



HAL
open science

Difficultés conceptuelles dans l'écriture d'algorithmes itératifs chez des élèves de collège

Béatriz Mejias-Dayoub

► **To cite this version:**

Béatriz Mejias-Dayoub. Difficultés conceptuelles dans l'écriture d'algorithmes itératifs chez des élèves de collège. Modélisation et simulation. Université Joseph-Fourier - Grenoble I, 1985. Français. NNT : . tel-00316715

HAL Id: tel-00316715

<https://theses.hal.science/tel-00316715>

Submitted on 3 Sep 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THESE

présentée à

l'Université Scientifique et Médicale de Grenoble

pour obtenir le grade de
DOCTEUR DE 3^{ème} CYCLE
«Mathématiques appliquées»
option Didactique

par

Béatriz MEJIAS épouse DAYOUB



DIFFICULTES CONCEPTUELLES DANS L'ECRITURE

D'ALGORITHMES ITERATIFS CHEZ DES ELEVES DE COLLEGE.



Thèse soutenue le 17 décembre 1985 devant la commission d'examen.

C. BENZAKEN
N. BALACHEFF
Ph. JORRAND
C. LABORDE
J. ROGALSKI

Président

Examineurs

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Année universitaire 1982-1983

Président de l'Université : M. TANCHE

MEMBRES DU CORPS ENSEIGNANT DE L'U.S.M.G.

(RANG A)

SAUF ENSEIGNANTS EN MEDECINE ET PHARMACIE

PROFESSEURS DE 1ère CLASSE

ARNAUD Paul	Chimie organique
ARVIEU Robert	Physique nucléaire I.S.N.
AUBERT Guy	Physique C.N.R.S.
AYANT Yves