US": "Level 1",
          "fr": "Niveau 1"
        },
        "type": "https://py-rates.org/xAPI/activities/level"
      }
    }
  ]
}

```

```

    }
  }
]
},
"extensions": {
  "https://py-rates.org/xAPI/extensions/game-time": "PT345S"
}
},
"timestamp": "2021-08-26T11:46:36.236+02:00"
}

```

## F.2 Copié-collé de contenus

Il est possible de copier du contenu (modèles des structures de code ou exemples d'utilisation) depuis le mémo de programmation, soit en cliquant sur les boutons « copier », soit de manière plus classique (Ctrl+c ou clic-droit > copier). Il est également possible de copier des éléments depuis l'éditeur de code, la zone de présentation des fonctions de contrôle (ou certaines aides {V2}). Ces contenus copiés peuvent ensuite être collés dans l'éditeur de code.

### F.2.1 Copie de contenu

Verbe : <https://py-rates.org/xAPI/verbs/copied>

#### Déclenchement

L'appui sur un des boutons « copier » du mémo de programmation ou la copie classique d'un élément du mémo, de l'éditeur de code, de la zone de présentation des fonctions de contrôle ou de certaines aides provoque l'enregistrement d'une trace d'activité.

#### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) identifiant du contenu parmi : *code-editor / control-functions / help-content {V2} / startup-goal / startup-operation / startup-save / base-program / base-error / base-structure / base-trace / base-comment / var-creation / var-modification / var-usage / var-type / condi-1bran / condi-2bran / condi-3bran / for-simple / for-counter-0 / for-counter-n / while-simple* ;
- (object.definition.extensions.code) code copié dans le presse-papier ;
- (context.contextActivities.parent[o].id) identifiant du parent du contenu parmi : *code-editor / control-functions / help-content / startup / base / var / condi / for / while* ;
- (context.contextActivities.other[o].id) niveau du jeu lors de la copie ;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date de la copie.

## Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "K1TSLwV"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/copied",
    "display": {
      "en-US": "copied",
      "fr": "a copié"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/contents/for-simple",
    "definition": {
      "name": {
        "en-US": "Simple repeat",
        "fr": "Répétition simple"
      },
      "type": "https://py-rates.org/xAPI/activities/content",
      "extensions": {
        "https://py-rates.org/xAPI/extensions/code": "for _ in range
(4):\n\tprint(\"répétition\")"
      }
    }
  },
  "context": {
    "contextActivities": {
      "parent": [
        {
          "objectType": "Activity",
          "id": "https://py-rates.org/xAPI/activities/contents/for",
          "definition": {
            "name": {
              "en-US": "For loop",
              "fr": "Boucle for"
            },
            "type": "https://py-rates.org/xAPI/activities/content"
          }
        }
      ]
    }
  }
}
```

```

    }
  }
],
"other": [
  {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/levels/level1",
    "definition": {
      "name": {
        "en-US": "Level 1",
        "fr": "Niveau 1"
      },
      "type": "https://py-rates.org/xAPI/activities/level"
    }
  }
]
},
"extensions": {
  "https://py-rates.org/xAPI/extensions/game-time": "PT230S"
}
},
"timestamp": "2021-08-26T12:19:49.884+02:00"
}

```

## F.2.2 Collage de contenu

Verbe : <https://py-rates.org/xAPI/verbs/pasted>

### Déclenchement

Le collage (Ctrl+v ou clic-droit > coller) du contenu du presse-papier dans l'éditeur de code provoque l'enregistrement d'une trace d'activité.

### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.definition.extensions.code) code collé depuis le presse-papier dans l'éditeur de code ;
- (context.contextActivities.other[0].id) niveau du jeu lors du collage ;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date du collage.

### Exemple

```

{
  "actor": {
    "objectType": "Agent",
    "account": {

```



```

    "homePage": "https://py-rates.org",
    "name": "DZNBs76"
  }
},
"verb": {
  "id": "https://py-rates.org/xAPI/verbs/pasted",
  "display": {
    "en-US": "pasted",
    "fr": "a collé"
  }
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/properties/copied-text",
  "definition": {
    "name": {
      "en-US": "copied text",
      "fr": "Texte copié"
    },
    "type": "https://py-rates.org/xAPI/activities/content",
    "extensions": {
      "https://py-rates.org/xAPI/extensions/code": "avancer()"
    }
  }
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level1",
        "definition": {
          "name": {
            "en-US": "Level 1",
            "fr": "Niveau 1"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT536S"
  }
}

```

```

    }
  },
  "timestamp": "2021-08-26T12:42:51.908+02:00"
}

```

## F.3 Lancement de programmes

Verbe : <https://py-rates.org/xAPI/verbs/launched>

Un programme tapé dans l'éditeur de code peut être lancé en utilisant le bouton « Exécuter » de l'application. Ce programme a ensuite plusieurs issues possibles :

- il peut ne pas être exécuté pour quatre raisons : s'il comporte trop de lignes dans un niveau avec contrainte de longueur (*too-many-lines*), s'il comprend une erreur syntaxique (*syntactic-error*), s'il contient une erreur sémantique (*semantic-error*) ou s'il contient une erreur liée au jeu (*game-error*) ;
- dans le cas où il est exécuté, il peut être arrêté par l'utilisateur à l'aide du bouton « arrêter » (*user\_stopped*), il peut entraîner la perte du niveau (*level-lost*), il peut permettre de terminer le niveau (*level-completed*), ou il peut être exécuté complètement sans faire perdre ni terminer le niveau (*fully-executed*).

### Déclenchement

L'appuie sur le bouton « Exécuter » lance l'enregistrement de la trace d'activité. Le chronomètre s'arrête à différents moments selon l'issue du programme (voir les différents types).

### Données enregistrées

- (*actor.account.name*) identifiant de la partie ;
- (*object.id*) type de programme parmi : *too-many-lines* (trop de lignes) / *syntactic-error* (erreur de syntaxe) / *semantic-error* (erreur sémantique) / *game-error* (erreur liée au jeu) / *user-stopped* (exécution stoppée par l'utilisateur) / *fully-executed* (exécution complète sans perdre ni terminer le niveau) / *level-lost* (niveau perdu) / *level-completed* (niveau terminé) ;
- (*object.definition.extensions.code*) code du programme lancé ;
- (*result.duration*) durée de l'exécution ;
- (*result.completion*) complétion de l'exécution (le programme arrive à son terme) ;
- (*result.extensions.execution-speed-multiplier*) vitesse d'exécution ;
- (*context.contextActivities.other[o].id*) niveau du jeu lors du lancement ;
- (*context.extensions.game-time*) temps écoulé depuis le début du niveau ;
- (*timestamp*) date du lancement ;

Optionnel (selon le type) :

- [*too-many-lines*] (*result.extensions.extra-lines-number*) nombre de lignes en trop ;
- [*syntactic-error*, *semantic-error* et *game-error*] (*result.extensions.error*) message d'erreur ;

- [game-error] (result.extensions.game-error-reason) raison de l'erreur parmi : *open-chest-location* (tentative d'ouverture du coffre à un mauvais emplacement) / *open-chest-key* (tentative d'ouverture du coffre sans la clé) / *read-message-location* (lecture d'un message à un mauvais emplacement) / *walk-location* (impossibilité d'avancer vers un emplacement) / *not-allowed-function* (utilisation d'une fonction non autorisée) / *function\_parameters* (mauvais paramètre pour une fonction)
- [user-stopped] (result.extensions.stopped-line) numéro de la ligne stoppée ;
- [level-lost] (result.extensions.level-lost-reason) raison de la perte parmi : *spike-touch* (pique touchée) / *barrel-explosion* (explosion du tonneau) / *pirate-shot* (autre pirate tué) ;
- {V2} [user-stopped et fully-executed] (result.extensions.progression) progression dans le niveau : avancement dans le niveau rendu possible par l'exécution du programme (en pourcentage)

## emples

### [too-many-lines]

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "s5X4xhe"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {
      "en-US": "launched execution",
      "fr": "a lancé l'exécution"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/programs/too-many-lines",
    "definition": {
      "name": {
        "en-US": "Too many lines program",
        "fr": "Programme ayant trop de lignes"
      }
    },
    "type": "https://py-rates.org/xAPI/activities/program",
    "extensions": {
```

```

    "https://py-rates.org/xAPI/extensions/code":
"avancer()\ndroite()\navancer()\ngauche()\navancer()\ndroite()\navancer()\navan
cer()\navancer()\navancer()\navancer()\navancer()\navancer()"
    }
  }
},
"result": {
  "duration": "PT0.07S",
  "completion": false,
  "extensions": {
    "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 1.7,
    "https://py-rates.org/xAPI/extensions/extra-lines-number": 3
  }
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level1",
        "definition": {
          "name": {
            "en-US": "Level 1",
            "fr": "Niveau 1"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT38S"
  }
},
"timestamp": "2021-08-26T22:38:19.991+02:00"
}

```

### [syntactic-error]

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "LA5y1KP"
    }
  }
}

```

```

    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {
      "en-US": "launched execution",
      "fr": "a lancé l'exécution"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/programs/syntactic-error",
    "definition": {
      "name": {
        "en-US": "Syntactic error program",
        "fr": "Programme contenant une erreur syntaxique"
      },
      "type": "https://py-rates.org/xAPI/activities/program",
      "extensions": {
        "https://py-rates.org/xAPI/extensions/code": "#Ecrire le code de son programme ci-dessous\nnavancer(\n"
      }
    }
  },
  "result": {
    "duration": "PT0.231S",
    "completion": false,
    "extensions": {
      "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 2.4,
      "https://py-rates.org/xAPI/extensions/error": "Ligne 2 : Il manque une parenthèse fermante: )".
    }
  },
  "context": {
    "contextActivities": {
      "other": [
        {
          "objectType": "Activity",
          "id": "https://py-rates.org/xAPI/activities/levels/level1",
          "definition": {
            "name": {
              "en-US": "Level 1",
              "fr": "Niveau 1"
            }
          }
        }
      ]
    }
  }
}

```

```

    },
    "type": "https://py-rates.org/xAPI/activities/level"
  }
}
],
},
"extensions": {
  "https://py-rates.org/xAPI/extensions/game-time": "PT378S"
}
},
"timestamp": "2021-08-26T22:56:43.801+02:00"
}

```

### [semantic-error]

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "LA5y1KP"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {
      "en-US": "launched execution",
      "fr": "a lancé l'exécution"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/programs/semantic-error",
    "definition": {
      "name": {
        "en-US": "Semantic error program",
        "fr": "Programme contenant une erreur sémantique"
      },
      "type": "https://py-rates.org/xAPI/activities/program",
      "extensions": {
        "https://py-rates.org/xAPI/extensions/code": "#Ecrire le code de son programme ci-dessous\na = 3+\nhello!\n\n"
      }
    }
  },
}

```

```

"result": {
  "duration": "PT0.08S",
  "completion": false,
  "extensions": {
    "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 1,
    "https://py-rates.org/xAPI/extensions/error": "TypeError: unsupported
operand type(s) for +: 'int' and 'str' on line 2"
  }
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level1",
        "definition": {
          "name": {
            "en-US": "Level 1",
            "fr": "Niveau 1"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT832S"
  }
},
"timestamp": "2021-08-26T23:06:40.186+02:00"
}

```

### [game-error]

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "iTvcNqE"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {

```

```

    "en-US": "launched execution",
    "fr": "a lancé l'exécution"
  }
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/programs/game-error",
  "definition": {
    "name": {
      "en-US": "Game error program",
      "fr": "Programme contenant une erreur liée au jeu"
    },
    "type": "https://py-rates.org/xAPI/activities/program",
    "extensions": {
      "https://py-rates.org/xAPI/extensions/code": "#Ecrire le code de son programme ci-dessous\nouvrir()"
    }
  },
},
"result": {
  "duration": "PT0.161S",
  "completion": false,
  "extensions": {
    "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 1,
    "https://py-rates.org/xAPI/extensions/error": "Ligne 2 : Impossible d'ouvrir le coffre à cet emplacement",
    "https://py-rates.org/xAPI/extensions/game-error-reason": "open-chest-location"
  },
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level1",
        "definition": {
          "name": {
            "en-US": "Level 1",
            "fr": "Niveau 1"
          },
        },
        "type": "https://py-rates.org/xAPI/activities/level"
      }
    ]
  }
}

```



```

]
},
"extensions": {
  "https://py-rates.org/xAPI/extensions/game-time": "PT30S"
}
},
"timestamp": "2021-08-31T22:58:38.681+02:00"
}
[user-stopped]
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "LA5y1KP"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {
      "en-US": "launched execution",
      "fr": "a lancé l'exécution"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/programs/user-stopped",
    "definition": {
      "name": {
        "en-US": "User stopped program",
        "fr": "Programme arrêté par l'utilisateur"
      },
      "type": "https://py-rates.org/xAPI/activities/program",
      "extensions": {
        "https://py-rates.org/xAPI/extensions/code": "obstacle =
decteterObstacle()\nif obstacle == True:\n    sauter()\nelse:\n
avancer()\navancer()\nobstacle = decteterObstacle()\nif obstacle == True:\n
sauter()\nelse:\n    avancer()\navancer()\nobstacle = decteterObstacle()\n",
      }
    }
  },
  "result": {
    "duration": "PT2.388S",

```

```

"completion": false,
"extensions": {
  "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 3,
  "https://py-rates.org/xAPI/extensions/stopped-line": 7,
  "https://py-rates.org/xAPI/extensions/progression": 65
}
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level3",
        "definition": {
          "name": {
            "en-US": "Level 3",
            "fr": "Niveau 3"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT331S"
  }
},
"timestamp": "2021-08-26T23:14:39.632+02:00"
}

```

### [fully-executed]

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "LA5y1KP"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {
      "en-US": "launched execution",

```

```

    "fr": "a lancé l'exécution"
  }
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/programs/fully-executed",
  "definition": {
    "name": {
      "en-US": "Fully executed program",
      "fr": "Programme complètement exécuté"
    },
    "type": "https://py-rates.org/xAPI/activities/program",
    "extensions": {
      "https://py-rates.org/xAPI/extensions/code": "for _ in range(4):\n obstacle
= detecterObstacle()\n if obstacle == True:\n sauter()\n if obstacle == False:\n
avancer()\n avancer()\nouvrir()\n",
    }
  }
},
"result": {
  "duration": "PT7.969S",
  "completion": true,
  "extensions": {
    "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 2.5,
    "https://py-rates.org/xAPI/extensions/progression": 34
  }
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level3",
        "definition": {
          "name": {
            "en-US": "Level 3",
            "fr": "Niveau 3"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  }
},

```

```

    "extensions": {
      "https://py-rates.org/xAPI/extensions/game-time": "PT30S"
    }
  },
  "timestamp": "2021-08-26T23:10:52.487+02:00"
}
[level-lost]
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "xJSiRnG"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {
      "en-US": "launched execution",
      "fr": "a lancé l'exécution"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/programs/level-lost",
    "definition": {
      "name": {
        "en-US": "Level loss program",
        "fr": "Programme provoquant la perte du niveau"
      },
      "type": "https://py-rates.org/xAPI/activities/program",
      "extensions": {
        "https://py-rates.org/xAPI/extensions/code": "#Ecrire le code de son programme ci-dessous\nsauter()",
      }
    }
  },
  "result": {
    "duration": "PT1.461S",
    "completion": true,
    "extensions": {
      "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 3,
      "https://py-rates.org/xAPI/extensions/level-lost-reason": "spikes-touch"
    }
  }
}

```

```

    }
  },
  "context": {
    "contextActivities": {
      "other": [
        {
          "objectType": "Activity",
          "id": "https://py-rates.org/xAPI/activities/levels/level3",
          "definition": {
            "name": {
              "en-US": "Level 3",
              "fr": "Niveau 3"
            },
            "type": "https://py-rates.org/xAPI/activities/level"
          }
        }
      ]
    },
    "extensions": {
      "https://py-rates.org/xAPI/extensions/game-time": "PT56S"
    }
  },
  "timestamp": "2021-08-26T21:31:05.013+02:00"
}

```

### [level-completed]

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "eQNEhYp"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/launched",
    "display": {
      "en-US": "launched execution",
      "fr": "a lancé l'exécution"
    }
  },
  "object": {
    "objectType": "Activity",

```

```

"id": "https://py-rates.org/xAPI/activities/programs/level-completed",
"definition": {
  "name": {
    "en-US": "Level complete program",
    "fr": "Programme provoquant la fin du niveau"
  },
  "type": "https://py-rates.org/xAPI/activities/program",
  "extensions": {
    "https://py-rates.org/xAPI/extensions/code": "for _ in range(6):\n obstacle
= detecterObstacle()\n if obstacle == True:\n sauter()\n else:\n avancer()\n
avancer()\nouvrir()\n",
  }
},
"result": {
  "duration": "PT12.838S",
  "completion": true,
  "extensions": {
    "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 3
  }
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level3",
        "definition": {
          "name": {
            "en-US": "Level 3",
            "fr": "Niveau 3"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT567S"
  }
},
"timestamp": "2021-08-26T21:55:50.054+02:00"
}

```

## F.4 Navigation entre les niveaux

### F.4.1 Niveau commencé

Verbe : <https://py-rates.org/xAPI/verbs/started>

#### Déclenchement

L'enregistrement de la trace d'activité a lieu lorsqu'un nouveau niveau commence.

#### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) niveau du jeu commencé ;
- (timestamp) date de début du niveau.

#### Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "eQNEhYp"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/started",
    "display": {
      "en-US": "started",
      "fr": "a commencé"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/levels/level3",
    "definition": {
      "name": {
        "en-US": "Level 3",
        "fr": "Niveau 3"
      },
      "type": "https://py-rates.org/xAPI/activities/level"
    }
  },
  "context": {
    "contextActivities": {
```

```

"other": [
  {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/levels/level3",
    "definition": {
      "name": {
        "en-US": "Level 3",
        "fr": "Niveau 3"
      },
      "type": "https://py-rates.org/xAPI/activities/level"
    }
  }
]
},
"timestamp": "2021-08-26T21:51:37.306+02:00"
}

```

#### F.4.2 Niveau terminé

Verbe : <https://py-rates.org/xAPI/verbs/completed>

#### Déclenchement

L'enregistrement de la trace d'activité a lieu lorsqu'un niveau est terminé.

#### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) niveau du jeu terminé ;
- (result.duration) temps écoulé depuis le début du niveau ;
- (context.extensions.game-time) durée du niveau
- (timestamp) date de fin du niveau.

#### Exemple

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "eQNEhYp"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/completed",
    "display": {
      "en-US": "completed",

```



```

    "fr": "a terminé"
  }
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/levels/level2",
  "definition": {
    "name": {
      "en-US": "Level 2",
      "fr": "Niveau 2"
    },
    "type": "https://py-rates.org/xAPI/activities/level"
  }
},
"result": {
  "duration": "PT18S"
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level2",
        "definition": {
          "name": {
            "en-US": "Level 2",
            "fr": "Niveau 2"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT18S"
  }
},
"timestamp": "2021-08-26T21:51:31.725+02:00"
}

```

### F.4.3 Niveau recommencé

Verbe : <https://py-rates.org/xAPI/verbs/restarted>

#### Déclenchement

L'enregistrement de la trace d'activité a lieu lorsqu'un niveau est recommencé soit en cliquant sur le bouton « Rejouer niveau » à la fin d'un niveau, soit en cliquant sur le numéro d'un niveau terminé précédemment dans l'espace dédié aux niveaux.

### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) niveau du jeu recommencé ;
- (timestamp) date de recommencement du niveau.

### Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "eQNEhYp"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/restarted",
    "display": {
      "en-US": "restarted",
      "fr": "a recommencé"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/levels/level1",
    "definition": {
      "name": {
        "en-US": "Level 1",
        "fr": "Niveau 1"
      },
      "type": "https://py-rates.org/xAPI/activities/level"
    }
  },
  "context": {
    "contextActivities": {
      "other": [
        {
          "objectType": "Activity",
          "id": "https://py-rates.org/xAPI/activities/levels/level1",
          "definition": {
```

```

        "name": {
            "en-US": "Level 1",
            "fr": "Niveau 1"
        },
        "type": "https://py-rates.org/xAPI/activities/level"
    }
}
]
}
},
"timestamp": "2021-08-26T21:47:35.938+02:00"
}

```

#### F.4.4 Niveau repris

Verbe : <https://py-rates.org/xAPI/verbs/resumed>

#### Déclenchement

L'enregistrement de la trace d'activité a lieu lorsqu'un niveau est repris alors qu'il avait été quitté sans qu'il ne soit terminé.

#### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) niveau du jeu repris ;
- (timestamp) date de reprise du niveau.

#### Exemple

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "eQNEhYp"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/resumed",
    "display": {
      "en-US": "resumed",
      "fr": "a repris"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/levels/level1",

```

```

"definition": {
  "name": {
    "en-US": "Level 1",
    "fr": "Niveau 1"
  },
  "type": "https://py-rates.org/xAPI/activities/level"
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level1",
        "definition": {
          "name": {
            "en-US": "Level 1",
            "fr": "Niveau 1"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  }
},
"timestamp": "2021-08-26T21:47:35.938+02:00"
}

```

#### F.4.5 Niveau quitté {V2}

Verbe :

##### Déclenchement

L'enregistrement de la trace d'activité a lieu lorsqu'un niveau est quitté (en cliquant sur le lien d'un autre niveau).

##### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) niveau du jeu repris ;
- (timestamp) date du départ du niveau.

##### Exemple

```

{
  "actor": {
    "objectType": "Agent",

```

```

"account": {
  "homePage": "https://py-rates.org",
  "name": "ef5EhT3"
},
"verb": {
  "id": "https://py-rates.org/xAPI/verbs/leaved",
  "display": {
    "en-US": "leaved",
    "fr": "a quitté"
  }
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/levels/level3",
  "definition": {
    "name": {
      "en-US": "Level 3",
      "fr": "Niveau 3"
    },
    "type": "https://py-rates.org/xAPI/activities/level"
  }
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level3",
        "definition": {
          "name": {
            "en-US": "Level 3",
            "fr": "Niveau 3"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  }
},
"timestamp": "2022-09-04T22:03:43.936+02:00"
}

```

## F.5 Aide

L'enseignant encadre l'utilisation de l'application par ses élèves. Il peut ainsi être amené à aider les utilisateurs en répondant à leurs questions ou en leur prêtant main forte. Dans le mode recherche, une partie de l'interface nommé « Réservé à l'enseignant » est dédiée au professeur afin qu'il puisse renseigner le type d'aide qu'il a fourni à l'utilisateur en cliquant sur différents boutons disponibles.

Verbe : <https://py-rates.org/xAPI/verbs/was-helped>

### Déclenchement

L'appuie sur un bouton de la zone « Réservé à l'enseignant » amène l'enregistrement d'une trace d'activité.

### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) type d'aide parmi : *application* (liée à l'utilisation de l'application) / *game* (liée au jeu lui-même) / *syntax* (liée à un problème de syntaxe) / *semantics* (liée à un problème de sémantique) / *notion* (liée au choix d'une notion) / *other* (autre type d'aide).
- (context.contextActivities.other[o].id) niveau du jeu au moment de l'aide;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date de l'aide.

### Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "LA5y1KP"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/was-helped",
    "display": {
      "en-US": "was helped",
      "fr": "a été aidé"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/helps/application",
    "definition": {
      "name": {
```

```

    "en-US": "Help related to the use of the application",
    "fr": "Aide liée à l'utilisation de l'application"
  },
  "type": "https://py-rates.org/xAPI/activities/help"
}
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level3",
        "definition": {
          "name": {
            "en-US": "Level 3",
            "fr": "Niveau 3"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT365S"
  }
},
"timestamp": "2021-08-26T23:27:08.604+02:00"
}

```

## F.6 Aide {V2}

L'enseignant encadre l'utilisation de l'application par ses élèves. Il peut ainsi être amené à déclencher une aide sur une demande de la part d'une élève (bouton « Aide » cliqué par l'élève) ou en décidant d'aller vers eux (bouton « Réservé enseignant » cliqué par lui-même). Une partie de l'interface est dédiée au professeur afin qu'il puisse choisir le type d'aide qu'il souhaite déclencher en cliquant sur différents boutons. Les aides disponibles sont : objectif du niveau (obj), fonction de contrôle (fon), notion en jeu (not), implémentation de la notion (imp), solution complète (sol) et autre aide (aut).

Le cycle de vie d'une aide est le suivant :

- l'élève demande une aide en cliquant sur le bouton « Aide » ou l'enseignant propose une aide en cliquant sur le bouton « Réservé enseignant » : aide demandé ;

- l’enseignant choisit parmi les aides possibles dans l’interface qu’il a disposition : aide reçue ;

Ensuite le fonctionnement dépend du type d’aide :

- [obj et fon] l’élève lance l’aide avec le bouton « Commencer » : aide lancée ;
- [obj et fon] l’élève peut relancer l’aide avec le bouton « Recommencer » : aide relancée ;
- [obj,fon,not,imp et sol] l’élève quitte l’aide avec le bouton « OK » : aide quittée ;
- [aut] l’aide se quitte toute seule après avoir été reçue.

### F.6.1 Aide demandée

Verbe : <https://py-rates.org/xAPI/verbs/asked>

#### Déclenchement

L’appuie sur le bouton « Aide » ou « Réserve enseignant » amène l’enregistrement d’une trace d’activité.

#### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (context.contextActivities.other[o].id) niveau du jeu au moment de la demande d’aide;
- (context.extensions.help-origin) origine de la demande d’aide : *student* ou *teacher*
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date de la demande d’aide.

#### Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "wLn9yF5"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/asked",
    "display": {
      "en-US": "asked",
      "fr": "a demandé"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/help",

```



```

"definition": {
  "name": {
    "en-US": "Help",
    "fr": "Aide"
  },
  "type": "https://py-rates.org/xAPI/activities/help"
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level1",
        "definition": {
          "name": {
            "en-US": "Level 1",
            "fr": "Niveau 1"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT6S",
    "https://py-rates.org/xAPI/extensions/help-origin": "student"
  }
},
"timestamp": "2022-09-05T18:07:41.832+02:00"
}

```

## F.6.2 Aide reçue

Verbe : <https://py-rates.org/xAPI/verbs/received>

### Déclenchement

L'appuie sur un des boutons d'aide de la zone « Réservé à l'enseignant » amène l'enregistrement d'une trace d'activité.

### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) type d'aide parmi : *game* (liée au jeu lui-même) / *control* (liée au choix des fonctions de contrôle) / *notion* (liée au choix d'une notion) / *implementation*

(liée à l'implémentation d'une notion) / *solution* (liée à la solution complète du niveau) / *other* (autre type d'aide).

- (context.contextActivities.other[o].id) niveau du jeu au moment du déclenchement de l'aide;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date du l'aide.

### Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "LA5y1KP"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/received",
    "display": {
      "en-US": "received",
      "fr": "a reçu"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/helps/application",
    "definition": {
      "name": {
        "en-US": "Help related to the use of the application",
        "fr": "Aide liée à l'utilisation de l'application"
      },
      "type": "https://py-rates.org/xAPI/activities/help"
    }
  },
  "context": {
    "contextActivities": {
      "other": [
        {
          "objectType": "Activity",
          "id": "https://py-rates.org/xAPI/activities/levels/level3",
          "definition": {
            "name": {
              "en-US": "Level 3",

```

```

        "fr": "Niveau 3"
      },
      "type": "https://py-rates.org/xAPI/activities/level"
    }
  ]
},
"extensions": {
  "https://py-rates.org/xAPI/extensions/game-time": "PT365S"
}
},
"timestamp": "2021-08-26T23:27:08.604+02:00"
}

```

### F.6.3 Aide lancée

Verbe : <https://py-rates.org/xAPI/verbs/started>

#### Déclenchement

L'appuie sur le bouton « Commencer » d'une aide de type « obj » ou « fon » amène l'enregistrement d'une trace d'activité.

#### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) type d'aide parmi : *game* (liée au jeu lui-même) / *control* (liée au choix des fonctions de contrôle) / *notion* (liée au choix d'une notion) / *implementation* (liée à l'implémentation d'une notion) / *solution* (liée à la solution complète du niveau) / *other* (autre type d'aide).
- (context.contextActivities.other[0].id) niveau du jeu au moment du lancement de l'aide;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date du lancement de l'aide.

#### Exemple

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "N7MrPvj"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/started",
    "display": {

```

```

    "en-US": "started",
    "fr": "a lancé"
  }
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/helps/game",
  "definition": {
    "name": {
      "en-US": "Help related to the game itself",
      "fr": "Aide liée au jeu lui même"
    },
    "type": "https://py-rates.org/xAPI/activities/help"
  }
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level7",
        "definition": {
          "name": {
            "en-US": "Level 7",
            "fr": "Niveau 7"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT6S"
  }
},
"timestamp": "2022-09-05T18:38:28.193+02:00"
}

```

#### F.6.4 Aide relancée

Verbe : <https://py-rates.org/xAPI/verbs/restarted>

#### Déclenchement

L'appuie sur le bouton « Recommencer » d'une aide de type « obj » ou « fon » entraîne l'enregistrement d'une trace d'activité.

## Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.id) type d'aide parmi : *game* (liée au jeu lui-même) / *control* (liée au choix des fonctions de contrôle) / *notion* (liée au choix d'une notion) / *implementation* (liée à l'implémentation d'une notion) / *solution* (liée à la solution complète du niveau) / *other* (autre type d'aide).
- (context.contextActivities.other[o].id) niveau du jeu au moment de la relance de l'aide;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date de la relance de l'aide.

## Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "N7MrPvj"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/restarted",
    "display": {
      "en-US": "restarted",
      "fr": "a relancé"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/helps/game",
    "definition": {
      "name": {
        "en-US": "Help related to the game itself",
        "fr": "Aide liée au jeu lui-même"
      },
      "type": "https://py-rates.org/xAPI/activities/help"
    }
  },
  "context": {
    "contextActivities": {
      "other": [
        {
```

```

    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/levels/level7",
    "definition": {
      "name": {
        "en-US": "Level 7",
        "fr": "Niveau 7"
      },
      "type": "https://py-rates.org/xAPI/activities/level"
    }
  ]
},
"extensions": {
  "https://py-rates.org/xAPI/extensions/game-time": "PT57S"
}
},
"timestamp": "2022-09-05T18:39:18.501+02:00"
}

```

## F.6.5 Aide quittée

### Déclenchement

L'appuie sur le bouton « OK » d'une aide de type « obj », « fon », « not », « imp » ou « sol », ainsi que, de façon automatique, le déclenchement d'une aide de type « aut » amène l'enregistrement d'une trace d'activité.

### Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (context.contextActivities.other[0].id) niveau du jeu au moment du départ de l'aide ;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date de départ depuis l'aide.

### Exemple

```

{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "N7MrPvj"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/leaved",
    "display": {

```

```

    "en-US": "leaved",
    "fr": "a quitté"
  },
},
"object": {
  "objectType": "Activity",
  "id": "https://py-rates.org/xAPI/activities/help",
  "definition": {
    "name": {
      "en-US": "Help",
      "fr": "Aide"
    },
  },
  "type": "https://py-rates.org/xAPI/activities/help"
},
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level7",
        "definition": {
          "name": {
            "en-US": "Level 7",
            "fr": "Niveau 7"
          },
        },
        "type": "https://py-rates.org/xAPI/activities/level"
      }
    ]
  },
},
"extensions": {
  "https://py-rates.org/xAPI/extensions/game-time": "PT88S"
},
},
"timestamp": "2022-09-05T18:39:49.593+02:00"
}

```

## F.7 Modification de la vitesse d'exécution

Il est possible de modifier la vitesse d'exécution des programmes en jouant sur un curseur allant de « tortue » à « lièvre ». Cela a pour effet d'appliquer un coefficient multiplicateur décimal allant de 1 à 3 sur la vitesse d'exécution.

Verbe : <https://py-rates.org/xAPI/verbs/changed>

## Déclenchement

Le déplacement puis le relâchement du curseur entraîne l'enregistrement de la trace d'activité.

## Données enregistrées

- (actor.account.name) identifiant de la partie ;
- (object.definition.extensions.execution-speed-multiplier) nouvelle valeur du coefficient multiplicateur ;
- (context.contextActivities.other[o].id) niveau du jeu lors de la modification ;
- (context.extensions.game-time) temps écoulé depuis le début du niveau ;
- (timestamp) date de modification de la vitesse d'exécution.

## Exemple

```
{
  "actor": {
    "objectType": "Agent",
    "account": {
      "homePage": "https://py-rates.org",
      "name": "LA5y1KP"
    }
  },
  "verb": {
    "id": "https://py-rates.org/xAPI/verbs/changed",
    "display": {
      "en-US": "changed",
      "fr": "a changé"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "https://py-rates.org/xAPI/activities/properties/execution-speed",
    "definition": {
      "name": {
        "en-US": "Execution speed",
        "fr": "Vitesse d'exécution"
      }
    },
    "type": "https://py-rates.org/xAPI/activities/property",
    "extensions": {
      "https://py-rates.org/xAPI/extensions/execution-speed-multiplier": 1.9
    }
  }
}
```



```
},
"context": {
  "contextActivities": {
    "other": [
      {
        "objectType": "Activity",
        "id": "https://py-rates.org/xAPI/activities/levels/level3",
        "definition": {
          "name": {
            "en-US": "Level 3",
            "fr": "Niveau 3"
          },
          "type": "https://py-rates.org/xAPI/activities/level"
        }
      }
    ]
  },
  "extensions": {
    "https://py-rates.org/xAPI/extensions/game-time": "PT73S"
  }
},
"timestamp": "2021-08-26T23:37:14.418+02:00"
}
```

# GAnalyse à priori des situations

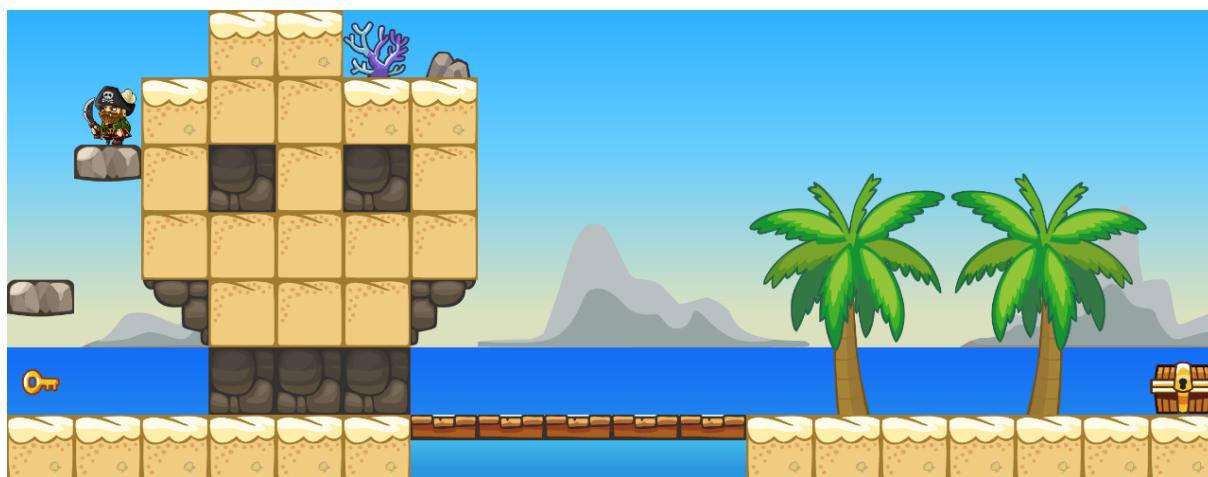
## G.1 Niveau 1

### G.1.1 Description

**Objectif** : Ramasser la clé puis ouvrir le coffre.

**Fonctions de contrôle** : avancer(), droite(), gauche(), ouvrir()

**Contraintes** : Le programme ne doit pas dépasser 10 lignes.



### G.1.2 Savoirs en jeu

xxx = concepts introduits dans ce niveau

Concepts fondamentaux
Utiliser une fonction sans retour et sans argument (avancer, droite, gauche, ouvrir)
Utiliser une boucle for sans variable de boucle (une seule instruction dans le corps de la boucle)

### G.1.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Action à répéter 16 fois : avancer vers le coffre	Favorise l'utilisation d'une boucle for
Taille du programme limitée à 10 lignes	Rend nécessaire l'utilisation d'une boucle for

### G.1.4 Stratégies gagnantes

Description	Code
-------------	------

<p><b>[Stratégie visée]</b>  S1.1 (for-re) :  Ramasser la clé en enchainant séquentiellement les instructions de déplacement.  Parcourir la ligne droite de 16 blocs à l'aide d'une boucle for jusqu'au coffre.  Ouvrir le coffre.  <u>Concepts</u> :  FOR-RE</p>	<pre> 1 avancer() 2 droite() 3 avancer() 4 gauche() 5 avancer() 6 droite() 7 for _ in range(16): 8     avancer() 9 ouvrir() </pre>
<p>S1.1 (for-re_variante) :  Il est possible d'avancer une fois de plus avant d'ouvrir le coffre.  <u>Concepts</u> :  FOR-RE</p>	<pre> 1 avancer() 2 droite() 3 avancer() 4 gauche() 5 avancer() 6 droite() 7 for _ in range(17): 8     avancer() 9 ouvrir() </pre>
<p>S1.1 (for-re_variante) :  Utiliser deux boucles for pendant le parcours  <u>Concepts</u> :  FOR-RE, FOR-RE</p>	<pre> 1 for _ in range(2): 2     avancer() 3     droite() 4     avancer() 5     gauche() 6 droite() 7 for _ in range(15): 8     avancer() 9 ouvrir() </pre>

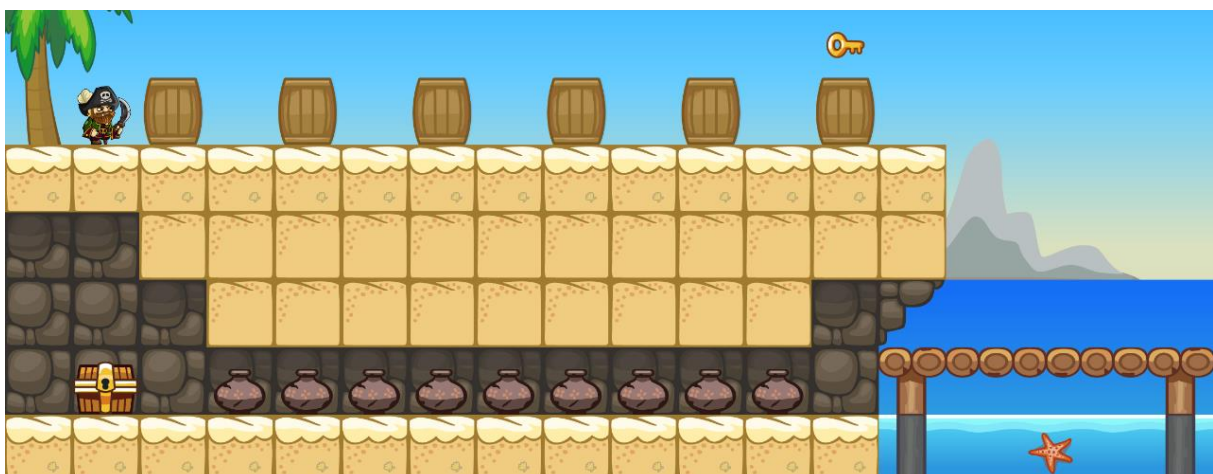
## G.2 Niveau 2

### G.2.1 Description

**Objectif** : Ramasser la clé puis ouvrir le coffre.

**Fonctions de contrôle** : avancer(), droite(), gauche(), sauter(), coup(), ouvrir()

**Contraintes** : Le programme ne doit pas dépasser 14 lignes.



## G.2.2 Savoirs en jeu

xxx = concepts introduits dans ce niveau

Concepts fondamentaux
Utiliser une fonction sans retour et sans argument (avancer, droite, gauche, sauter, coup, ouvrir)
Utiliser une boucle for sans variable de boucle (plusieurs instructions dans le corps de la boucle)

## G.2.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Action à répéter 6 fois : sauter un tonneau puis avancer	Favorise l'utilisation d'une boucle for dont le corps contient plusieurs instructions
Action à répéter 9 fois : détruire un pot puis avancer	Favorise l'utilisation d'une boucle for dont le corps contient plusieurs instructions
Taille du programme limitée à 14 lignes	Rend nécessaire l'utilisation d'au moins deux boucles for

## G.2.4 Stratégies gagnantes

Description	Code
<p><b>[Stratégie visée]</b>            S2.1 (for-re) :            Sauter par-dessus chacun des 6 tonneaux à l'aide d'une boucle for.            Descendre sur la plateforme inférieure.            Détruire les 9 pots à l'aide d'une boucle for.            Approcher puis ouvrir le coffre.  <u>Concepts</u> :            FOR-RE, FOR-RE</p>	<pre> 1 for _ in range(6): 2     sauter() 3     avancer() 4 avancer() 5 gauche() 6 avancer() 7 avancer() 8 for _ in range(9): 9     coup() 10    avancer() 11 avancer() 12 ouvrir() </pre>
<p>S2.1 (for-re_variante) :            Faire le saut sur la plateforme inférieure en exécutant une fois de plus la boucle sauter-avancer.  <u>Concepts</u> :            FOR-RE, FOR-RE</p>	<pre> 1 for _ in range(7): 2     sauter() 3     avancer() 4 gauche() 5 avancer() 6 avancer() 7 avancer() 8 for _ in range(9): 9     coup() 10    avancer() 11 avancer() 12 ouvrir() </pre>
<p>S2.1 (for-re_variante) :            Inversion des instructions dans la deuxième boucle.  <u>Concepts</u> :            FOR-RE, FOR-RE</p>	<pre> 1 for _ in range(6): 2     sauter() 3     avancer() 4 avancer() 5 gauche() </pre>

	<pre> 6 avancer() 7 for _ in range(9): 8     avancer() 9     coup() 10 avancer() 11 avancer() 12 ouvrir() </pre>
<p>S2.1 (for-re) Utiliser une troisième boucle for pour avancer vers les pots. <u>Concepts</u> : FOR-RE, FOR-RE, FOR-RE</p>	<pre> 1 for _ in range(6): 2     sauter() 3     avancer() 4 avancer() 5 gauche() 6 for _ in range(2): 7     avancer() 8 for _ in range(9): 9     coup() 10    avancer() 11 avancer() 12 ouvrir() </pre>
<p>S2.1 (for-re) : Utiliser une quatrième boucle for pour approcher le coffre <u>Concepts</u> : FOR-RE, FOR-RE, FOR-RE, FOR-RE</p>	<pre> 1 for _ in range(6): 2     sauter() 3     avancer() 4 avancer() 5 gauche() 6 for _ in range(2): 7     avancer() 8 for _ in range(9): 9     coup() 10    avancer() 11 for _ in range(2): 12    avancer() 13 ouvrir() </pre>
<p>S2.1 (for-re) : Utiliser une cinquième boucle for pour avancer sur la plateforme. <u>Concepts</u> : FOR-RE, FOR-RE, FOR-RE, FOR-RE, FOR-RE</p>	<pre> 1 for _ in range(6): 2     sauter() 3     avancer() 4 for _ in range(2): 5     avancer() 6 gauche() 7 for _ in range(3): 8     avancer() 9 for _ in range(9): 10    coup() 11    avancer() 12 for _ in range(2): 13    avancer() 14 ouvrir() </pre>

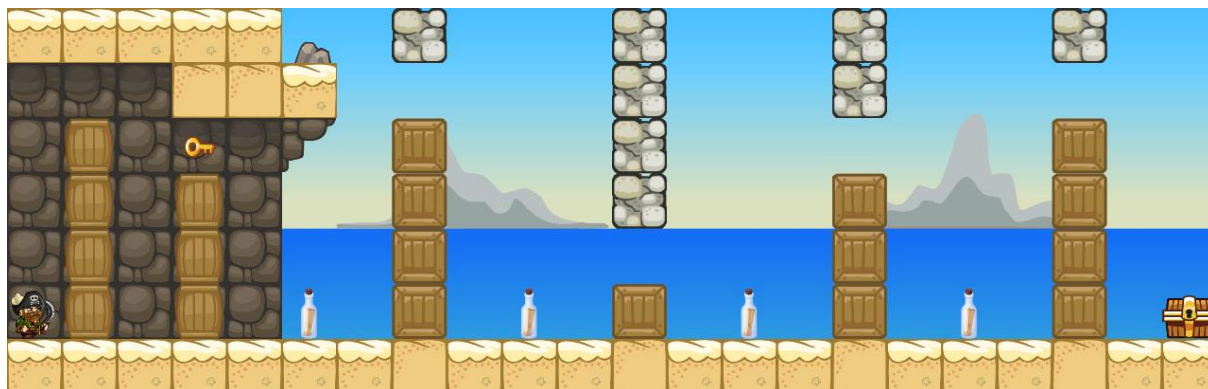
## G.3 Niveau 3

### G.3.1 Description

**Objectif** : Ramasser la clé puis ouvrir le coffre. Les deux premières piles de tonneaux ont une hauteur fixe. Ensuite, la hauteur des piles de caisses est aléatoire et change à chaque exécution. Il faut donc lire le message dans la bouteille avant chaque pile qui indique sa hauteur.

**Fonctions de contrôle** : avancer(), sauter\_hauteur(hauteur), lire\_nombre(), ouvrir()

**Contraintes** : Le programme ne doit pas dépasser 14 lignes.



### G.3.2 Savoirs en jeu

xxx = concepts introduits dans ce niveau

Concepts fondamentaux
Utiliser une fonction sans retour et sans argument (avancer, ouvrir)
Utiliser une fonction sans retour et avec argument (sauter_hauteur)
Utiliser une fonction avec retour et sans argument (lire_nombre)
Affecter et utiliser une variable
Manipuler une variable de type entier
Utiliser une boucle for sans variable de boucle (plusieurs instructions dans le corps de la boucle)

### G.3.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Présence de plusieurs hauteurs de saut (de 1 à 5)	Rend nécessaire l'utilisation d'une fonction avec un argument (sauter_hauteur)
Hauteurs aléatoires des piles de caisses (entre 1 et 5) associé à la répétition de quatre piles donne $5^4 = 625$ parcours aléatoires différents	Augmente de coût de la stratégie consistant à coder un parcours particulier (S3.2) : $p = \frac{1}{5^4} = \frac{1}{625} = 0,16\%$ Favorise l'utilisation d'une fonction avec retour (lire_nombre pour connaître la hauteur des piles)
Présence de « murs » au-dessus des caisses	Empêche l'enchaînement de 4 sauts de hauteur maximale
Distance d'1 bloc entre la bouteille et la pile de caisses	Rend nécessaire l'affectation d'une variable afin de garder en mémoire la valeur contenue dans la bouteille

Action à répéter 4 fois : lire le message dans la bouteille avancer et sauter de la hauteur indiquée dans le message	Favorise l'utilisation d'une boucle for dont le corps contient plusieurs instructions
Taille du programme limitée à 14 lignes	Rend nécessaire l'utilisation d'une boucle for

### G.3.4 Stratégies gagnantes

Description	Code
<p><b>[Stratégie visée]</b>  S3.1 (var-af):  Sauter les deux premières piles de tonneaux.  Répéter quatre fois à l'aide d'une boucle for :</p> <ul style="list-style-type: none"> <li>- lire la hauteur d'une pile de caisses dans la bouteille ;</li> <li>- la stocker dans une variable ;</li> <li>- avancer jusqu'à la pile ;</li> <li>- sauter par-dessus la pile de caisse.</li> </ul> <p>Ouvrir le coffre.  <u>Concepts</u> :  FOR-RE, VAR-AF</p>	<pre> 1 sauter_hauteur(4) 2 avancer() 3 sauter_hauteur(3) 4 avancer() 5 avancer() 6 for _ in range(4): 7     message = lire_nombre() 8     avancer() 9     sauter_hauteur(message) 10 avancer() 11 avancer() 12 ouvrir() </pre>
<p>S3.1 (var-af_variante)  Changement du motif répété dans la boucle.  <u>Concepts</u> :  FOR-RE, VAR-AF</p>	<pre> 1 sauter_hauteur(4) 2 avancer() 3 sauter_hauteur(3) 4 for _ in range(4): 5     avancer() 6     avancer() 7     message = lire_nombre() 8     avancer() 9     sauter_hauteur(message) 10 avancer() 11 avancer() 12 ouvrir() </pre>
<p>S3.1 (var-af_variante)  Changement du motif répété dans la boucle.  <u>Concepts</u> :  FOR-RE, VAR-AF</p>	<pre> 1 sauter_hauteur(4) 2 avancer() 3 sauter_hauteur(3) 4 avancer() 5 for _ in range(4): 6     avancer() 7     message = lire_nombre() 8     avancer() 9     sauter_hauteur(message) 10 avancer() 11 avancer() 12 ouvrir() </pre>
<p>S3.1 (var-af_variante):  Utilisation d'une deuxième boucle for pour avancer vers la première bouteille.  <u>Concepts</u> :  FOR-RE, FOR-RE, VAR-AF</p>	<pre> 1 sauter_hauteur(4) 2 avancer() 3 sauter_hauteur(3) 4 for _ in range(2): 5     avancer() 6 for _ in range(4): </pre>

	<pre> 7     message = lire_nombre() 8     avancer() 9     sauter_hauteur(message) 10    avancer() 11    avancer() 12    ouvrir() </pre>
<p>S3.1 (var-af_variante): Utilisation d'une troisième boucle for pour avancer vers les bouteilles.</p> <p><u>Concepts</u> : FOR-RE, FOR-RE, FOR-RE, VAR-AF</p>	<pre> 1     sauter_hauteur(4) 2     avancer() 3     sauter_hauteur(3) 4     for _ in range(2): 5         avancer() 6     for _ in range(4): 7         message = lire_nombre() 8         avancer() 9         sauter_hauteur(message) 10        for _ in range(2): 11            avancer() 12    ouvrir() </pre>
<p>S3.1 (var-af_variante): Lecture inutile en dehors de la boucle de la première bouteille.</p> <p><u>Concepts</u> : FOR-RE, VAR-AF, VAR-AF</p>	<pre> 1     sauter_hauteur(4) 2     avancer() 3     sauter_hauteur(3) 4     avancer() 5     avancer() 6     message = lireNombre() 7     for _ in range(4): 8         message = lireNombre() 9         avancer() 10        sauter_hauteur(message) 11        avancer() 12        avancer() 13    ouvrir() </pre>
<p>S3.1 (var-af_variante): Lecture inutile de la première bouteille en dehors de la boucle.</p> <p><u>Concepts</u> : FOR-RE, FOR-RE, VAR-AF, VAR-AF</p>	<pre> 1     sauter_hauteur(4) 2     avancer() 3     sauter_hauteur(3) 4     for _ in range(2): 5         avancer() 6     message = lire_nombre() 7     for _ in range(4): 8         message = lire_nombre() 9         avancer() 10        sauter_hauteur(message) 11        avancer() 12        avancer() 13    ouvrir() </pre>
<p>S3.2 (no-var-af) : Codage d'un parcours « en dur » sans soucier de la hauteur aléatoire des caisses.</p> <p><u>Concepts</u> : FOR-RE</p>	<pre> 1     sauter_hauteur(4) 2     avancer() 3     sauter_hauteur(3) 4     avancer() 5     avancer() 6     lire_nombre() 7     avancer() </pre>



```

8 sauter_hauteur(4)
9 avancer()
10 avancer()
11 for _ in range (12):
12     avancer()
13 sauter_hauteur(2)
14 ouvrir()

```

## G.4 Niveau 4

### G.4.1 Description

**Objectif** : Ramasser la clé puis ouvrir le coffre. L'emplacement du coffre et de la clé sont aléatoires et changent à chaque exécution. Il faut donc lire les messages dans les bouteilles qui indiquent le chemin à suivre (gauche ou droite) à chaque étage. Le fait de tomber dans les piques fait perdre puis recommencer le niveau.

**Fonctions de contrôle** : avancer(), gauche(), droite(), lire\_chaine(), ouvrir()

**Contraintes** : Le programme ne doit pas dépasser 18 lignes.



### G.4.2 Savoirs en jeu

xxx = concepts introduits dans ce niveau

Notions algorithmiques
Utiliser une fonction sans retour et sans argument (avancer, gauche, droite, ouvrir)
Utiliser une fonction avec retour et sans argument (lire_chaine)
Affecter et utiliser une variable
Manipuler une variable de type chaîne de caractères
Utiliser une structure conditionnelle à deux branches

### Tester une égalité

Utiliser une boucle for sans variable de boucle (plusieurs instructions dans le corps de la boucle)

## G.4.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Emplacement aléatoire de la clé, du coffre et des bouteilles sur 5 niveaux de plateforme. Donne $2^5 = 32$ parcours différents.	Augmente de coût de la stratégie consistant à coder un parcours particulier (S4.2) : $p = \frac{1}{32} \approx 3,1\%$ Favorise l'utilisation d'une fonction avec retour (lire_chaine pour connaître le chemin à suivre) Favorise l'utilisation d'une structure conditionnelle à deux branches et d'un test d'égalité pour suivre l'indication du message.
Message dans les bouteilles de type chaînes de caractères	Rend nécessaire la manipulation de chaînes de caractères
Action à répéter 5 fois : lire le message dans la bouteille, s'orienter en fonction du message, et descendre de la plateforme	Favorise l'utilisation d'une boucle for dont le corps contient plusieurs instructions
Taille du programme limitée à 14 lignes	Rend nécessaire l'utilisation d'une boucle for

## G.4.4 Stratégies gagnantes

Description	Code
<b>[Stratégie visée]</b> S4.1 (if-else) : Avancer jusqu'à la première bouteille. Utiliser une boucle for pour répéter cinq fois les instructions : - appeler la fonction lire_chaine et affecter une variable du retour de cette fonction ; - réaliser un test sur cette variable à l'aide d'une structure conditionnelle if-else permettant d'aller à gauche ou à droite ; - avancer de deux blocs pour descendre d'un étage. Ouvrir le coffre. <u>Concepts</u> : FOR-RE, VAR-AF, IF, ELSE	<pre>1 avancer() 2 for _ in range(5): 3     msg = lire_chaine() 4     if msg == "gau": 5         gauche() 6     else: 7         droite() 8     avancer() 9     avancer() 10 ouvrir()</pre>
S4.1 (if-else_variante) : Utilisation d'une deuxième boucle for. <u>Concepts</u> : FOR-RE, VAR-AF, IF, ELSE,	<pre>1 avancer() 2 for _ in range(5): 3     msg = lire_chaine() 4     if msg == "gau": 5         gauche() 6     else: 7         droite() 8         for _ in range(2): 9             avancer() 10 ouvrir()</pre>

<p>S4.1 (if-else_variante) :</p> <p>Non « factorisation » de la fonction avancer à la suite de la conditionnelle.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 for _ in range(5): 3     msg = lire_chaine() 4     if msg == "gau": 5         gauche() 6         avancer() 7         avancer() 8     else: 9         droite() 10        avancer() 11        avancer() 12 ouvrir() </pre>
<p>S4.1 (if-else_variante) :</p> <p>Modification de l'ordre des instructions dans la boucle for.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 for _ in range (5): 2     avancer() 3     msg = lire_chaine() 4     if msg == "gau": 5         gauche() 6     else : 7         droite() 8     avancer() 9 avancer() 10 ouvrir() </pre>
<p>S4.1 (if-else_variante) :</p> <p>Non « factorisation » de la fonction avancer à la suite de la conditionnelle.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 for _ in range (5): 2     avancer() 3     msg = lire_chaine() 4     if msg == "gau": 5         gauche() 6         avancer() 7     else: 8         droite() 9         avancer() 10 avancer() 11 ouvrir() </pre>
<p>S4.1 (if-else_variante) :</p> <p>Faire l'appel de fonction directement dans la condition.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, IF, ELSE</p>	<pre> 1 avancer() 2 for _ in range(5): 3     if lire_chaine() == "gau": 4         gauche() 5     else: 6         droite() 7     avancer() 8     avancer() 9 ouvrir() </pre>
<p>S4.1 (if-else_variante) :</p> <p>Non « factorisation » de la fonction avancer à la suite de la conditionnelle.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, IF, ELSE</p>	<pre> 1 avancer() 2 for _ in range(3): 3     if lire_chaine() == "gau": 4         gauche() 5         avancer() 6         avancer() 7     else: 8         droite() 9         avancer() </pre>

	10 11	<pre>    avancer() ouvrir()</pre>
S4.1 (if-else_variante) : Modification de l'ordre des instructions dans la boucle <u>Concepts</u> : FOR-RE, IF, ELSE	1 2 3 4 5 6 7 8 9	<pre>for _ in range (3):     avancer()     if lire_chaine() == "gau":         gauche()     else:         droite()     avancer() avancer() ouvrir()</pre>
S4.1 (if-else_variante) : Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, IF, VAR-AF, ELSE	1 2 3 4 5 6 7 8 9 10	<pre>for _ in range (3):     avancer()     if lire_chaine() == "gau":         gauche()         avancer()     else:         droite()         avancer() avancer() ouvrir()</pre>
S4.2 (if-if) : Utiliser deux structures conditionnelles if. (non optimal du point de vue de l'efficacité algorithmique) <u>Concepts</u> : FOR-RE, IF, VAR-AF, IF	1 2 3 4 5 6 7 8 9 10	<pre>avancer() for _ in range(4):     msg = lire_chaine()     if msg == "gau":         gauche()     if msg == "droi":         droite()     avancer() avancer() ouvrir()</pre>
S4.2 (if-if_variante) : Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, IF, VAR-AF, IF	1 2 3 4 5 6 7 8 9 10 11 12	<pre>avancer() for _ in range(4):     msg = lire_chaine()     if msg == "gau":         gauche()         avancer()     avancer()     if msg == "droi":         droite() avancer() avancer() ouvrir()</pre>
S4.2 (if-if_variante) : Modification de l'ordre des instructions dans la boucle <u>Concepts</u> : FOR-RE, IF, VAR-AF, IF	1 2 3 4 5 6 7 8	<pre>for _ in range (4):     avancer()     msg = lire_chaine()     if msg == "gau":         gauche()     if msg == "droi":         droite()     avancer()</pre>

	9 10	<code>avancer() ouvrir()</code>
S4.2 (if-if_variante) : Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, IF, VAR-AF, IF	1 2 3 4 5 6 7 8 9 10 11	<code>for _ in range (4):     avancer()     msg = lire_chaine()     if msg == "gau" :         gauche()         avancer()     if msg == "droi":         droite()         avancer() avancer() ouvrir()</code>
S4.2 (if-if_variante) : Faire l'appel de fonction directement dans la condition. <u>Concepts</u> : FOR-RE, IF, IF	1 2 3 4 5 6 7 8 9	<code>avancer() for _ in range(4):     if lire_chaine() == "gau":         gauche()     if lire_chaine() == "droi":         droite()     avancer()     avancer() ouvrir()</code>
S4.2 (if-if_variante) : Modification de l'ordre des instructions dans la boucle. <u>Concepts</u> : FOR-RE, IF, IF	1 2 3 4 5 6 7 8 9	<code>for _ in range (4):     avancer()     if lire_chaine() == "gau":         gauche()     if lire_chaine() == "droi":         droite()     avancer() avancer() ouvrir()</code>
S4.3 (if-elif) Utiliser une structure conditionnelle if-elif. (non optimal du point de vue de l'efficacité algorithmique) <u>Concepts</u> : FOR-RE, VAR-AF, IF, ELIF	1 2 3 4 5 6 7 8 9 10	<code>avancer() for _ in range(4):     msg = lire_chaine()     if msg == "gau":         gauche()     elif msg == "droi":         droite()     avancer()     avancer() ouvrir()</code>
S4.3 (if-elif variante) Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, VAR-AF, IF, ELIF	1 2 3 4 5 6 7 8 9 10	<code>avancer() for _ in range(4):     msg = lire_chaine()     if msg == "gau":         gauche()         avancer()         avancer()     elif msg == "droi":         droite()         avancer()</code>

	11	<code>avancer()</code>
	12	<code>ouvrir()</code>
S4.3 (if-elif_variante) Modification de l'ordre des instructions dans la boucle <u>Concepts</u> : FOR-RE, VAR-AF, IF, ELIF	1	<code>for _ in range (4) :</code>
	2	<code>avancer()</code>
	3	<code>msg = lire_chaine()</code>
	4	<code>if msg == "gau":</code>
	5	<code>gauche()</code>
	6	<code>elif msg == "droi":</code>
	7	<code>droite()</code>
	8	<code>avancer()</code>
	9	<code>avancer()</code>
	10	<code>ouvrir()</code>
S4.3 (if-elif_variante) Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, VAR-AF, IF, ELIF	1	<code>for _ in range (4) :</code>
	2	<code>avancer()</code>
	3	<code>msg = lire_chaine()</code>
	4	<code>if msg == "gau":</code>
	5	<code>gauche()</code>
	6	<code>avancer()</code>
	7	<code>elif msg == "droi":</code>
	8	<code>droite()</code>
	9	<code>avancer()</code>
	10	<code>avancer()</code>
	11	<code>ouvrir()</code>
S4.3 (if-elif_variante) Faire l'appel de fonction directement dans la condition. <u>Concepts</u> : FOR-RE, IF, ELIF	1	<code>avancer()</code>
	2	<code>for _ in range(4):</code>
	3	<code>if lire_chaine() == "gau":</code>
	4	<code>gauche()</code>
	5	<code>elif lire_chaine() == "droi":</code>
	6	<code>droite()</code>
	7	<code>avancer()</code>
	8	<code>avancer()</code>
	9	<code>ouvrir()</code>
S4.3 (if-elif_variante) Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, IF, ELIF	1	<code>avancer()</code>
	2	<code>for _ in range(4):</code>
	3	<code>if lire_chaine() == "gau":</code>
	4	<code>gauche()</code>
	5	<code>avancer()</code>
	6	<code>avancer()</code>
	7	<code>elif lire_chaine() == "droi":</code>
	8	<code>droite()</code>
	9	<code>avancer()</code>
	10	<code>avancer()</code>
	11	<code>ouvrir()</code>
S4.3 (if-elif_variante) Modification de l'ordre des instructions dans la boucle <u>Concepts</u> : FOR-RE, IF, ELIF	1	<code>for _ in range (4) :</code>
	2	<code>avancer()</code>
	3	<code>if lire_chaine() == "gau":</code>
	4	<code>gauche()</code>
	5	<code>elif lire_chaine() == "droi":</code>
	6	<code>droite()</code>
	7	<code>avancer()</code>
	8	<code>avancer()</code>

	9	<code>ouvrir()</code>
S4.3 (if-elif_variante) Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, IF, ELIF	1 2 3 4 5 6 7 8 9 10	<code>for _ in range (4):     avancer()     if lire_chaine() == "gau":         gauche()         avancer()     elif lire_chaine() == "droi":         droite()         avancer() avancer() ouvrir()</code>
S4.4 (no-cond) : Essayer un parcours particulier. <u>Concepts</u> : $\emptyset$	1 2 3 4 5 6 7 8 9	<code>gauche() avancer() avancer() avancer() avancer() avancer() avancer() avancer() ouvrir()</code>
S4.4 (no-cond_variante) : Essayer un parcours particulier en utilisant une boucle. <u>Concepts</u> : FOR-RE	1 2 3	<code>for _ in range(9):     avancer() ouvrir()</code>
S4.4 (no-cond_variante)) Essayer un parcours particulier en utilisant deux boucles <u>Concepts</u> : FOR-RE, FOR RE	1 2 3 4 5	<code>for _ in range(6):     avancer() for _ in range(3):     avancer() ouvrir()</code>

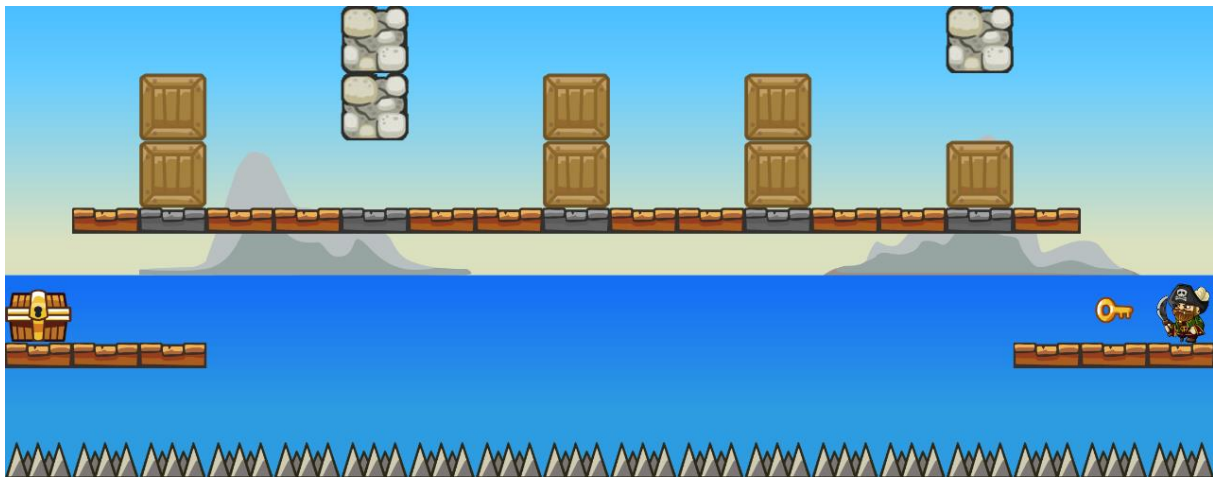
## G.5 Niveau 5

### G.5.1 Description

**Objectif** : Ramasser la clé puis ouvrir le coffre. Sur la plateforme supérieure, la hauteur des caisses et des murs est aléatoire et change à chaque exécution. Le fait de tomber dans les piques fait perdre puis recommencer le niveau.

**Fonctions de contrôle** : `avancer()`, `sauter()`, `sauter_haut()`, `mesurer_hauteur()`, `ouvrir()`

**Contraintes** : Le programme ne doit pas dépasser 18 lignes.



### G.5.2 Savoirs en jeu

xxx = concepts introduits dans ce niveau

Concepts fondamentaux
Utiliser une fonction sans retour et sans argument (avancer, sauter, sauter_haut, ouvrir)
Utiliser une fonction avec retour et sans argument (mesurer_hauteur)
Affecter et utiliser une variable
Manipuler une variable de type entier
Utiliser une structure conditionnelle à trois branches
Tester une égalité
Utiliser une boucle for sans variable de boucle (plusieurs instructions dans le corps de la boucle)

### G.5.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Hauteurs aléatoires des piles de caisses (entre 0 et 2) associé à la répétition de cinq piles donne $3^5 = 243$ parcours aléatoires différents	Augmente de coût de la stratégie consistant à coder un parcours particulier (S5.4) : $p = \frac{1}{3^5} = \frac{1}{243} \approx 0,4\%$ Favorise l'utilisation d'une fonction avec retour (mesurer_hauteur pour connaître la hauteur des piles des caisses) Favorise l'utilisation d'une structure conditionnelle et d'un test d'égalité
Pas d'accès à la fonction sauter_hauteur mais présence de trois fonction différentes pour passer les obstacles selon leur hauteur : avancer, sauter et sauter_haut	Favorise l'utilisation d'une structure conditionnelle à trois branches
Présence de « murs » au-dessus des caisses	Empêche l'enchaînement de 5 sauts de hauteur maximale
Action à répéter 5 fois : mesurer la hauteur des caisses,	Favorise l'utilisation d'une boucle for dont le corps contient plusieurs instructions
Taille du programme limitée à 18 lignes	Rend nécessaire l'utilisation d'une boucle for

### G.5.4 Stratégies gagnantes



Description	Code
<p><b>[Stratégie visée]</b>  S5.1 (if-elif-else) :  Ramasser la clé sur la plateforme inférieure, puis monter à l'aide d'un saut haut.  Utiliser une boucle for pour répéter cinq fois les instructions :  - mesurer la hauteur des caisses se situant devant soi à l'aide de la fonction mesurer_hauteur puis affecter une variable à l'aide de cette valeur ;  - utiliser une structure conditionnelle if-elif-else afin d'effectuer l'action adéquate en fonction de la hauteur des caisses (avancer, sauter ou sauter_haut) ;  - avancer deux fois jusqu'aux prochaines caisses.  Avancer pour descendre sur la plateforme inférieure puis ouvrir le coffre.  <u>Concepts</u> :  FOR-RE, VAR-AF, IF, ELIF, ELSE</p>	<pre> 1 avancer() 2 sauter_haut() 3 for _ in range(5): 4     hauteur = mesurer_hauteur() 5     if hauteur == 0: 6         avancer() 7     elif hauteur == 1: 8         sauter() 9     else: 10        sauter_haut() 11        avancer() 12        avancer() 13 ouvrir() </pre>
<p>S5.1 (if-elif-else_variante) :  Utiliser une boucle for supplémentaire  <u>Concepts</u> :  FOR-RE, VAR-AF, IF, ELIF, ELSE, FOR-RE</p>	<pre> 1 avancer() 2 sauter_haut() 3 for _ in range(5): 4     hauteur = mesurer_hauteur() 5     if hauteur == 0: 6         avancer() 7     elif hauteur == 1: 8         sauter() 9     else: 10        sauter_haut() 11        for _ in range(2): 12            avancer() 13 ouvrir() </pre>
<p>S5.1 (if-elif-else_variante) :  Non « factorisation » de la fonction avancer à la suite de la conditionnelle.  <u>Concepts</u> :  FOR-RE, VAR-AF, IF, ELIF, ELSE</p>	<pre> 1 avancer() 2 sauter_haut() 3 for _ in range(5): 4     hauteur = mesurer_hauteur() 5     if hauteur == 0: 6         avancer() 7         avancer() 8         avancer() 9     elif hauteur == 1: 10        sauter() 11        avancer() 12        avancer() 13    else: 14        sauter_haut() 15        avancer() 16        avancer() 17 ouvrir() </pre>

<p>S5.1 (if-elif-else_variante) :</p> <p>Détection d'obstacle tout au long de plateforme supérieure (et pas seulement sur les 5 emplacements affectés par le hasard).</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, ELIF, ELSE</p>	<p>1 2 3 4 5 6 7 8 9 10 11</p>	<pre> avancer() sauter_haut() for _ in range(15):     hauteur = mesurer_hauteur()     if hauteur == 0:         avancer()     elif hauteur == 1:         sauter()     else:         sauter_haut() ouvrir() </pre>
<p>S5.1 (if-elif-else_variante) :</p> <p>Faire l'appel de fonction directement dans la condition du if et du elif.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, IF, ELIF, ELSE</p>	<p>1 2 3 4 5 6 7 8 9 10 11 12</p>	<pre> avancer() sauter_haut() for _ in range(5):     if mesurer_hauteur() == 0:         avancer()     elif mesurer_hauteur() == 1:         sauter()     else:         sauter_haut() avancer() avancer() ouvrir() </pre>
<p>S5.1 (if-elif-else_variante) :</p> <p>Non « factorisation » de la fonction avancer à la suite de la conditionnelle.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, IF, ELIF, ELSE</p>	<p>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</p>	<pre> avancer() sauter_haut() for _ in range(4):     if mesurer_hauteur() == 0:         avancer()         avancer()         avancer()     elif mesurer_hauteur() == 1:         sauter()         avancer()         avancer()     else:         sauter_haut()         avancer()         avancer() ouvrir() </pre>
<p>S5.1 (if-elif-else_variante) :</p> <p>Détection d'obstacle tout au long de plateforme supérieure (et pas seulement sur les 4 emplacements affectés par le hasard).</p> <p><u>Concepts</u> :</p> <p>FOR-RE, IF, ELIF, ELSE</p>	<p>1 2 3 4 5 6 7 8 9 10</p>	<pre> avancer() sauter_haut() for _ in range(14):     if mesurer_hauteur() == 0:         avancer()     elif mesurer_hauteur() == 1:         sauter()     else:         sauter_haut() ouvrir() </pre>
<p>S5.2 (if-if-if) :</p>	<p>1 2</p>	<pre> avancer() sauter_haut() </pre>

<p>Utiliser trois structures conditionnelles if. (non optimal du point de vue de l'efficacité algorithmique)</p> <p><u>Concepts</u> : FOR-RE, VAR-AF, IF, IF, IF</p>	<p>3 4 5 6 7 8 9 10 11 12 13</p>	<pre> for _ in range(5):     hauteur = mesurer_hauteur()     if hauteur == 0:         avancer()     if hauteur == 1:         sauter()     if hauteur == 2:         sauter_haut()     avancer()     avancer() ouvrir() </pre>
<p>S5.2 (if-if-if_variante) : Non « factorisation » de la fonction avancer à la suite de la conditionnelle.</p> <p><u>Concepts</u> : FOR-RE, VAR-AF, IF, IF, IF</p>	<p>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17</p>	<pre> avancer() sauter_haut() for _ in range(5):     hauteur = mesurer_hauteur()     if hauteur == 0:         avancer()         avancer()         avancer()     if hauteur == 1:         sauter()         avancer()         avancer()     if hauteur == 2:         sauter_haut()         avancer()         avancer() ouvrir() </pre>
<p>S5.2 (if-if-if_variante) : Détection d'obstacle tout au long de plateforme supérieure (et pas seulement sur les 4 emplacements affectés par le hasard).</p> <p><u>Concepts</u> : FOR-RE, VAR-AF, IF, IF, IF</p>	<p>1 2 3 4 5 6 7 8 9 10 11</p>	<pre> avancer() sauter_haut() for _ in range(14):     hauteur = mesurer_hauteur()     if hauteur == 0:         avancer()     if hauteur == 1:         sauter()     if hauteur == 2:         sauter_haut() ouvrir() </pre>
<p>S5.2 (if-if-if_variante) : Faire l'appel de fonction directement dans la condition des if.</p> <p><u>Concepts</u> : FOR-RE, IF, IF, IF</p>	<p>1 2 3 4 5 6 7 8 9 10 11 12</p>	<pre> avancer() sauter_haut() for _ in range(5):     if mesurer_hauteur() == 0:         avancer()     if mesurer_hauteur() == 1:         sauter()     if mesurer_hauteur() == 2:         sauter_haut()     avancer()     avancer() ouvrir() </pre>

<p>S5.3 (if-elif-elif) :</p> <p>Utiliser une structure conditionnelle if-elif-elif. (non optimal du point de vue de l'efficacité algorithmique)</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, ELIF, ELIF</p>	<pre> 1 avancer() 2 sauter_haut() 3 for _ in range(5): 4     hauteur = mesurer_hauteur() 5     if hauteur == 0: 6         avancer() 7     elif hauteur == 1: 8         sauter() 9     elif hauteur == 2: 10        sauter_haut() 11    avancer() 12    avancer() 13 ouvrir() </pre>
<p>S5.3 (if-elif-elif_variante) :</p> <p>Non « factorisation » de la fonction avancer à la suite de la conditionnelle.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, ELIF, ELIF</p>	<pre> 1 avancer() 2 sauter_haut() 3 for _ in range(5): 4     hauteur = mesurer_hauteur() 5     if hauteur == 0: 6         avancer() 7         avancer() 8         avancer() 9     elif hauteur == 1: 10        sauter() 11        avancer() 12        avancer() 13    elif hauteur == 2: 14        sauter_haut() 15        avancer() 16        avancer() 17 ouvrir() </pre>
<p>S5.3 (if-elif-elif_variante) :</p> <p>Détection d'obstacle tout au long de plateforme supérieure (et pas seulement sur les 5 emplacements affectés par le hasard).</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, ELIF, ELIF</p>	<pre> 1 avancer() 2 sauter_haut() 3 for _ in range(14): 4     hauteur = mesurer_hauteur() 5     if hauteur == 0: 6         avancer() 7     elif hauteur == 1: 8         sauter() 9     elif hauteur == 2: 10        sauter_haut() 11 ouvrir() </pre>
<p>S5.3 (if-elif-elif_variante) :</p> <p>Faire l'appel de fonction directement dans la condition du if et des elif.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, IF, ELIF, ELIF</p>	<pre> 1 avancer() 2 sauter_haut() 3 for _ in range(5): 4     if mesurer_hauteur() == 0: 5         avancer() 6     elif mesurer_hauteur() == 1: 7         sauter() 8     elif mesurer_hauteur() == 2: 9         sauter_haut() 10    avancer() </pre>

	11	<code>avancer()</code>
	12	<code>ouvrir()</code>
S5.3 (if-elif-elif_variante) : Non « factorisation » de la fonction avancer à la suite de la conditionnelle. <u>Concepts</u> : FOR-RE, IF, ELIF, ELIF	1	<code>avancer()</code>
	2	<code>sauter_haut()</code>
	3	<code>for _ in range(4):</code>
	4	<code>    if mesurer_hauteur() == 0:</code>
	5	<code>        avancer()</code>
	6	<code>        avancer()</code>
	7	<code>        avancer()</code>
	8	<code>    elif mesurer_hauteur() == 1:</code>
	9	<code>        sauter()</code>
	10	<code>        avancer()</code>
	11	<code>        avancer()</code>
	12	<code>    elif mesurer_hauteur() == 2:</code>
	13	<code>        sauter_haut()</code>
	14	<code>        avancer()</code>
	15	<code>        avancer()</code>
	16	<code>ouvrir()</code>
S5.3 (if-elif-elif_variante) : Détection d'obstacle tout au long de plateforme supérieure (et pas seulement sur les 5 emplacements affectés par le hasard). <u>Concepts</u> : FOR-RE, IF, ELIF, ELIF	1	<code>avancer()</code>
	2	<code>sauter_haut()</code>
	3	<code>for _ in range(14):</code>
	4	<code>    if mesurer_hauteur() == 0:</code>
	5	<code>        avancer()</code>
	6	<code>    elif mesurer_hauteur() == 1:</code>
	7	<code>        sauter()</code>
	8	<code>    elif mesurer_hauteur() == 2:</code>
	9	<code>        sauter_haut()</code>
	10	<code>ouvrir()</code>
S5.4 (no-cond) : Essayer un parcours en particulier. <u>Concepts</u> : ∅	1	<code>avancer()</code>
	2	<code>sauter_haut()</code>
	3	<code>avancer()</code>
	4	<code>sauter()</code>
	5	<code>avancer()</code>
	6	<code>avancer()</code>
	7	<code>sauter()</code>
	8	<code>avancer()</code>
	9	<code>avancer()</code>
	10	<code>sauter()</code>
	11	<code>avancer()</code>
	12	<code>avancer()</code>
	13	<code>sauter()</code>
	14	<code>avancer()</code>
	15	<code>avancer()</code>
	16	<code>avancer()</code>
	17	<code>ouvrir()</code>
S5.4 (no-cond_variante) : Essayer un parcours en particulier en utilisant une boucle for <u>Concepts</u> : FOR-RE	1	<code>avancer()</code>
	2	<code>sauter_haut()</code>
	3	<code>for _ in range(14):</code>
	4	<code>    avancer()</code>
	5	<code>ouvrir()</code>

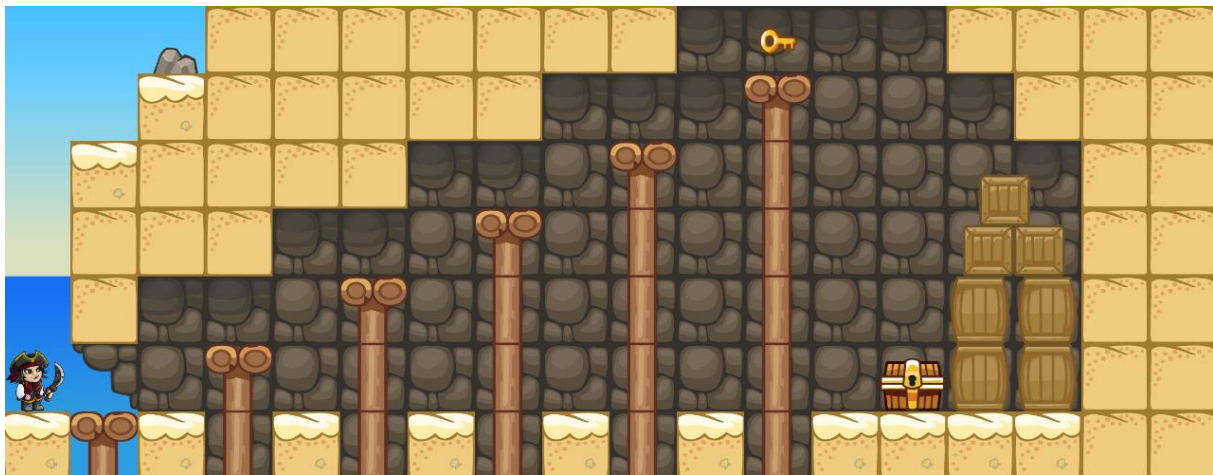
## G.6 Niveau 6

### G.6.1 Description

**Objectif** : Ramasser la clé puis ouvrir le coffre, la hauteur des piliers est fixe.

**Fonctions de contrôle** : avancer(), sauter\_hauteur(hauteur), ouvrir()

**Contraintes** : Le programme ne doit pas dépasser 4 lignes.



### G.6.2 Savoirs en jeu

xxx = concepts introduits dans ce niveau

Concepts fondamentaux
Utiliser une fonction sans retour et sans argument (avancer, ouvrir)
Utiliser une fonction sans retour et avec argument (sauter_hauteur)
Utiliser une variable (variable de boucle)
Utiliser une boucle for avec variable de boucle (commence à zéro)

### G.6.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Action à répéter 6 fois : sauter d'une hauteur qui augmente d'un bloc à chaque pilier	Favorise l'implémentation d'une boucle for en utilisant la variable de boucle
Permettre d'avancer en employant la fonction sauter_hauteur(o)	Favoriser l'utilisation d'une variable de boucle qui commence à zéro
Présence du plafond de la grotte « en escalier »	Empêche l'enchaînement de 6 sauts de hauteur maximale
Taille du programme limitée à 4 lignes	Rend nécessaire l'utilisation d'une boucle for avec variable de boucle

### G.6.4 Stratégies gagnantes

Description	Code
S6.1 (for-co-0) : [Stratégie visée]	1   for compteur in range(6):

Gravir les piliers à l'aide d'une boucle en utilisant la variable de boucle afin de sauter à la bonne hauteur. Ouvrir le coffre. <u>Concepts</u> : FOR-VA-o	2 3 4	<pre>sauter_hauteur(compteur) avancer() ouvrir()</pre>
S6.1 (for-co-o_variante) : Utiliser la fonction sauter_hauteur pour avancer <u>Concepts</u> : FOR-VA-o	1 2 3 4	<pre>for compteur in range(6): sauter_hauteur(compteur) sauter_hauteur(0) ouvrir()</pre>
S6.1 (for-co-o_variante) : Expliciter le début de la variable de boucle à 0 même si cela est inutile (valeur par défaut) <u>Concepts</u> : FOR-VA-o	1 2 3 4	<pre>for compteur in range(0,6): sauter_hauteur(compteur) avancer() ouvrir()</pre>
S6.1 (for-co-o_variante) : Expliciter le pas d'incréméntation de la variable de boucle même si cela est inutile (valeur par défaut) <u>Concepts</u> : FOR-VA-o	1 2 3 4	<pre>for compteur in range(0,6,1): sauter_hauteur(compteur) avancer() ouvrir()</pre>
S6.2 (for-co-n) : Implémenter une boucle for en utilisant la variable de boucle qui ne commence pas à zéro <u>Concepts</u> : FOR-VA-N	1 2 3 4	<pre>for k in range (1,6): sauter_hauteur(k-1) avancer() ouvrir()</pre>

## G.7 Niveau 7

### G.7.1 Description

**Objectif** : Tirer sur les différentes séries de noix de coco qui apparaissent des deux côtés puis, à la fin, sur le coffre. L'autre pirate se place toujours derrière la dernière noix. Le fait de tirer sur ce pirate fait perdre puis recommencer le niveau. Il est possible d'atteindre plusieurs noix de coco à l'aide d'une seule balle.

**Fonctions de contrôle** : tourner, tirer(distance)

**Contraintes** : Le programme ne doit pas dépasser 5 lignes.





## G.7.2 Savoirs en jeu

**xxx** = concepts introduits dans ce niveau

Concepts fondamentaux
Utiliser une fonction sans retour et sans argument (tourner)
Utiliser une fonction sans retour et avec argument (tirer)
Utiliser une variable (variable de boucle)
Utiliser une boucle for avec variable de boucle <b>(ne commence pas à zéro)</b>

## G.7.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Action à répéter 7 fois : tirer sur des séries de noix de coco dont le nombre augmente d'une unité à chaque fois	Favorise l'implémentation d'une boucle for en utilisant la variable de boucle
La première série contient deux noix de coco et la fonction tirer ne peut être utilisée avec un paramètre nul	Favorise l'utilisation d'une variable de boucle qui ne commence pas à zéro
Présence d'un personnage derrière la dernière noix de coco	Empêche l'enchaînement de 7 tirs de distance maximal
Apparition finale du coffre (et du personnage) à une distance de trois blocs	Oblige à effectuer un tir en dehors de la boucle (et à bien calculer le nombre de tours de boucle)
Taille du programme limitée à 5 lignes	Rend nécessaire l'utilisation d'une boucle for avec variable de boucle

## G.7.4 Stratégies gagnantes

Description	Code
<p>S7.1 (for-co-n) : <b>[Stratégie visée]</b>            Tirer alternativement à droite puis à gauche et de plus en plus loin sur les différentes séries de noix de coco à l'aide d'une boucle for et de sa variable de boucle initialisée à 2 à prenant 8 pour dernière valeur.            Tirer sur le coffre.  <u>Concepts</u> :            FOR-VA-N</p>	<pre> 1 for compteur in range(2,9): 2     tirer(compteur) 3     tourner() 4 tirer(3)           </pre>
<p>S7.1 (for-co-n_variante) :            Utiliser deux boucles imbriquées pour tirer sur les noix de coco une à une.  <u>Concepts</u> :            FOR-VA-N, FOR-VA-N</p>	<pre> 1 for cpt in range(3,10): 2     for dist in range(1,cpt): 3         tirer(dist) 4     tourner() 5 tirer(3)           </pre>
<p>S7.1 (for-co-n_variante) :            Utiliser deux boucles imbriquées afin de tirer sur les noix de coco une à une (on tire sept fois quel que soit le nombre de noix)  <u>Concepts</u> :            FOR-RE, FOR-VA-N</p>	<pre> 1 for _ in range (9): 2     for dist in range (2,9): 3         tirer (dist) 4     tourner ()           </pre>



<p>S7.1 (for-co-n_variante) :</p> <p>Utiliser deux boucles imbriquées pour tirer sur les noix de coco une à une.</p> <p><u>Concepts</u> :</p> <p>FOR-VA-N, FOR-VA-o</p>	<pre> 1 for cpt in range(2,9): 2     for dist in range(cpt): 3         tirer(dist+1) 4     tourner() 5 tirer(3) </pre>
<p>S7.2 (for-co-o) :</p> <p>Utiliser une boucle for avec une variable de boucle commençant à zéro.</p> <p><u>Concepts</u> :</p> <p>FOR-VA-o</p>	<pre> 1 for compteur in range (7): 2     tirer(compteur+2) 3     tourner() 4 tirer(3) </pre>
<p>S7.2 (for-co-o_variante) :</p> <p>Même chose avec des variables de boucle qui commencent à zéro</p> <p><u>Concepts</u>:</p> <p>FOR-VA-o, FOR-VA-o</p>	<pre> 1 for i in range(7): 2     for j in range(i+2): 3         tirer(j+1) 4     tourner() 5 tirer(3) </pre>
<p>S7.2 (for-co-o_variante) :</p> <p>Même chose avec une deuxième boucle qui commence à zéro</p> <p><u>Concepts</u> :</p> <p>FOR-RE, FOR-VA-o</p>	<pre> 1 for _ in range (9): 2     for dist in range (7): 3         tirer(dist+2) 4     tourner () </pre>
<p>S7.3 (for-re) :</p> <p>Utiliser une boucle for pour la répétition est gestion «à la main» d'un compteur pour ajuster la distance de tir.</p> <p><b>/!\ Nécessite de mettre deux instructions sur une seule ligne</b></p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, VAR-AF</p>	<pre> 1 dist=2 2 for _ in range(7): 3     tirer(dist); tourner() 4     dist += 1 5 tirer(3) </pre>

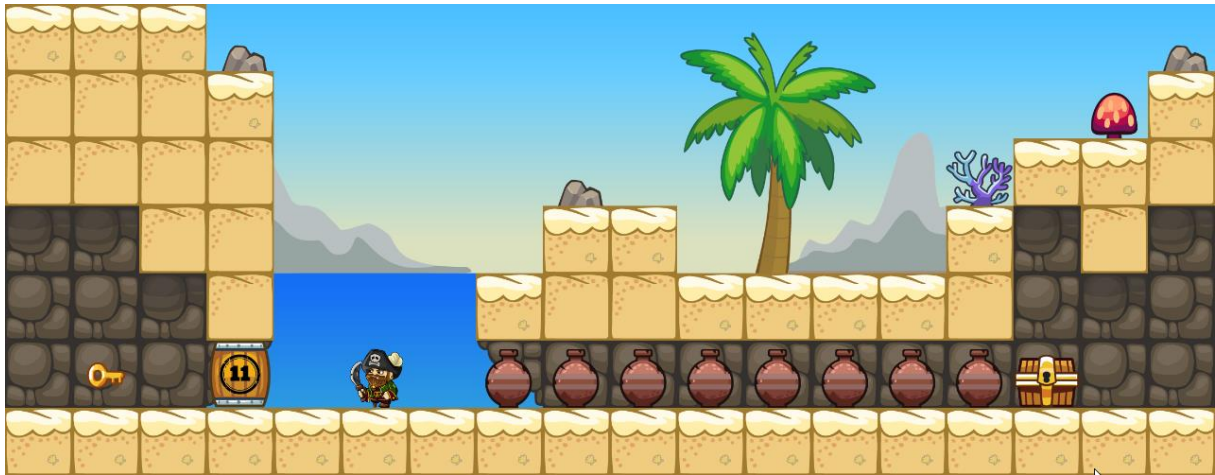
## G.8 Niveau 8

### G.8.1 Description

**Objectif** : Ramasser la clé puis ouvrir le coffre. La solidité du tonneau (nombre dans le cercle), le nombre de pots ainsi que l'emplacement du coffre sont aléatoires et changent à chaque exécution. Un fois le tonneau détruit, de la dynamite apparaît à sa place. Il ne faut pas donner de coup dans cette dynamite sous peine d'une explosion qui fait perdre puis recommencer le niveau.

**Fonctions de contrôle** : avancer(), gauche(), droite(), coup(), detecter\_obstacle(), ouvrir()

**Contraintes** : Aucunes



## G.8.2 Savoirs en jeu

xxx = concepts introduits dans ce niveau

Concepts fondamentaux
Utiliser une fonction sans retour et sans argument (avancer, gauche, droite, coup, ouvrir)
Utiliser une fonction avec retour et sans argument (detecter_obstacle)
Affecter et utiliser une variable
Manipuler une variable de <b>type booléen</b>
Utiliser une <b>boucle while</b>
Utiliser une boucle for sans variable de boucle (plusieurs instructions dans le corps de la boucle)

## G.8.3 Variables didactiques

Variables didactiques	Influence sur les procédures
Le tonneau possède une solidité aléatoire (entre 1 et 20) et le nombre de pots est également aléatoire (entre 1 et 10). Donne $20 \times 10 = 200$ parcours différents.	Augmente de coût de la stratégie consistant à coder un parcours particulier (S8.4) : $p = \frac{1}{20 \times 10} = \frac{1}{200} = 0,5\%$ Favorise l'utilisation d'une boucle while
Actions répétitives : donner plusieurs coups sur le tonneau, plusieurs coup/avancer pour détruire les pots	Favorise l'utilisation d'une boucle
Apparition de bâton de dynamite qui ne doit pas être détruit lorsque le tonneau est détruit.	Empêche d'assener un nombre de coups maximum sur le tonneau (20) quel que soit sa solidité Favorise l'utilisation d'une boucle while
Placement du coffre derrière le dernier pot	Empêche d'enchaîner les coup/avancer un nombre maximum de fois (10) quel que soit le nombre de pots. Favorise l'utilisation d'une boucle while

## G.8.4 Stratégies gagnantes

Description	Code
S8.1 (while-while) : <b>[Stratégie visée]</b>	1   avancer()

<p>Utiliser une boucle while afin de donner des coups de sabre tant qu'un obstacle (le tonneau) se trouve devant soi.  Avancer de trois blocs jusqu'à la clé.  Se retourner puis avancer de 5 blocs jusqu'au premier pot.  Utiliser une boucle while afin de porter un coup de sabre et d'avancer tant qu'un obstacle (un pot) se trouve devant nous.  Ouvrir le coffre.</p> <p><u>Concepts</u> :</p> <p>VAR-AF, WHI, VAR-AF, FOR-RE, FOR-RE, VAR-AF, WHI, VAR-AF</p>	<pre> 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 for _ in range(3) : 7     avancer() 8 droite() 9 for _ in range(5): 10    avancer() 11 obstacle = detecter_obstacle() 12 while obstacle == True : 13    coup() 14    avancer() 15    obstacle = detecter_obstacle() 16 ouvrir() </pre>
<p>S8.1 (while-while_variante) :</p> <p>Avancer vers la clé sans boucle for.</p> <p><u>Concepts</u> :</p> <p>VAR-AF, WHI, VAR-AF, FOR-RE, VAR-AF, WHI, VAR-AF</p>	<pre> 1 avancer() 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 avancer() 7 avancer() 8 avancer() 9 droite() 10 for _ in range(5): 11    avancer() 12 obstacle = detecter_obstacle() 13 while obstacle == True : 14    coup() 15    avancer() 16    obstacle = detecter_obstacle() 17 ouvrir() </pre>
<p>S8.1 (while-while_variante) :</p> <p>Avancer vers les pots sans boucle for.</p> <p><u>Concepts</u> :</p> <p>VAR-AF, WHI, VAR-AF, FOR-RE, VAR-AF, WHI, VAR-AF</p>	<pre> 1 avancer() 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 for _ in range(3): 7     avancer() 8 droite() 9 avancer() 10 avancer() 11 avancer() 12 avancer() 13 avancer() 14 obstacle = detecter_obstacle() 15 while obstacle == True : 16    coup() 17    avancer() 18    obstacle = detecter_obstacle() 19 ouvrir() </pre>

<p>S8.1 (while-while_variante) :  Avancer vers la clé et les pots sans boucle for.  <u>Concepts</u> :  VAR-AF, WHI, VAR-AF, FOR-RE, VAR-AF, WHI, VAR-AF</p>	<pre> 1 avancer() 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 avancer() 7 avancer() 8 avancer() 9 droite() 10 avancer() 11 avancer() 12 avancer() 13 avancer() 14 avancer() 15 obstacle = detecter_obstacle() 16 while obstacle == True : 17     coup() 18     avancer() 19     obstacle = detecter_obstacle() 20 ouvrir() </pre>
<p>S8.1 (while-while_variante) :  Faire l'appel de fonction directement dans la condition de la boucle.  <u>Concepts</u> :  WHI, FOR-RE, FOR-RE, WHI</p>	<pre> 1 avancer() 2 while detecter_obstacle() == True: 3     coup() 4 for _ in range(3): 5     avancer() 6 droite() 7 for _ in range(5): 8     avancer() 9 while detecter_obstacle() == True: 10    coup() 11    avancer() 12 ouvrir() </pre>
<p>S8.1 (while-while_variante) :  Avancer vers la clé sans boucle for.  <u>Concepts</u> :  WHI, FOR-RE, WHI</p>	<pre> 1 avancer() 2 while detecter_obstacle() == True: 3     coup() 4 avancer() 5 avancer() 6 avancer() 7 droite() 8 for _ in range(5): 9     avancer() 10 while detecter_obstacle() == True: 11    coup() 12    avancer() 13 ouvrir() </pre>
<p>S8.1 (while-while_variante) :  Avancer vers les pots sans boucle for.  <u>Concepts</u> :  WHI, FOR-RE, WHI</p>	<pre> 1 avancer() 2 while detecter_obstacle() == True: 3     coup() 4 for _ in range(3): 5     avancer() 6 droite() </pre>

	<pre> 7 avancer() 8 avancer() 9 avancer() 10 avancer() 11 avancer() 12 while detector_obstacle() == True: 13     coup() 14     avancer() 15 ouvrir() </pre>
<p>S8.1 (while-while_variante) :</p> <p>Avancer vers la clé et les pots sans boucle for.</p> <p><u>Concepts</u> :</p> <p>WHI, WHI</p>	<pre> 1 avancer() 2 while detector_obstacle() == True: 3     coup() 4 avancer() 5 avancer() 6 avancer() 7 droite() 8 avancer() 9 avancer() 10 avancer() 11 avancer() 12 avancer() 13 while detector_obstacle() == True: 14     coup() 15     avancer() 16 ouvrir() </pre>
<p>S8.2 (for-cond-for-cond) :</p> <p>Utiliser des boucles for ainsi que des conditionnelles pour remplacer les deux boucles whiles</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, FOR-RE, FOR-RE, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 for _ in range(21): 3     obstacle = detector_obstacle() 4     if obstacle == True: 5         coup() 6 for _ in range(3): 7     avancer() 8 droite() 9 for _ in range(5): 10    avancer() 11 for _ in range(11): 12    obstacle = detector_obstacle() 13    if obstacle == True: 14        coup() 15        avancer() 16    else: 17        avancer() 18 ouvrir() </pre>
<p>S8.2 (for-cond-for-cond_variante) :</p> <p>Avancer vers la clé sans boucle for.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, FOR-RE, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 for _ in range(15): 3     obstacle = detector_obstacle() 4     if obstacle == True: 5         coup() 6 avancer() 7 avancer() 8 avancer() </pre>

	<pre> 9 droite() 10 for _ in range(5): 11     avancer() 12 for _ in range(11): 13     obstacle = detecter_obstacle() 14     if obstacle == True: 15         coup() 16         avancer() 17     else: 18         avancer() 19         ouvrir() </pre>
<p>S8.2 (for-cond-for-cond_variante) :  Avancer vers les pots sans boucle for.  <u>Concepts</u> :  FOR-RE, VAR-AF, IF, FOR-RE, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 for _ in range(15): 3     obstacle = detecter_obstacle() 4     if obstacle == True: 5         coup() 6 for _ in range(3): 7     avancer() 8 droite() 9 avancer() 10 avancer() 11 avancer() 12 avancer() 13 avancer() 14 for _ in range(11): 15     obstacle = detecter_obstacle() 16     if obstacle == True: 17         coup() 18         avancer() 19     else: 20         avancer() 21         ouvrir() </pre>
<p>S8.2 (for-cond-for-cond_variante) :  Avancer vers la clé et les pots sans boucle for.  <u>Concepts</u> :  FOR-RE, VAR-AF, IF, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 for _ in range(15): 3     obstacle = detecter_obstacle() 4     if obstacle == True: 5         coup() 6 avancer() 7 avancer() 8 avancer() 9 droite() 10 avancer() 11 avancer() 12 avancer() 13 avancer() 14 avancer() 15 for _ in range(11): 16     obstacle = detecter_obstacle() 17     if obstacle == True: 18         coup() 19         avancer() </pre>

	20 21 22	<pre> else:     avancer()     ouvrir() </pre>
<p>S8.3 (for-cond-while) :</p> <p>Utiliser une boucle for ainsi qu'une conditionnelle pour remplacer la première boucle while</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, FOR-RE, FOR-RE, VAR-AF, WHI, VAR-AF</p>	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16	<pre> avancer() for _ in range(15):     obstacle = detecter_obstacle()     if obstacle == True:         coup() for _ in range(3):     avancer() droite() for _ in range(5):     avancer() obstacle = detecter_obstacle() while obstacle == True :     coup()     avancer() obstacle = detecter_obstacle() ouvrir() </pre>
<p>S8.3 (for-cond-while_variante) :</p> <p>Avancer vers la clé sans boucle for.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, FOR-RE, VAR-AF, WHI, VAR-AF</p>	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17	<pre> avancer() for _ in range(15):     obstacle = detecter_obstacle()     if obstacle == True:         coup() avancer() avancer() avancer() droite() for _ in range(5):     avancer() obstacle = detecter_obstacle() while obstacle == True :     coup()     avancer() obstacle = detecter_obstacle() ouvrir() </pre>
<p>S8.3 (for-cond-while_variante) :</p> <p>Avancer vers les pots sans boucle for.</p> <p><u>Concepts</u> :</p> <p>FOR-RE, VAR-AF, IF, FOR-RE, VAR-AF, WHI, VAR-AF</p>	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	<pre> avancer() for _ in range(15):     obstacle = detecter_obstacle()     if obstacle == True:         coup() for _ in range(3):     avancer() droite() avancer() avancer() avancer() avancer() avancer() obstacle = detecter_obstacle() while obstacle == True : </pre>

	<pre> 16     coup() 17     avancer() 18     obstacle = detecter_obstacle() 19 ouvrir() </pre>
<p>S8.3 (for-cond-while_variante) :  Avancer vers la clé et les pots sans boucle for.  <u>Concepts</u> :  FOR-RE, VAR-AF, IF, VAR-AF, WHI, VAR-AF</p>	<pre> 1 avancer() 2 for _ in range(15): 3     obstacle = detecter_obstacle() 4     if obstacle == True: 5         coup() 6 avancer() 7 avancer() 8 avancer() 9 droite() 10 avancer() 11 avancer() 12 avancer() 13 avancer() 14 avancer() 15 obstacle = detecter_obstacle() 16 while obstacle == True : 17     coup() 18     avancer() 19     obstacle = detecter_obstacle() 20 ouvrir() </pre>
<p>S8.3 (for-cond-while_variante) :  Utiliser une boucle for ainsi qu'une conditionnelle pour remplacer la deuxième boucle while.  <u>Concepts</u> :  VAR-AF, WHI, VAR-AF, FOR-RE, FOR-RE, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 for _ in range(3): 7     avancer() 8 droite() 9 for _ in range(5): 10    avancer() 11 for _ in range(11): 12    obstacle = detecter_obstacle() 13    if obstacle == True: 14        coup() 15        avancer() 16    else: 17        avancer() 18 ouvrir() </pre>
<p>S8.3 (for-cond-while_variante) :  Avancer vers la clé sans boucle for.  <u>Concepts</u> :  VAR-AF, WHI, VAR-AF, FOR-RE, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 avancer() 7 avancer() 8 avancer() 9 droite() </pre>



	<pre> 10 for _ in range(5): 11     avancer() 12 for _ in range(11): 13     obstacle = detecter_obstacle() 14     if obstacle == True: 15         coup() 16         avancer() 17     else: 18         avancer() 19         ouvrir() </pre>
<p>S8.3 (for-cond-while_variante) :  Avancer vers les pots sans boucle for.  <u>Concepts</u> :  VAR-AF, WHI, VAR-AF, FOR-RE, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 for _ in range(3): 7     avancer() 8 droite() 9 avancer() 10 avancer() 11 avancer() 12 avancer() 13 avancer() 14 for _ in range(11): 15     obstacle = detecter_obstacle() 16     if obstacle == True: 17         coup() 18         avancer() 19     else: 20         avancer() 21         ouvrir() </pre>
<p>S8.3 (for-cond-while_variante) :  Avancer vers la clé et les pots sans boucle for.  <u>Concepts</u> :  VAR-AF, WHI, VAR-AF, FOR-RE, VAR-AF, IF, ELSE</p>	<pre> 1 avancer() 2 obstacle = detecter_obstacle() 3 while obstacle == True : 4     coup() 5     obstacle = detecter_obstacle() 6 avancer() 7 avancer() 8 avancer() 9 droite() 10 avancer() 11 avancer() 12 avancer() 13 avancer() 14 avancer() 15 for _ in range(11): 16     obstacle = detecter_obstacle() 17     if obstacle == True: 18         coup() 19         avancer() 20     else: </pre>

	21	<code>avancer()</code>
	22	<code>ouvrir()</code>
S8.4 (no-while-no-if): Essayer un parcours en particulier <u>Concepts</u> : ∅	1	<code>avancer()</code>
	2	<code>coup()</code>
	3	<code>coup()</code>
	4	<code>avancer()</code>
	5	<code>avancer()</code>
	6	<code>avancer()</code>
	7	<code>droite()</code>
	8	<code>avancer()</code>
	9	<code>avancer()</code>
	10	<code>avancer()</code>
	11	<code>avancer()</code>
	12	<code>avancer()</code>
	13	<code>coup()</code>
	14	<code>avancer()</code>
	15	<code>coup()</code>
	16	<code>avancer()</code>
	17	<code>ouvrir()</code>
S8.4 (no-while-no-if_variante): Parcours particulier avec une boucle for <u>Concepts</u> : FOR-RE	1	<code>avancer()</code>
	2	<code>for _ in range(2):</code>
	3	<code>    coup()</code>
	4	<code>avancer()</code>
	5	<code>avancer()</code>
	6	<code>avancer()</code>
	7	<code>droite()</code>
	8	<code>avancer()</code>
	9	<code>avancer()</code>
	10	<code>avancer()</code>
	11	<code>avancer()</code>
	12	<code>avancer()</code>
	13	<code>coup()</code>
	14	<code>avancer()</code>
	15	<code>coup()</code>
	16	<code>avancer()</code>
	17	<code>ouvrir()</code>
S8.4 (no-while-no-if_variante): Parcours particulier avec deux boucles for <u>Concepts</u> : FOR-RE, FOR-RE	1	<code>avancer()</code>
	2	<code>for _ in range(2):</code>
	3	<code>    coup()</code>
	4	<code>for _ in range(3):</code>
	5	<code>    avancer()</code>
	6	<code>droite()</code>
	7	<code>avancer()</code>
	8	<code>avancer()</code>
	9	<code>avancer()</code>
	10	<code>avancer()</code>
	11	<code>avancer()</code>
	12	<code>coup()</code>
	13	<code>avancer()</code>
	14	<code>coup()</code>
	15	<code>avancer()</code>

<p>S8.4 (no-while-no-if_variante):  Parcours particulier avec trois boucles  for  <u>Concepts</u> :  FOR-RE, FOR-RE, FOR-RE</p>	<pre> 16 ouvrir() 1  avancer() 2  for _ in range(2): 3      coup() 4  for _ in range(3): 5      avancer() 6  droite() 7  for _ in range(5): 8      avancer() 9  coup() 10 avancer() 11 coup() 12 avancer() 13 ouvrir() </pre>
<p>S8.4 (no-while-no-if_variante):  Parcours particulier avec quatre boucles  for  <u>Concepts</u> :  FOR-RE, FOR-RE, FOR-RE, FOR-RE</p>	<pre> 1  avancer() 2  for _ in range(2): 3      coup() 4  for _ in range(3): 5      avancer() 6  droite() 7  for _ in range(5): 8      avancer() 9  for _ in range(2): 10     coup() 11     avancer() 12 ouvrir() </pre>

# H Guide de démarrage

## Guide de démarrage

### Objectif du jeu




- Ce jeu met en scène les aventures d'un ou d'une **pirate** que vous avez choisi.
- **Chaque niveau** du jeu doit lui permettre **d'ouvrir un coffre** et d'en récupérer les pièces d'or 🏴‍☠️
- Le personnage se **contrôle à l'aide d'un programme informatique** qu'il faut écrire en 🐍python

### Fonctionnement

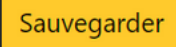
L'écran est divisé en plusieurs zones:

	C : Jeu	
A : Niveaux	D : Contrôle	E : Console
B : Mémo	F : Programmation	

- Zone **A** : description des **objectifs**, des **contraintes** et des **fonctions de contrôle** du personnage pour le niveau en cours. Cette zone permet également de naviguer parmi les niveaux terminés.

- Zone **B** : **mémo de programmation** expliquant **comment programmer** en  python. Pour les utilisateurs de , nous donnons quelques comparaisons qui peuvent aider.
- Zone **C** : **interface graphique** du jeu.
- Zone **D** : **panneau de contrôle** du programme qui permettent de : **lancer l'exécution**, de **forcer l'arrêt** et de **contrôler la vitesse d'exécution**.
- Zone **E** : **console** d'affichage des **sorties du programme** au cours de son exécution. Cette zone affiche : les informations de **début et de fin d'exécution**, et les éventuelles **erreurs de programmation**.
- Zone **F** : **éditeur de programme**, c'est à cet endroit qu'il faut écrire le **code de son programme**  python qui va permettre de contrôler le personnage.

### Enregistrement

- Le jeu s'enregistre **manuellement** lorsque l'on **appuie sur le bouton** 
- Le jeu s'enregistre également **automatiquement** au début de **chaque niveau**.
- Il faut cependant penser à **enregistrer avant de quitter** (fermeture de l'onglet ou du navigateur Internet).
- L'**identifiant de la partie** est nécessaire pour **reprendre le jeu** après l'avoir quitté.

# I Mémo Python

## I.1 Notions de base

### Notions de base

#### Utilité

Quelques **notions essentielles** qui permettent de **débuter la programmation en Python**.

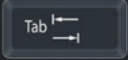
#### Programme

- Un programme est un **texte** qui permet de **commander** un ordinateur. Dans notre cas, il s'agit de contrôler un personnage.
- Ce texte est composé **d'instructions** compréhensibles par l'ordinateur. Certaines sont **spécifiques à ce jeu** et permettent de contrôler le personnage (instruction **avancer()** par exemple), d'autres sont **communes à tous les programmes** en Python (voir la suite du mémo programmation).
- Une fois rédigé, le programme est **exécuté** par un **interpréteur**. Dans notre cas, l'interpréteur Python est intégré dans le site web et est lancé lorsque l'on clique sur le bouton **Exécuter**

#### Erreurs

- Les instructions d'un programme doivent être **très précises** et ne comporter **aucune erreur**, attention aux fautes de frappe !
- En cas d'erreur, l'interpréteur **signale un problème dans la console**. Par exemple, la ligne de code erronée **avance()** provoque l'affichage d'une erreur : **> Ligne 1 : le nom 'avance()' n'est pas défini**
- Il faudra donc **modifier le programme** à partir de ce **message d'erreur**, puis relancer l'exécution jusqu'à ce que le programme soit correct.

## Structuration

- Un programme doit être **structuré** par des **décalages de texte** lors de l'utilisation de certaines instructions (`if`, `while`, `for`, etc.). On appelle cela l'**indentation** du code.
- Si l'indentation est **mauvaise**, le programme **ne réalise pas ce que l'on veut** et cela peut même **provoquer des erreurs**.
- Ces décalages s'effectuent avec la **touche tabulation** du clavier :  A noter que **l'éditeur de programme ajoute automatiquement des décalages** lors de l'utilisation de certaines instructions.

## Commentaires

- Les lignes commençant par `#` ne sont **pas prises en compte** par l'interpréteur. Par exemple, la ligne de code `# Ceci est un commentaire` n'a aucun effet.
- Ces lignes permettent **de donner des explications** dans le programme, on les appelle des **commentaires**.
- Les commentaires ne sont **pas obligatoires** mais ils peuvent **aider à la compréhension** des programmes par les autres programmeurs.

## I.2 Variable

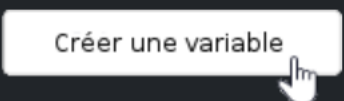
### Variable

#### Utilité

Permet de **garder en mémoire** des informations au cours de l'exécution d'un programme.

#### Création

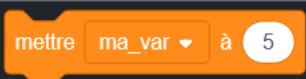

Donne un **nom** à un **espace mémoire** qui pourra par la suite **contenir des valeurs**.

Scratch	python
<p>Exemple</p> 	<p>Exemple</p> <pre>ma_variable = 0</pre> <p>Copier</p>

- Le nom de la variable est **choisi par le programmeur**, il doit être **unique dans le programme** et ne doit contenir **ni espace ni accent**.
- En Python, une variable doit **obligatoirement être initialisée** à l'aide de l'opérateur = lors de sa création (par exemple initialisée à 0 comme ci-dessus).

#### Modification

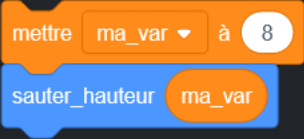
**Change la valeur** contenue dans l'**espace mémoire**.

Scratch	python
<p>Exemple</p> 	<p>Exemple</p> <pre>ma_var = 5</pre> <p>Copier</p>
<p>Exemple</p> 	<p>Exemple</p> <pre>ma_var = lire_nombre()</pre> <p>Copier</p>



## Utilisation

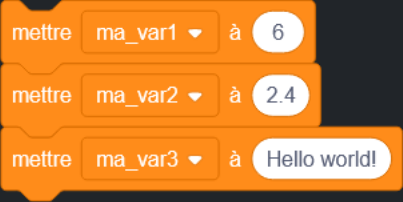
**Met à disposition** d'une autre instruction la valeur contenue dans **l'espace mémoire**.

Scratch	python
<p>Exemple</p> 	<p>Exemple</p> <pre>ma_var = 8 sauter_hauteur(ma_var)</pre> <p><small>Copier</small></p>

## Type

Une variable peut avoir différents **types** selon la **nature des valeurs** qu'elle contient. En Python, on trouve (entre autres) :

- des **nombres entiers**
- des **nombres décimaux**
- des **chaînes de caractères** (texte)
- des **booléens** (vrai ou faux).

Scratch	python
<p>Exemple</p> 	<p>Exemple</p> <pre>ma_var1 = 6 ma_var2 = 2.4 ma_var3 = "Hello world!" ma_var4 = True</pre> <p><small>Copier</small></p>

- Le **séparateur des nombres décimaux** est un **point**.
- Les **chaînes de caractères** doivent être entourées par des **guillemets ""**.
- Les variables de **type booléen** (qui n'existent pas en Scratch) peuvent prendre **uniquement** les valeurs **True** (vrai) ou **False** (faux).

## I.3 Conditionnelle



### Conditionnelle


#### Utilité

Permet d'**exécuter des instructions** lorsque **certaines conditions sont respectées**.

#### Conditionnelle à une branche

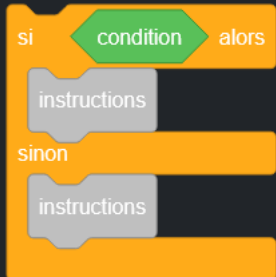
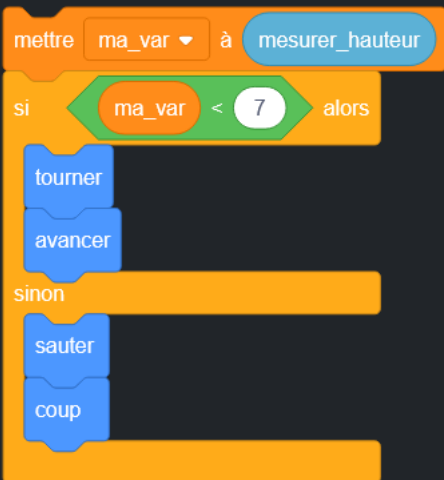
Permet d'**exécuter des instructions** si une **condition est vraie**.

SCRATCH	python
<b>Modèle</b> 	<b>Modèle</b> <pre>if condition:     instructions</pre> <span>Copier</span>
<b>Exemple</b> 	<b>Exemple</b> <pre>ma_var = mesurer_hauteur() if ma_var == 3:     coup()     sauter()</pre> <span>Copier</span>

-  pour **tester une égalité** on utilise un **double égal ==**.
- Les **instructions dans la branche if** (corps) doivent être **décalées** à l'aide de la touche tabulation.

## Conditionnelle à deux branches

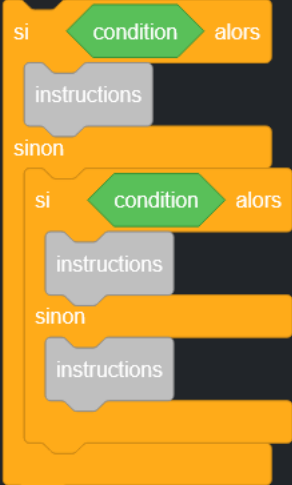

Permet d'**exécuter des instructions** si une **condition est vraie** et **d'autres instructions sinon** (si la condition est fausse).

Scratch	python
<p>Modèle</p>  <p>The Scratch 'if-then-else' block consists of a yellow 'si' block with a green 'condition' block and an 'alors' block. Below the 'alors' block are two grey 'instructions' blocks. A yellow 'sinon' block is attached to the bottom of the 'si' block, containing two grey 'instructions' blocks.</p>	<p>Modèle</p> <pre data-bbox="783 501 1066 680">if condition:     instructions else:     instructions</pre> <p>Copier</p>
<p>Exemple</p>  <p>The Scratch example code starts with an orange 'mettre' block setting 'ma_var' to 'mesurer_hauteur'. This is followed by a yellow 'si' block with a green 'ma_var &lt; 7' condition. The 'alors' branch contains blue 'tourner' and 'avancer' blocks. The 'sinon' branch contains blue 'sauter' and 'coup' blocks.</p>	<p>Exemple</p> <pre data-bbox="783 860 1241 1189">ma_var = mesurer_hauteur() if ma_var &lt; 7:     tourner()     avancer() else:     sauter()     coup()</pre> <p>Copier</p>

- Les **instructions dans les branches if et else** (corps) doivent être **décalées** à l'aide de la touche tabulation.

## Conditionnelle à trois branches et plus

Permet d'exécuter des instructions différentes en fonction de différentes conditions.

Scratch	python
<p>Modèle</p> 	<p>Modèle</p> <pre data-bbox="783 517 1066 786">if condition:     instructions elif condition:     instructions else:     instructions</pre> <p>Copier</p>
<p>Exemple</p> 	<p>Exemple</p> <pre data-bbox="783 1084 1241 1451">ma_var = mesurer_hauteur() if ma_var == 5:     tourner() elif ma_var == 7:     avancer()     sauter() else:     coup()</pre> <p>Copier</p>

- Il est possible d'ajouter autant de branches **elif** que l'on souhaite.
- La branche **else** n'est pas obligatoire, une conditionnelle peut n'avoir que des branches **if** et **elif**.
- Les **instructions dans les branches if, elif et else** (corps) doivent être **décalées** à l'aide de la touche tabulation.

## I.4 Boucle for

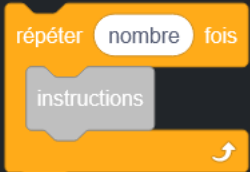

### Boucle for

#### Utilité

Permet de **répéter** des instructions **un certain nombre de fois**.

#### Répétition simple



Permet de **répéter** des instructions **un certain nombre de fois**.

Scratch	python
<p>Modèle</p> 	<pre>for _ in range(nombre):     instructions</pre> <p>Copier</p>
<p>Exemple</p> 	<pre>for _ in range(4):     sauter()     coup()</pre> <p>Copier</p>

- Le nombre entre parenthèses dans `range(nombre)` indique le **nombre de répétitions** des instructions.
- Les **instructions répétées dans la boucle** (corps) doivent être **décalées** à l'aide de la touche tabulation.

## Répétition avec compteur (commence à zéro)



Permet de **répéter** des instructions **un certain nombre de fois** tout en **mettant à jour automatiquement** une variable **compteur** qui est **initialisée à zéro**.


Scratch	python
<p>Modèle</p> 	<p>Modèle</p> <pre>for compteur in range(nombre):     instructions</pre> <p><small>Copier</small></p>
<p>Exemple</p> 	<p>Exemple</p> <pre>for compteur in range(3):     tirer(compteur)     avancer()</pre> <p><small>Copier</small></p>

- Le nombre entre parenthèses dans `range(nombre)` indique le **nombre de répétitions** des instructions.
- La variable `compteur` est **automatiquement initialisée à 0** et **automatiquement augmentée de 1** à la fin de chaque tour de boucle (sauf au dernier).
- La variable `compteur` peut **être utilisée** dans les instructions répétées dans la boucle.

## Répétition avec compteur (ne commence pas à zéro)

Permet de **répéter** des instructions **un certain nombre de fois** tout en **mettant à jour automatiquement** une variable **compteur**.

Scratch	python
<p>Modèle</p> 	<p>Modèle</p> <pre>for compteur in range(n1, n2):     instructions</pre> <p><small>Copier</small></p>
<p>Exemple</p> 	<p>Exemple</p> <pre>for compteur in range(3, 7):     sauter_hauteur(compteur)     coup()</pre> <p><small>Copier</small></p>

- La variable `compteur` est **automatiquement initialisée** à `n1` (3 dans l'exemple) et **automatiquement augmenté de 1** à la fin de chaque tour de boucle (sauf au dernier).
-  les **répétitions s'arrêtent** lorsque le compteur atteint `n2-1` (6 dans l'exemple: `7-1`).

## I.5 Boucle while

### Boucle while

**Utilité**

Permet de **répéter** des instructions **tant que certaines conditions sont respectées.**

**Répétition tant que ...**

Permet de **répéter** des instructions **tant qu'une condition est vraie.**

Scratch	python
<p><b>Modèle</b></p> 	<pre>while condition: instructions</pre> <p>Copier</p>
<p><b>Exemple</b></p> 	<pre>h = mesurer_hauteur() while h &gt; 0: sauter() h = mesurer_hauteur()</pre> <p>Copier</p>

- La boucle **continue de tourner tant que la condition est vraie** et elle **s'arrête dès qu'elle devient fausse.**
- ⚠ en Python la boucle **tourne** tant que la **condition est vraie**, en Scratch la boucle **s'arrête** quand la **condition est vraie.**
- Les **instructions répétées dans la boucle** (corps) doivent être **décalées** à l'aide de la touche tabulation.



# J Courriels

## J.1 Invitation participation enquête enseignants

Chères et chers collègues,

Dans le cadre d'une thèse en didactique de l'informatique effectuée à l'université de Bretagne Occidentale, je mène une **étude sur la programmation informatique dans l'enseignement secondaire**.

Dans le but d'établir un **état des lieux des pratiques enseignantes lors de la transition du collège au lycée**, je vous sollicite pour **répondre à un questionnaire en ligne**.

Ce questionnaire s'adresse aux **professeurs**:

- de **mathématiques** au niveau **3ème**
- de **technologie** au niveau **3ème**
- de **mathématiques** au niveau **2nd**
- de **SNT** au niveau **2nd**

Il prend environ **15 minutes**, est strictement **anonyme** et ne recueille **aucune donnée personnelle**.

Pour participer, cliquer sur le lien ci-dessous:

<http://enquetes.espe-bretagne.fr/index.php?r=survey/index&sid=235875&lang=fr>

Vous remerciant de votre participation,

Matthieu BRANTHÔME

Courriel d'invitation à participer à l'enquête en ligne concernant les pratiques d'enseignement, envoyé le 07/05/2021

## J.2 Invitation participation expérimentation

Bonsoir,

Dans le cadre de ma thèse, vous aviez répondu au mois de mai dernier à un questionnaire en ligne sur la programmation dans l'enseignement secondaire. Suite à quoi, vous m'aviez contacté afin d'en obtenir les résultats.

Je ne suis pas encore en mesure de vous communiquer cette synthèse. Cependant, j'ai développé, en parallèle de ce travail, une application conçue pour accompagner la transition collège-lycée dans l'apprentissage de la programmation.

L'application s'appelle Pyrates (<https://py-rates.fr>) et se présente sous la forme d'un jeu de plateforme avec un personnage dirigeable en Python. Vous trouverez en pièce jointe le guide pédagogique vous permettant de la découvrir. N'hésitez pas également à la tester par vous-même pour vous en faire une idée.

Cette application est actuellement en phase de test dans les classes. Je suis donc à la recherche d'enseignants volontaires susceptibles de l'essayer avec leurs élèves. Cette expérimentation impose cependant quelques contraintes:

- l'établissement doit être situé en Bretagne;
- les élèves doivent être au niveau seconde en classe de SNT ou de maths;
- les élèves ne doivent pas avoir été exposés, plus tôt dans l'année, à la programmation en Python.

Si vous rentrez dans ces critères, et si cette expérimentation vous intéresse, elle pourrait se dérouler de la sorte:

- sur deux séances de cours dans une salle équipée d'ordinateurs, d'un navigateur (Firefox ou Chrome idéalement) et d'une connexion Internet;
- l'application dispose d'un mode particulier destiné à la recherche dans lequel toutes les actions des élèves et les aides apportées par l'enseignant sont enregistrées dans une base de données dédiée;
- l'expérimentation ne nécessite aucun dispositif de captation particulier (ni micro, ni caméra etc.), seul un navigateur Internet est requis;
- aucune autorisation parentale n'est nécessaire, les données étant anonymes;
- le seul impératif étant pour l'enseignant de renseigner, par l'intermédiaire de clics sur des boutons, les types d'aides qu'il apporte aux élèves suite à ses interventions;
- je serai présent lors de l'expérimentation afin de présenter rapidement l'application aux élèves et pour observer leur activité.

Dites-moi si cette proposition pourrait vous intéresser. Je reste à votre disposition pour toutes informations complémentaires.

Bien cordialement,

Matthieu BRANTHÔME

Courriel d'invitation à participer à l'évaluation de l'application Python, envoyé le 11/09/2021

### J.3 Courriel de diffusion

Bonjour,

Dans le cadre d'une thèse en didactique de l'informatique, j'ai développé l'application Pyrates (<https://py-rates.fr>).

Il s'agit d'un site Internet accessible via un navigateur web et qui permet une première approche de la programmation en Python pour les élèves de seconde sous la forme d'un jeu de plateforme sur le thème des pirates. L'application est simple d'usage (page unique) et ne nécessite aucune inscription (conforme au RGPD).

Vous trouverez de plus amples informations (présentation, conseils de mise en œuvre et solutions) dans le guide pédagogique disponible à cette adresse: <https://py-rates.fr/guide/FR/index.html>

N'hésitez pas à me contacter pour toutes informations ou remarques,

Bien cordialement,

Matthieu BRANTHÔME  
Doctorant en informatique / sciences de l'éducation  
Université de Bretagne Occidentale

Courriel de diffusion de l'application Pyrates envoyé sur les listes RENATER de mathématiques, SNT et NSI le 12/10/21


# K Certificat de dépôt de code

**Inter Deposit Digital Number**  
Certificat délivré par  
**Agence pour la Protection des Programmes**  
25 rue de la Plaine - 75020 PARIS - FRANCE / SIREN 385 385 844 / T. +33(0)1 40 35 03 03 / app@app.asso.fr

**IDDN.FR.001.050001.000.S.P.2022.000.10000**  
(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

Pour l'œuvre : **Pyrates version 1.0 en date du 15 novembre 2021**

Identité du titulaire de droits\* :  
**UBO - UNIVERSITÉ DE BRETAGNE  
OCCIDENTALE  
3 rue des Archives  
CS 93837  
29238 BREST CEDEX  
FRANCE  
Siren : 192903466**



Membre sous le numéro : **08.29.12136**  
Support utilisé : 1 clé USB en double exemplaire

Logibox conservée par l'adhérent : 127927  
Logibox conservée par l'APP : 127928

Fait à Paris, le 24/01/2022

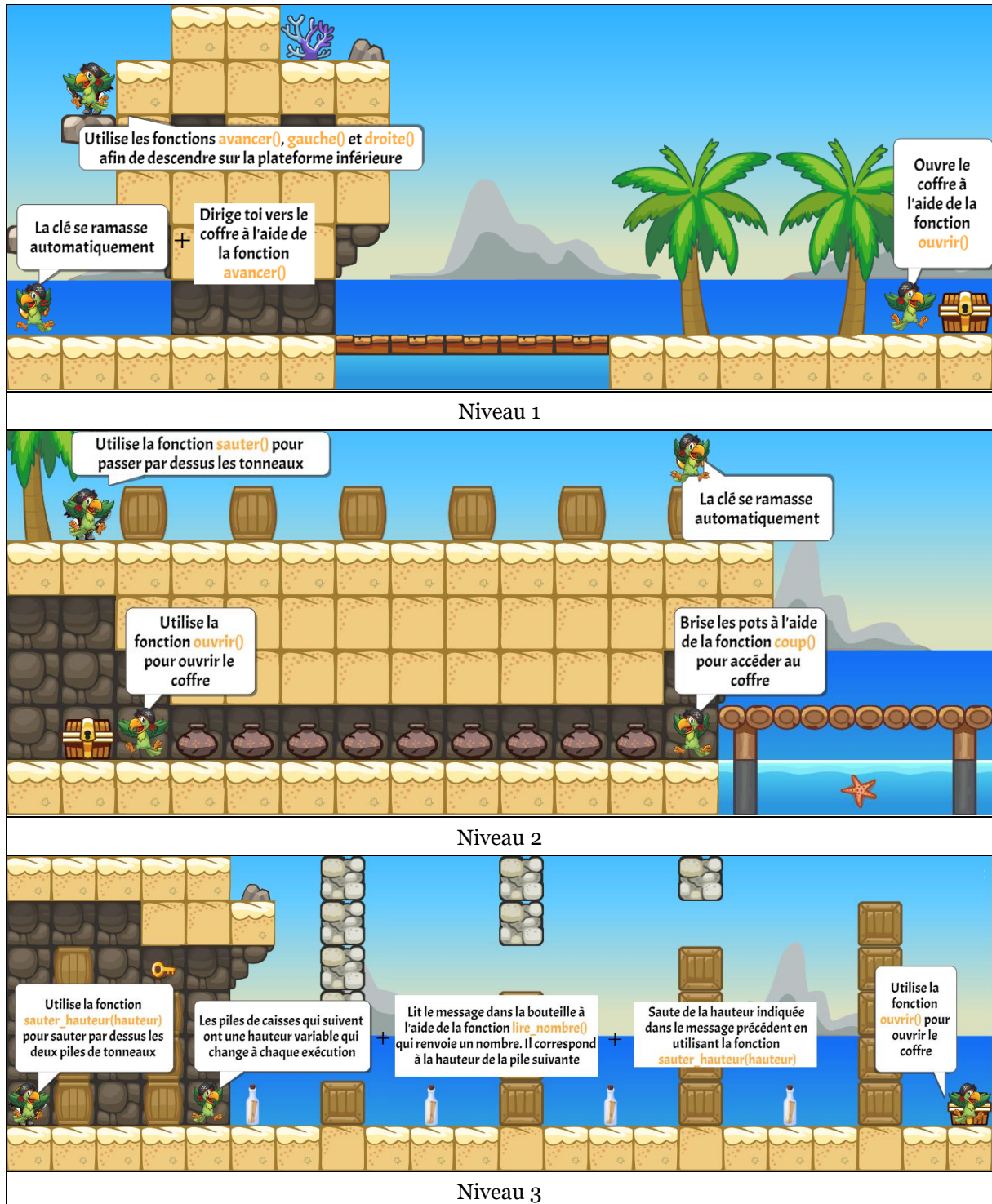
**APP** AGENCE POUR LA PROTECTION DES PROGRAMMES  
19, rue de la Plaine - 75020 Paris - France  
T. +33(0)1 40 35 03 03 / app@app.asso.fr  
SIREN 385 385 844  
app.asso.fr

(1) Inter Deposit Digital Number (3) Numéro de l'organisme d'enregistrement (5) Numéro de version (7) Type de l'œuvre  
(2) Nationalité de l'œuvre (4) Numéro d'ordre de l'enregistrement (6) Type d'enregistrement (8) Année d'enregistrement

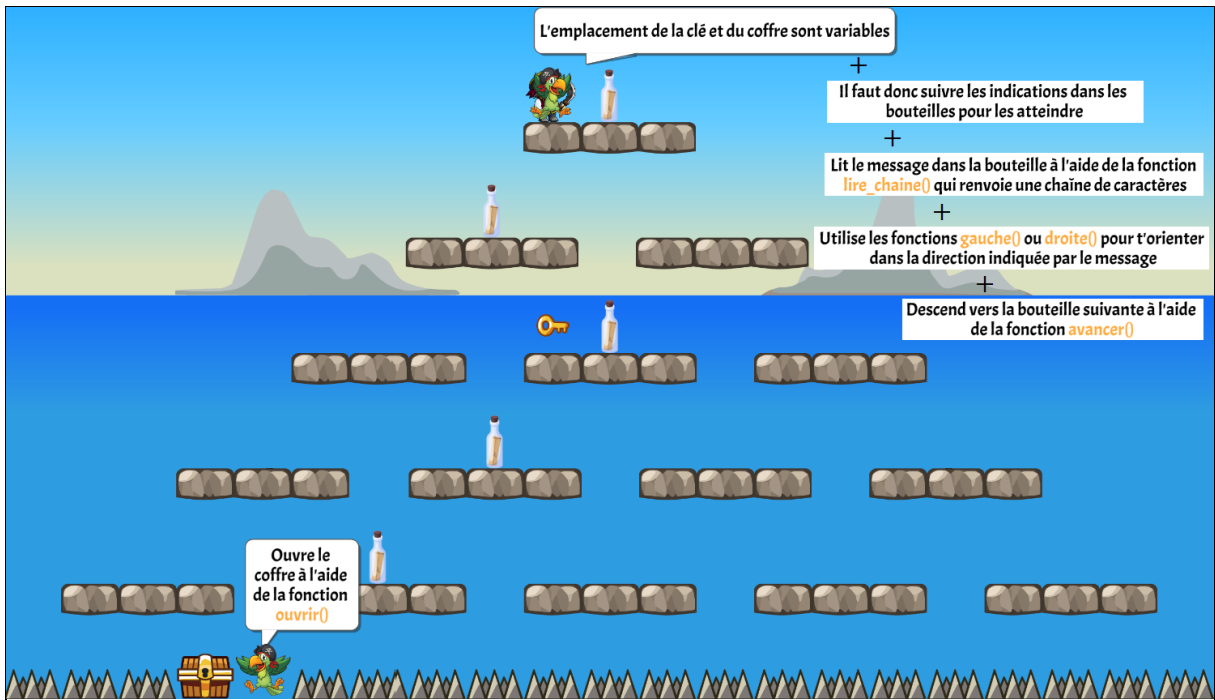
*\* Le titulaire s'engage à informer l'APP de toute cession ou aliénation, totale ou partielle, de ses droits de propriété intellectuelle.  
Seules les inscriptions de type S et C permettent un éventuel accès au programme source.*

# L Rétroactions

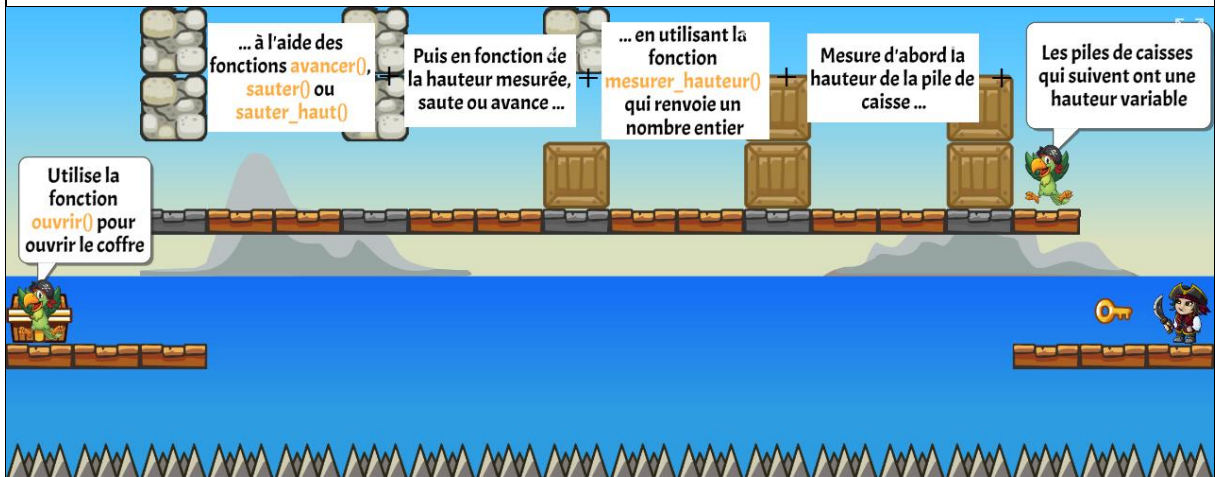
## L.1 Tâche







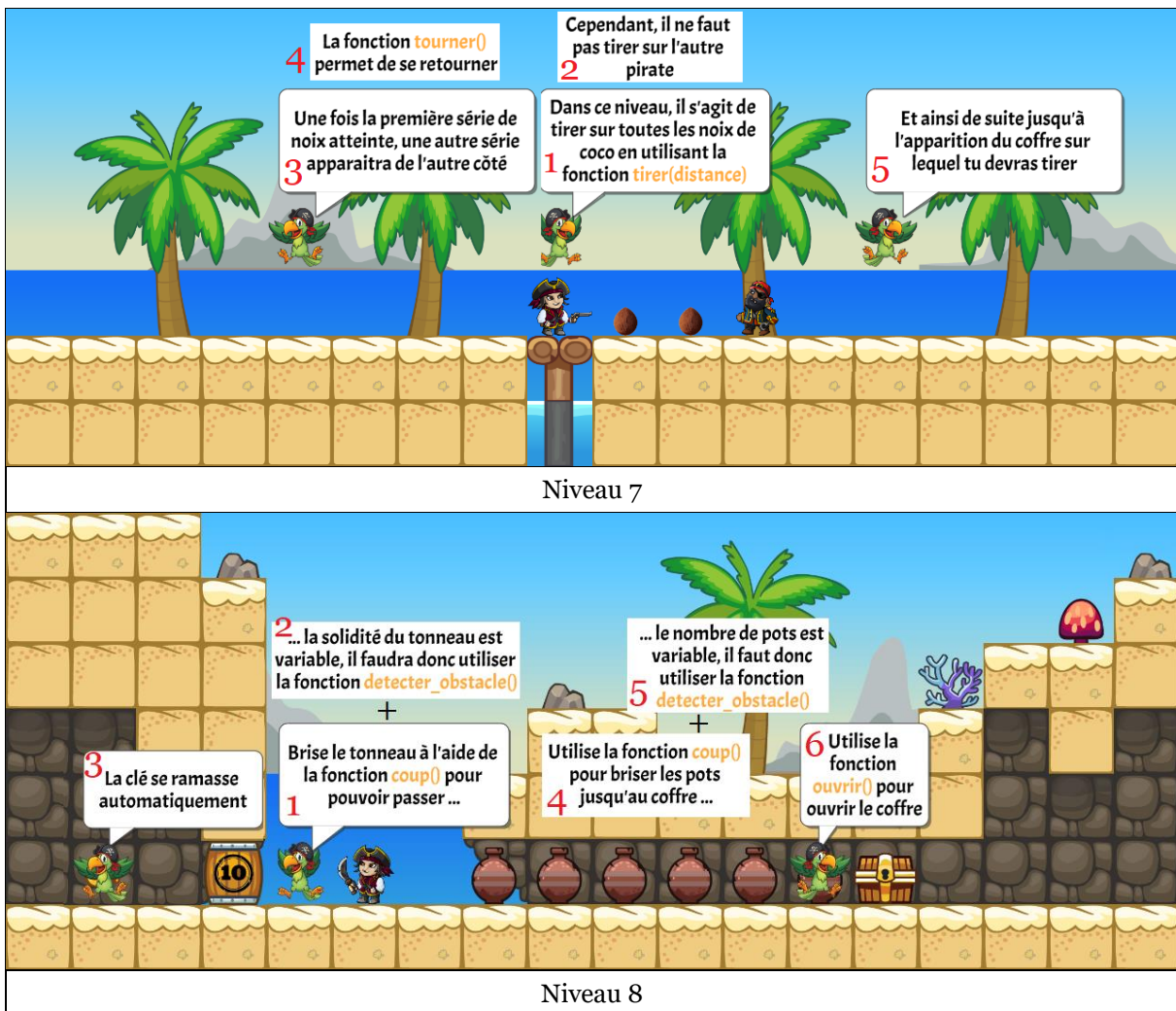
#### Niveau 4



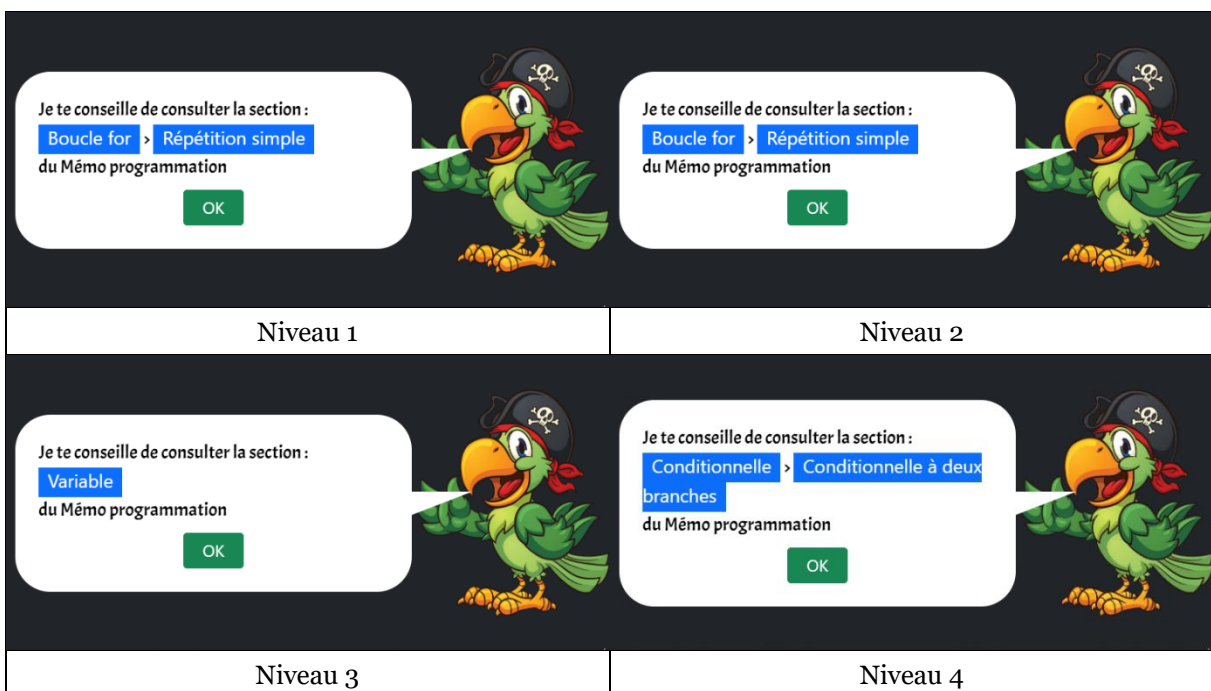
#### Niveau 5







#### Niveau 6













## L.2 Rétroactions niveau concept



<p>Je te conseille de consulter la section :</p> <p><a href="#">Conditionnelle</a> &gt; <a href="#">Conditionnelle à trois branches et plus</a> du Mémorandum de programmation</p> <p>OK</p>		<p>Je te conseille de consulter la section :</p> <p><a href="#">Boucle for</a> &gt; <a href="#">Répétition avec compteur (commence à zéro)</a> du Mémorandum de programmation</p> <p>OK</p>	
Niveau 5		Niveau 6	
<p>Je te conseille de consulter la section :</p> <p><a href="#">Boucle for</a> &gt; <a href="#">Répétition avec compteur (ne commence pas à zéro)</a> du Mémorandum de programmation</p> <p>OK</p>		<p>Je te conseille de consulter la section :</p> <p><a href="#">Boucle while</a> du Mémorandum de programmation</p> <p>OK</p>	
Niveau 7		Niveau 8	

### L.3 Rétroaction niveau implémentation

<p>Voici le code te permettant d'avancer 16 fois jusqu'au coffre :</p> <pre>for _ in range(16):     avancer()</pre> <p>Copier</p> <p>OK</p>		<p>Voici le code te permettant de sauter les 6 tonneaux :</p> <pre>for _ in range(6):     sauter()     avancer()</pre> <p>Copier</p> <p>Voici le code te permettant de casser les 9 pots :</p> <p>OK</p>	
Niveau 1		Niveau 2-1	
<p><code>sauter()</code> <code>avancer()</code></p> <p>Voici le code te permettant de casser les 9 pots :</p> <pre>for _ in range(9):     coup()     avancer()</pre> <p>Copier</p> <p>OK</p>		<p>Voici le code te permettant de lire le message dans une bouteille puis de sauter à la hauteur indiquée :</p> <pre>message = lire_nombre() avancer() sauter_hauteur(message)</pre> <p>Copier</p> <p>OK</p>	
Niveau 2-2		Niveau 3	
<p>Voici le code te permettant de lire le message dans une bouteille puis de s'orienter dans la direction indiquée :</p> <pre>message = lire_chaine() if message == "gauche":     gauche() else:</pre> <p>Copier</p> <p>OK</p>		<p>Voici le code te permettant de mesurer la hauteur des piles de caisses puis d'avancer ou de sauter :</p> <pre>hauteur = mesurer_hauteur() if hauteur == 0:     avancer() elif hauteur == 1:</pre> <p>Copier</p> <p>OK</p>	
Niveau 4		Niveau 5	

<p>Voici le code te permettant de sauter par dessus tous les piliers :</p> <pre>for compteur in range(6):     sauter_hauteur(compteur)     avancer()</pre> <p>OK</p>		<p>Voici le code te permettant de tirer sur toutes les noix de coco :</p> <pre>for compteur in range(2,9):     tirer(compteur)     tourner()</pre> <p>OK</p>	
Niveau 6		Niveau 7	
<p>Voici le code te permettant de casser le tonneau :</p> <pre>obstacle = detecter_obstacle() while obstacle == True:     coup()     obstacle = detecter_obstacle()</pre> <p>OK</p>		<p>Voici le code te permettant de casser les pots :</p> <pre>obstacle = detecter_obstacle() while obstacle == True:     coup()     avancer()     obstacle = detecter_obstacle()</pre> <p>OK</p>	
Niveau 8-1		Niveau 8-2	

## L.4 Rétroaction niveau solution

<p>Voici le code te permettant de terminer le niveau :</p> <pre>avancer() droite() avancer() gauche() avancer()</pre> <p>OK</p>		<p>Voici le code te permettant de terminer le niveau :</p> <pre>for _ in range(6):     sauter()     avancer() avancer() gauche()</pre> <p>OK</p>	
Niveau 1		Niveau 2	
<p>Voici le code te permettant de terminer le niveau :</p> <pre>sauter_hauteur(4) avancer() sauter_hauteur(3) avancer() avancer()</pre> <p>OK</p>		<p>Voici le code te permettant de terminer le niveau :</p> <pre>avancer() for _ in range(5):     message = lire_chaine()     if message == "gau":         gauche()</pre> <p>OK</p>	
Niveau 3		Niveau 4	
<p>Voici le code te permettant de terminer le niveau :</p> <pre>avancer() sauter_haut() for _ in range(5):     hauteur = mesurer_hauteur()     if hauteur == 0:</pre> <p>OK</p>		<p>Voici le code te permettant de terminer le niveau :</p> <pre>for compteur in range(6):     sauter_hauteur(compteur)     avancer()     ouvrir()</pre> <p>OK</p>	
Niveau 5		Niveau 6	



Voici le code te permettant de terminer le niveau:

```
for compteur in range(2,9):  
    tirer(compteur)  
    tourner()  
tirer(3)
```

OK



Niveau 7

Voici le code te permettant de terminer le niveau:

```
avancer()  
obstacle = detecter_obstacle()  
while obstacle == True:  
    coup()  
    obstacle = detecter_obstacl
```

OK



Niveau 8

# M Apprentissage automatique

## M.1 Détail des caractéristiques

Programme lancé : too-many-lines, syntactic-error, semantic-error, game-error, user-stopped, level-lost, level-completed, fully-executed.

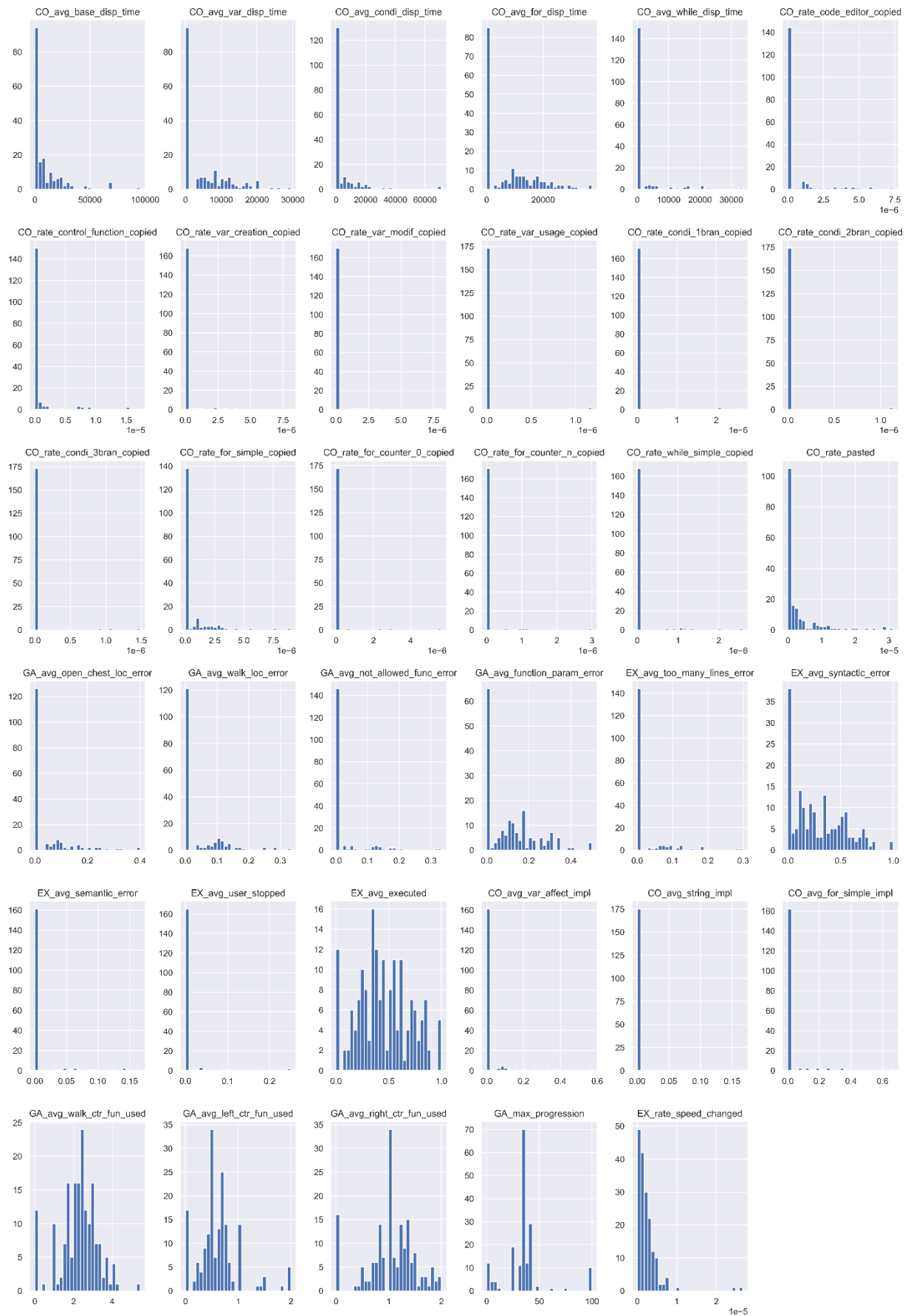
Programme exécuté : fully-executed, level-completed

Description	Mode de calcul	Code
Progression dans le niveau	Maximum depuis de début du niveau : $\max(\textit{progression})$	GA_max_progression
Fonctions de contrôle utilisées	Nombre moyen par programme exécuté : $\frac{\textit{Nb fct. contr. utilisés}}{\textit{Nb programmes exécutés}}$	GA_avg_walk_ctr_fun_used
		GA_avg_left_ctr_fun_used
		GA_avg_right_ctr_fun_used
		GA_avg_open_ctr_fun_used
		GA_avg_jump_ctr_fun_used
		GA_avg_jump_height_ctr_fun_used
		GA_avg_jump_high_ctr_fun_used
		GA_avg_get_height_ctr_fun_used
		GA_avg_read_string_ctr_fun_used
		GA_avg_read_int_ctr_fun_used
		GA_avg_attack_ctr_fun_used
		GA_avg_detect_obstacle_ctr_fun_used
		GA_avg_turn_ctr_fun_used
GA_avg_shoot_ctr_fun_used		
Erreurs dans le jeu	Nombre moyen par programme exécuté : $\frac{\textit{Nb erreurs}}{\textit{Nb programmes lancés}}$	GA_avg_open_chest_loc_error
		GA_avg_open_chest_key_error
		GA_avg_read_message_loc_error
		GA_avg_walk_loc_error
		GA_avg_not_allowed_func_error
		GA_avg_function_param_error
Perte de niveaux	Nombre moyen par programme exécuté : $\frac{\textit{Nb perte niveaux}}{\textit{Nb programmes lancés}}$	GA_avg_spike_touch_lost
		GA_avg_barrel_explosion_lost
		GA_avg_pirate_shot_lost
Consultations de contenus	Temps moyen par consultation : $\frac{\textit{Temps total consultation}}{\textit{Nb consultations}}$	CO_avg_base_disp_time
		CO_avg_var_disp_time
		CO_avg_condi_disp_time
		CO_avg_for_disp_time
		CO_avg_while_disp_time

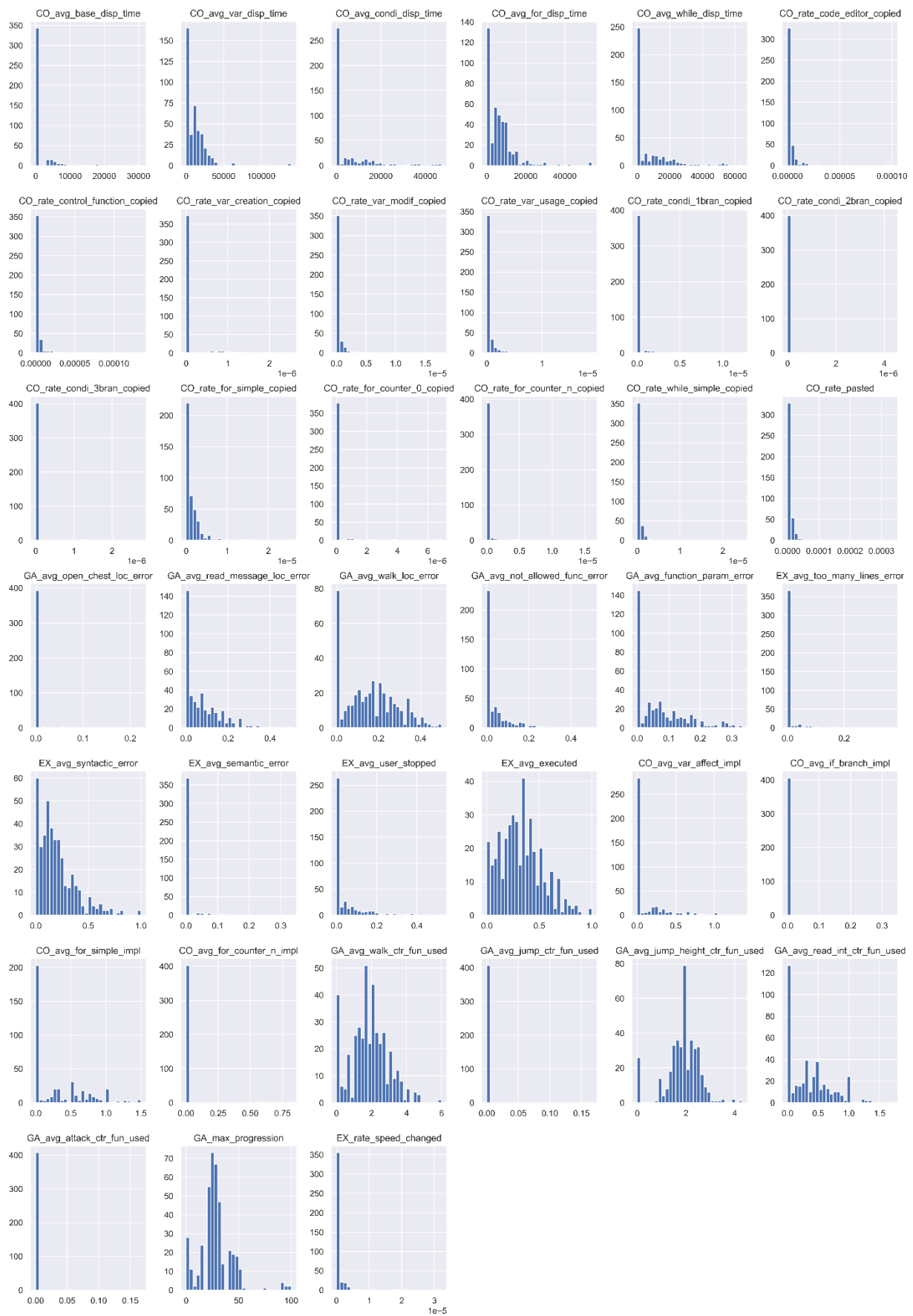
Copies de contenus	Taux temporel : $\frac{Nb\ contenus\ copiés}{Temps\ passé}$	CO_rate_code_editor_copied
		CO_rate_control_function_copied
		CO_rate_base_program_copied
		CO_rate_base_error_copied
		CO_rate_base_structuration_copied
		CO_rate_base_comment_copied
		CO_rate_var_creation_copied
		CO_rate_var_modif_copied
		CO_rate_var_usage_copied
		CO_rate_var_type_copied
		CO_rate_condi_1bran_copied
		CO_rate_condi_2bran_copied
		CO_rate_condi_3bran_copied
		CO_rate_for_simple_copied
		CO_rate_for_counter_o_copied
CO_rate_for_counter_n_copied		
CO_rate_while_simple_copied		
Collages de code	Ratio temporel : $\frac{Collage\ code}{Temps\ passé}$	CO_rate_pasted
Concepts implémentés	Nombre moyen par programme exécuté : $\frac{Nb\ concepts\ implémentés}{Nb\ programmes\ exécutés}$	CO_avg_var_affect_impl
		CO_avg_boolean_impl
		CO_avg_string_impl
		CO_avg_if_branch_impl
		CO_avg_elif_branch_impl
		CO_avg_else_branch_impl
		CO_avg_for_simple_impl
		CO_avg_for_counter_o_impl
		CO_avg_for_counter_n_impl
CO_avg_while_impl		
Erreurs dans le code	Nombre moyen par programme lancé : $\frac{Nb\ erreurs\ dans\ le\ code}{Nb\ programmes\ lancés}$	EX_avg_too_many_lines_error
		EX_avg_syntactic_error
		EX_avg_semantic_error
Programmes arrêtés	Nombre moyen par programme lancé : $\frac{Nb\ programmes\ arrêtés}{Nb\ programmes\ lancés}$	EX_avg_user_stopped
Programmes exécutés	Nombre moyen par programme lancé : $\frac{Nb\ programmes\ exécutés}{Nb\ programmes\ lancés}$	EX_avg_executed
Changement Vitesse	Ratio temporel :	EX_rate_speed_changed

	$\frac{\text{Nb changement vitesse}}{\text{Temps passé}}$	
--	-----------------------------------------------------------	--

# M.2 Histogrammes des caractéristiques



Niveau 1



### Niveau 3



## Niveau 4



## Niveau 5



# M.3 Matrices de corrélation des caractéristiques

CO_avg_base_disp_time	1	0.23	0.11	0.068	-0.083	-0.025	0.02	-0.049	-0.0032	-0.019	0.39	0.037	-0.0220	0.00820	0.055	-0.0095	-0.089	0.082	-0.035	-0.17	0.17	-0.0052	0.15	0.09	-0.081	-0.078	0.04	0.18	-0.044	-0.0075	0.14	-0.032	0.13	0.13	-0.098
CO_avg_var_disp_time	0.23	1	0.19	0.18	0.13	-0.022	0.044	0.15	0.17	0.18	0.022	-0.039	0.25	0.11	0.046	0.068	-0.032	0.024	-0.084	-0.19	0.26	0.09	0.082	0.074	0.08	-0.048	-0.087	0.35	0.051	0.069	0.081	0.005	0.12	0.1	-0.089
CO_avg_condi_disp_time	0.11	0.19	1	0.027	0.16	0.092	0.032	-0.044	-0.034	0.046	0.24	0.13	0.28	0.061	0.021	0.037	-0.0046	0.084	0.0099	0.0029	0.19	-0.055	-0.041	0.0002	-0.047	-0.019	0.045	0.051	0.063	0.042	0.23	0.049	0.11	0.13	-0.057
CO_avg_for_disp_time	0.068	0.18	0.027	1	0.19	0.14	0.058	-0.074	-0.025	-0.014	-0.11	0.0046	0.17	0.41	0.14	0.16	0.14	0.097	-0.058	0.011	0.0038	0.18	0.13	0.26	0.066	-0.058	-0.21	0.038	0.081	0.18	0.062	0.059	0.11	0.2	-0.17
CO_avg_while_disp_time	-0.093	0.13	0.16	0.19	1	-0.04	0.071	-0.062	-0.041	0.11	0.12	0.084	0.2	0.043	0.098	0.1	0.67	0.055	-0.096	0.093	0.22	-0.098	-0.059	0.12	-0.067	-0.042	-0.029	0.1	-0.025	0.12	0.065	0.012	0.091	0.22	-0.088
CO_rate_code_editor_copied	-0.025	-0.022	0.092	0.14	-0.04	1	0.28	0.083	-0.055	-0.049	-0.029	0.14	0.14	0.29	-0.049	0.027	-0.078	0.6	0.02	0.12	-0.048	0.12	0.14	0.026	0.26	-0.038	-0.12	0.04	-0.029	-0.0079	-0.045	-0.087	-0.062	0.04	0.036
CO_rate_control_function_copied	0.02	0.044	0.032	0.058	0.071	0.28	1	-0.049	-0.012	-0.038	0.099	-0.031	0.12	0.021	-0.036	0.042	-0.00097	0.55	-0.1	-0.049	-0.059	0.21	-0.092	0.029	-0.076	0.11	-0.056	-0.063	-0.022	-0.0063	0.0086	-0.074	0.0033	0.09	0.03
CO_rate_var_creation_copied	-0.049	0.15	-0.044	-0.074	-0.052	0.083	-0.049	1	0.84	0.12	-0.026	-0.016	-0.022	0.12	0.11	0.23	-0.034	0.21	0.26	-0.08	0.36	-0.025	0.074	-0.18	0.31	-0.034	-0.13	0.014	-0.013	-0.029	-0.021	0.018	-0.084	-0.1	0.035
CO_rate_var_modif_copied	-0.032	0.17	-0.034	-0.025	-0.041	-0.055	-0.012	0.84	1	-0.019	-0.023	-0.016	-0.019	0.16	0.12	0.28	-0.029	0.11	0.15	-0.048	0.47	-0.069	0.16	-0.14	-0.039	0.05	-0.088	0.015	-0.011	-0.0097	0.089	0.099	0.012	-0.014	0.039
CO_rate_var_usage_copied	-0.019	0.18	0.046	-0.014	0.11	-0.049	-0.038	0.12	-0.019	1	-0.02	-0.014	-0.017	-0.023	-0.017	-0.019	0.087	-0.019	0.028	-0.075	0.53	-0.028	-0.052	-0.09	0.13	-0.027	0.1	0.68	-0.0089	0.024	0.12	0.062	0.17	0.14	0.0039
CO_rate_condi_1bran_copied	0.38	0.022	0.21	-0.11	0.12	-0.029	0.099	-0.028	-0.023	-0.02	1	-0.017	-0.02	-0.04	-0.00036	0.023	0.0016	0.089	-0.069	-0.015	0.38	-0.035	-0.033	0.005	-0.042	-0.032	-0.035	-0.037	-0.012	-0.037	0.029	-0.04	0.01	0.017	0.048
CO_rate_condi_2bran_copied	0.037	-0.038	0.13	0.046	0.084	0.14	-0.031	-0.018	-0.016	-0.014	-0.017	1	-0.014	0.19	-0.014	-0.015	-0.021	0.079	-0.054	-0.061	0.08	-0.072	-0.043	0.023	-0.029	-0.022	0.038	-0.025	-0.0081	0.043	-0.068	-0.0071	-0.053	-0.064	-0.021
CO_rate_condi_3bran_copied	-0.022	0.25	0.29	0.17	0.2	0.14	0.12	-0.022	-0.019	-0.017	-0.02	-0.014	1	0.17	-0.017	-0.019	-0.026	0.2	-0.065	0.053	-0.048	0.1	-0.051	-0.016	-0.035	-0.026	0.016	-0.03	-0.0097	-0.031	0.037	-0.017	0.024	-0.0048	-0.045
CO_rate_for_simple_copied	0.00082	0.11	0.061	0.41	0.043	0.29	0.021	0.12	0.16	-0.023	-0.04	0.19	0.17	1	0.34	0.46	0.021	0.22	-0.054	0.00046	0.35	-0.022	0.36	0.21	-0.013	-0.038	-0.022	-0.048	-0.03	0.050	0.15	0.11	0.084	0.14	-0.071
CO_rate_for_counter_0_copied	-0.065	0.046	0.021	0.14	0.096	-0.049	-0.036	0.11	0.12	-0.017	0.00038	0.014	-0.017	0.34	1	0.47	-0.0082	0.059	-0.038	-0.063	0.18	0.12	0.071	0.18	-0.035	-0.027	-0.18	-0.031	-0.0099	-0.031	0.047	0.042	0.059	-0.023	-0.02
CO_rate_for_counter_n_copied	-0.0095	0.065	0.037	0.16	0.1	0.027	0.042	0.23	0.26	-0.019	-0.023	-0.015	-0.019	0.46	0.47	1	0.047	0.17	0.011	-0.082	0.34	0.085	0.22	0.054	-0.025	-0.047	-0.18	-0.034	-0.011	0.018	0.077	0.052	0.028	0.047	-0.034
CO_rate_while_simple_copied	-0.089	-0.032	-0.0046	0.14	0.67	-0.076	0.00097	0.034	-0.029	0.087	0.0016	-0.021	-0.026	0.021	-0.00682	0.047	1	0.0058	-0.058	0.1	-0.028	-0.08	-0.032	0.13	-0.054	-0.02	-0.019	0.015	0.17	0.023	-0.0022	0.02	0.23	-0.053	
CO_rate_pasted	0.082	0.024	0.064	0.097	0.055	0.6	0.55	0.21	0.11	-0.019	0.039	0.079	0.2	0.22	0.059	0.17	0.0058	1	0.11	-0.1	0.053	0.16	0.1	-0.0095	0.17	0.013	-0.12	-0.042	-0.038	0.016	-0.036	-0.11	-0.077	0.061	0.24
GA_avg_open_chest_loc_error	-0.035	-0.084	0.0098	-0.058	-0.086	0.02	-0.1	0.26	0.15	0.028	-0.009	-0.054	-0.054	-0.038	0.011	-0.068	0.11	1	-0.075	-0.0099	-0.076	-0.04	-0.13	0.25	-0.032	-0.25	-0.028	-0.038	0.001	-0.049	0.083	-0.11	-0.087	0.055	
GA_avg_walk_loc_error	-0.17	-0.19	0.0029	0.011	0.093	0.12	-0.049	-0.08	-0.048	-0.075	-0.015	-0.061	0.053	-0.00048	0.063	-0.082	0.1	-0.1	-0.075	1	-0.030	-0.29	0.18	-0.034	-0.079	0.028	0.014	-0.12	-0.043	0.11	0.091	0.29	0.073	0.19	-0.052
GA_avg_not_allowed_func_error	0.17	0.26	0.19	0.0038	0.22	-0.048	-0.059	0.36	0.47	0.053	0.39	0.09	-0.048	0.36	0.18	0.34	-0.028	0.053	-0.0099	-0.039	1	-0.21	0.26	-0.048	-0.074	-0.068	-0.12	-0.028	-0.015	0.12	0.036	0.068	0.042	-0.027	
GA_avg_function_param_error	-0.0052	0.09	-0.055	0.18	-0.098	0.12	0.21	-0.025	-0.069	-0.028	-0.095	-0.072	0.1	-0.022	0.12	0.085	-0.08	0.16	-0.076	-0.29	-0.21	1	-0.058	0.0081	0.17	0.13	-0.34	0.076	-0.072	-0.12	-0.053	-0.14	-0.012	-0.11	-0.092
EX_avg_too_many_lines_error	0.15	0.082	-0.041	0.13	0.059	0.14	-0.092	0.074	0.16	-0.052	-0.033	-0.043	-0.051	0.38	0.071	0.22	-0.032	0.1	-0.04	0.18	0.26	0.088	1	-0.036	0.031	-0.003	-0.12	-0.021	-0.003	0.065	0.33	0.22	0.23	0.24	-0.031
EX_avg_syntactic_error	0.08	0.074	0.0002	0.28	0.12	0.026	0.029	-0.18	-0.14	-0.09	0.085	0.023	-0.016	0.21	0.18	0.054	0.13	-0.0095	-0.13	-0.034	-0.048	0.0081	-0.036	1	0.0098	-0.11	-0.52	-0.053	0.052	0.008	0.042	0.042	0.049	0.085	-0.12
EX_avg_semantic_error	-0.081	0.08	-0.047	0.088	-0.067	0.28	-0.076	0.31	-0.039	0.13	-0.042	-0.029	-0.035	-0.013	-0.035	-0.025	-0.054	0.17	0.25	-0.079	-0.074	0.17	0.031	0.0098	1	0.0024	-0.23	0.13	-0.02	0.02	-0.12	0.02	-0.058	-0.1	-0.01
EX_avg_usr_stopped	-0.078	-0.048	-0.019	-0.058	-0.042	-0.038	0.11	-0.034	0.05	-0.027	-0.032	-0.022	-0.026	-0.038	-0.027	-0.0047	-0.02	0.013	-0.032	0.028	-0.058	0.13	-0.003	-0.11	0.0024	1	-0.057	-0.048	0.015	0.22	-0.025	0.061	-0.043	0.087	-0.019
EX_avg_executed	0.01	-0.057	0.015	-0.21	-0.029	-0.12	-0.058	-0.13	-0.088	0.1	-0.035	0.039	0.016	-0.22	-0.18	-0.18	-0.019	-0.12	-0.26	0.014	-0.12	-0.34	-0.12	-0.52	-0.23	-0.067	1	0.086	0.029	0.021	0.23	0.091	0.27	0.17	0.014
CO_avg_var_affected_impl	0.018	0.35	0.051	0.038	0.1	0.04	-0.063	0.014	0.015	0.68	-0.037	-0.025	-0.03	-0.048	-0.031	-0.034	0.082	-0.042	-0.026	-0.12	0.12	0.076	-0.021	-0.053	0.13	-0.048	0.065	1	0.018	0.036	0.16	0.072	0.28	0.16	-0.037
CO_avg_string_impl	-0.044	0.067	0.063	0.081	-0.025	-0.029	-0.022	-0.013	-0.011	-0.0099	-0.012	-0.0081	0.0097	-0.03	-0.0099	-0.011	-0.015	-0.038	-0.038	-0.043	-0.028	-0.072	-0.03	0.052	-0.02	-0.015	0.029	-0.018	1	0.57	0.0045	0.011	0.049	0.012	-0.016
CO_avg_for_simple_impl	-0.0075	0.009	0.042	0.18	0.12	-0.0079	0.0063	-0.029	-0.0097	0.024	-0.037	0.043	-0.031	0.050	-0.031	0.018	0.17	0.016	0.001	0.11	-0.015	-0.12	0.095	0.008	0.02	0.22	0.024	0.036	0.57	1	0.093	0.22	0.18	0.55	-0.05
GA_avg_walk_ctr_fun_used	0.14	0.081	0.23	0.062	0.065	-0.045	-0.0086	-0.021	0.089	0.12	0.029	-0.068	0.037	0.15	0.047	0.077	0.023	-0.036	-0.049	0.091	0.12	-0.053	0.33	0.042	-0.12	-0.025	0.23	0.16	0.0045	0.093	1	0.61	0.88	0.56	-0.088
GA_avg_left_ctr_fun_used	-0.032	0.005	0.049	0.059	0.012	-0.087	-0.074	0.018	0.099	0.062	-0.04	-0.0071	-0.017	0.11	0.042	0.052	-0.0022	-0.11	0.083	0.29	0.036														

CO_avg_base_disp_time	1	0.0048	0.081	0.064	0.063	-0.011	-0.028	-0.026	0.011	0.017	-0.0380	0.00570	0.036	-0.06	-0.025	0.019	0.023	-0.04	-0.034	0.19	-0.14	0.002	0.081	-0.014	0.24	0.068	-0.094	0.099	0.014	-0.022	-0.1	-0.035	0.12	-0.016	0.029	0.077	-0.016	0.07	0.0067
CO_avg_var_disp_time	0.0048	1	0.037	0.35	0.085	-0.07	-0.035	0.18	0.072	0.13	-0.019	0.049	-0.019	-0.015	-0.024	-0.047	-0.023	-0.039	0.045	0.064	-0.067	-0.022	0.023	0.026	-0.023	2e-010	0.012	0.052	0.16	-0.011	0.04	0.087	0.096	0.070	0.032	0.14	-0.0077	0.17	-0.021
CO_avg_condi_disp_time	0.081	0.037	1	0.095	0.43	-0.016	0.035	-0.085	0.071	0.025	0.2	0.17	0.13	-0.068	-0.054	0.037	0.072	-0.016	-0.05	0.1	-0.032	0.082	0.0210	0.0089	0.024	0.04	0.019	-0.12	-0.061	0.069	0.0005	0.057	-0.09	0.063	-0.12	-0.0370	0.063	-0.059	-0.031
CO_avg_for_disp_time	0.064	0.35	0.095	1	0.13	0.054	0.065	-0.014	0.039	0.14	-0.021	-0.034	0.039	0.12	0.0046	0.49	-0.046	0.061	-0.036	0.088	-0.046	-0.047	0.016	-0.059	-0.041	-0.009	-0.049	0.066	0.082	0.037	0.032	0.1	0.023	0.029	0.075	0.062	0.029	0.12	0.053
CO_avg_while_disp_time	0.083	0.085	0.43	0.13	1	-0.049	-0.028	0.110	0.053	0.036	0.12	0.014	0.077	-0.087	-0.090	0.089	0.22	-0.045	-0.07	0.11	-0.074	0.14	-0.044	0.12	0.0068	0.0570	0.025	0.078	-0.038	0.061	-0.029	0.14	-0.11	0.052	-0.21	0.028	0.052	-0.068	0.037
CO_rate_code_editor_copied	-0.011	-0.07	-0.016	0.054	-0.048	1	0.63	-0.077	0.064	0.120	0.0068	0.11	0.029	0.39	-0.043	0.5	-0.017	0.8	0.57	-0.043	0.075	-0.025	-0.028	0.018	0.0034	0.0024	0.038	0.044	0.012	0.079	0.032	-0.021	0.016	0.11	0.028	0.015	0.11	0.033	0.53
CO_rate_control_function_copied	-0.028	0.035	-0.035	0.065	-0.028	0.63	1	0.021	-0.0017	0.1	0.09	-0.011	0.015	0.56	0.0093	0.15	0.084	0.92	-0.025	-0.034	0.069	0.043	-0.012	-0.03	-0.036	0.055	-0.004	-0.006	-0.017	0.028	0.019	0.043	0.044	0.043	0.039	0.06	0.043	0.01	0.076
CO_rate_var_creation_copied	-0.026	0.18	-0.095	-0.014	0.011	-0.007	0.021	1	-0.022	0.1	-0.038	-0.032	0.681	0.11	-0.028	0.026	0.014	0.033	-0.023	0.0074	-0.068	-0.049	0.072	0.0042	0.026	0.011	0.061	0.994	0.2	-0.019	0.16	0.027	0.13	-0.014	0.087	0.16	-0.014	0.086	-0.016
CO_rate_var_modif_copied	0.011	0.072	0.011	0.039	0.0053	0.0054	0.017	0.022	1	0.33	0.024	0.033	0.0074	0.13	0.24	0.17	0.064	0.11	-0.027	0.0048	0.041	0.023	0.11	-0.0073	0.023	0.079	-0.066	-0.021	0.078	-0.016	0.047	0.035	0.014	-0.012	0.025	0.026	-0.012	0.028	-0.026
CO_rate_var_usage_copied	0.017	0.13	0.025	0.14	-0.036	0.12	0.1	0.33	1	-0.014	0.063	-0.03	0.36	0.1	0.054	-0.025	0.24	-0.032	0.13	-0.067	-0.043	0.071	-0.021	-0.054	0.0097	0.049	0.034	0.2	-0.019	0.06	-0.022	0.084	-0.013	0.051	0.13	-0.013	0.093	0.31	
CO_rate_condi_1bran_copied	-0.038	-0.019	0.2	-0.021	0.124	0.0065	0.09	-0.038	0.024	-0.014	1	0.093	0.029	-0.032	0.034	0.016	0.82	0.13	-0.024	-0.049	0.0021	0.22	-0.09	-0.013	0.091	-0.037	-0.025	0.075	0.05	0.11	0.039	-0.017	-0.013	0.1	-0.02	0.078	0.1	-0.022	0.016
CO_rate_condi_2bran_copied	0.0057	0.049	0.17	-0.034	0.014	-0.011	-0.011	-0.032	0.033	0.063	0.083	1	-0.013	-0.01	0.007	0.0044	0.025	0.019	-0.024	0.048	-0.12	0.1	0.015	-0.029	-0.030	0.0014	0.058	0.06	-0.025	0.008	-0.015	0.025	-0.0058	0.0094	0.060	0.080	0.088	-0.018	
CO_rate_condi_3bran_copied	-0.036	-0.019	0.13	0.039	0.077	0.029	0.015	0.061	0.074	-0.03	0.029	-0.013	1	-0.011	-0.025	-0.016	0.031	0.22	-0.018	-0.012	0.077	0.059	0.0066	0.028	0.018	0.22	-0.058	0.024	-0.029	0.34	0.0079	0.012	0.008	0.41	0.0024	0.088	0.1	-0.0097	0.025
CO_rate_for_simple_copied	-0.06	-0.015	-0.086	0.12	-0.087	0.39	0.56	0.11	0.13	0.38	-0.032	-0.01	-0.011	1	-0.017	-0.025	-0.037	0.61	-0.0083	0.029	-0.037	0.11	0.058	-0.018	0.082	0.023	-0.051	-0.055	0.038	-0.011	0.2	-0.024	0.027	0.043	0.057	0.024	0.043	0.013	0.12
CO_rate_for_counter_0_copied	-0.025	-0.024	-0.050	0.045	-0.089	-0.044	0.0068	0.029	0.24	0.1	-0.034	0.007	-0.025	-0.017	1	0.033	-0.02	0.039	-0.030	0.016	0.06	-0.023	0.013	-0.043	-0.050	0.096	0.130	0.0090	0.015	-0.015	0.094	-0.025	-0.011	-0.011	0.053	-0.078	-0.011	-0.014	-0.005
CO_rate_for_counter_n_copied	0.019	-0.047	-0.037	0.049	0.083	0.5	-0.10	0.026	0.17	0.054	-0.016	0.044	-0.025	0.033	1	0.012	0.14	-0.022	-0.045	0.017	0.013	0.069	-0.018	-0.020	0.068	-0.050	0.068	0.049	0.009	0.016	0.1	-0.06	0.090	0.049	-0.020	0.009	0.042	0.64	
CO_rate_while_simple_copied	0.023	0.023	0.072	-0.046	0.22	-0.017	0.084	-0.014	0.084	-0.026	0.82	0.025	0.031	-0.037	-0.02	0.012	1	0.12	0.0017	0.082	-0.047	0.35	-0.1	-0.02	0.06	0.043	-0.015	-0.057	0.07	-0.015	0.052	0.084	0.073	0.011	-0.016	0.11	0.011	-0.019	0.017
CO_rate_pasted	-0.04	-0.039	-0.016	0.081	-0.045	0.8	0.92	0.033	0.11	0.24	0.13	0.019	0.022	0.61	0.039	0.14	0.12	1	0.005	-0.028	0.051	0.026	-0.017	0.008	0.011	0.027	-0.02	-0.02	0.021	0.04	0.06	-0.009	0.045	0.06	0.051	0.072	0.06	0.011	0.22
GA_avg_open_ches_loc_error	-0.034	0.045	-0.05	-0.036	-0.07	0.067	-0.025	-0.027	-0.032	-0.024	0.0024	0.018	0.080	0.0030	0.022	0.017	0.005	1	-0.09	-0.065	-0.04	0.02	-0.024	0.21	-0.034	0.034	-0.095	-0.049	-0.011	0.036	-0.017	-0.11	-0.0077	-0.13	-0.034	0.070	0.042	-0.027	
GA_avg_read_message_loc_error	0.19	0.064	0.1	0.088	0.11	-0.043	-0.034	0.074	0.048	0.13	-0.049	0.046	-0.012	-0.029	-0.016	0.045	-0.082	-0.026	-0.09	1	-0.33	-0.13	-0.064	0.03	0.078	0.078	-0.048	-0.23	0.081	-0.046	-0.088	-0.063	0.029	-0.043	-0.058	0.12	-0.043	0.037	-0.026
GA_avg_walk_loc_error	-0.14	-0.057	-0.032	-0.046	-0.074	0.075	0.069	-0.038	-0.041	-0.067	0.0021	-0.12	-0.077	-0.037	0.06	0.017	-0.047	0.054	-0.065	-0.33	1	-0.23	-0.42	0.15	0.028	-0.1	0.055	-0.37	-0.14	-0.082	0.1	-0.06	-0.048	-0.065	-0.012	-0.14	-0.065	-0.11	0.1
GA_avg_not_allowed_func_error	0.002	-0.022	0.082	-0.047	0.14	-0.025	0.043	0.019	0.023	-0.043	0.22	0.1	0.099	0.11	-0.023	0.013	0.85	0.026	-0.04	-0.13	-0.23	1	-0.18	-0.043	0.037	-0.038	-0.031	-0.16	-0.12	0.025	-0.061	0.088	-0.072	0.02	-0.034	-0.064	0.02	-0.15	0.0023
GA_avg_function_param_error	0.081	0.023	0.021	0.016	0.044	-0.028	-0.012	0.072	0.11	0.071	-0.08	0.015	0.0060	0.058	0.013	0.069	0.1	-0.017	0.02	-0.064	-0.42	-0.18	1	0.059	0.01	0.15	0.01	-0.082	0.038	0.062	0.12	-0.023	-0.054	-0.088	-0.031	-0.081	0.088	-0.036	-0.035
EX_avg_too_many_lines_error	0.014	0.026	0.009	0.059	0.12	0.018	-0.03	0.0042	0.073	0.021	-0.013	-0.029	-0.028	-0.018	-0.043	0.018	-0.02	-0.0082	0.04	0.03	0.15	-0.043	-0.059	-1	0.0017	0.0076	0.037	-0.17	0.73	-0.017	0.14	-0.027	0.052	0.012	-0.025	0.031	-0.012	0.13	-0.029
EX_avg_syntactic_error	0.24	-0.082	0.024	-0.040	0.0080	0.034	0.036	-0.028	-0.023	-0.054	0.091	-0.038	0.018	0.062	-0.055	-0.052	0.06	-0.011	0.21	0.078	0.028	0.037	0.01	0.0017	1	-0.028	-0.21	-0.46	0.027	0.025	-0.023	-0.12	-0.15	-0.0087	-0.12	-0.084	0.067	-0.11	-0.072
EX_avg_semantic_error	0.0683	2e-05	0.4	-0.009	0.057	-0.0021	0.055	0.041	0.079	0.0097	0.037	0.014	0.22	0.023	-0.0096	0.0080	0.043	0.027	-0.034	0.078	-0.1	-0.038	0.15	0.0076	0.028	1	-0.11	-0.055	-0.013	-0.018	-0.082	0.0030	0.096	0.013	-0.09	-0.084	0.013	-0.11	-0.026
EX_avg_user_stopped	-0.094	0.012	0.019	-0.048	0.025	-0.038	-0.04	0.061	-0.068	-0.049	-0.025	-0.058	-0.051	-0.013	-0.051	-0.015	-0.02	0.034	-0.048	0.055	-0.031	0.01	-0.037	-0.21	-0.11	1	-0.1	0.026	-0.036	0.16	-0.057	0.17	-0.026	0.12	0.12	-0.026	0.11	-0.011	
EX_avg_executed	-0.089	0.052	-0.12	0.066	-0.078	0.044	0.006	0.094	-0.024	0.0034	0.075	0.06	0.024	-0.055	0.0060	0.068	0.057	-0.02	-0.025	-0.23	-0.37	-0.16	-0.082	-0.17	-0.46	-0.055	-0.1	1	0.14	0.044	0.078	0.15	0.2	0.044	0.28	0.21	0.044	0.3	-0.038
CO_avg_var_affect_impl	0.014	0.16	-0.001	0.082	-0.036	-0.012	-0.017	0.2	0.078	0.2	0.05	-0.025	-0.029	0.038	-0.015	-0.049	0.07	0.021	-0.049	0.081	-0.14	-0.12	0.038	0.073	0.027	-0.013	0.026	0.14	1	0.048	0.14	-0.004	0.41	0.044	0.33	0.6	0.044	0.44	0.057
CO_avg_if_branch_impl	-0.022	0.011	0.085	0.037	0.061	0.079	0.028	-0.019	-0.016	-0.019	0.11	-0.0081	0.34	-0.011	-0.015	0.0060	0.015	0.04	-0.011	-0.048	0.062	0.025	0.062	-0.017	0.025	-0.018	-0.036	0.044	0.048	1	0.024	-0.0076	0.039	0.83	0.0020	0.013	0.83	-0.016	-0.016
CO_avg_for_simple_impl	-0.1	0.040	0.005	0.032	-0.023	0.032	0.019	0.15	0.047	0.06	0.039	-0.087	0.079	0.2	0.094	0.016	0.052	0.06	0.036	-0.088	0.1	-0.081	-0.12	0.14	-0.023	-0.062	0.16	0.078	0.14	0.024	1	-0.085	0.15	0.048	0.35	0.28	0.048	0.34	0.064
CO_avg_for_counter_n_impl	-0.035	0.087	-0.057	0.1	0.14	-0.021	0.043	0.027	0.035	-0.022	-0.017	-0.013	-0.012	-0.024	0.025	0.1	0.084	-0.0097	0.017																				









**Titre :** Apprentissage de la programmation informatique : analyses et ressources pour accompagner la transition collège-lycée

**Mots clés :** enseignement-apprentissage de la programmation, transition blocs-texte, jeux sérieux, rétroactions automatiques, analytique des apprentissages, apprentissage automatique.

**Résumé :** L'objectif de cette thèse est d'abord de concevoir une ressource d'enseignement à même de les accompagner dans ces différents changements. L'étude des traces d'activités générées nous permet de valider globalement cette conception. Enfin, nous créons un système de rétroactions automatiques sélectionnées par des modèles d'IA afin d'améliorer l'autonomie des élèves. Ces modèles, qui font des prédictions en fonction de l'activité des élèves, sont entraînés à partir de données étiquetées par des enseignants lors d'une seconde expérimentation dans les classes de seconde impliquant 215 élèves.

Pour ce faire, nous commençons par réaliser une analyse épistémologique portant sur la programmation informatique, puis effectuons une étude approfondie des curriculums s'appuyant sur la Théorie anthropologique du didactique, et examinons en détails les différences entre les langages Scratch et Python. Nous menons ensuite une enquête auprès de 480 enseignants visant à connaître leurs pratiques d'enseignement. En nous basant sur ces analyses préalables, nous concevons l'application en ligne Pyrates sous la forme d'un jeu sérieux d'apprentissage

**Title :** Computer programming education: analyses and resources to support the transition from middle school to high school

**Keywords :** computer programming education, block-to-text transition, serious games, automatic feedback, learning analytics, machine learning

**Abstract :** The aim of this thesis is first to analyse the teaching and learning of programming at the transition between lower and upper secondary school (from grade 9 to grade 10) in France in order to enlighten the discontinuities that students may encounter when they learn to program, and then to design teaching materials that can support them through these various changes.

We began with an epistemological analysis of computer programming, followed by an in-depth study of curricula based on the anthropological theory of didactics, and a detailed investigation of the differences between the Scratch and Python programming languages. We then conduct a survey with 480 teachers to find out about their teaching practices.

Based on these preliminary analyses, we

designed the Pyrates online application as a serious game for learning the Python language. Its environment was tailored to facilitate the transition from blocks to text, and its eight levels involve the fundamental concepts of programming through playful situations conceived according to the Theory of didactical situations. We then evaluated this application in the field with 240 tenth-grade students. By studying the generated activity traces, we were able to validate the overall design.

Finally, we created an automatic feedback system based on AI models to improve students' autonomy. These models make predictions relying on student activity, and were trained on data labelled by teachers during a second field experiment involving 215 students.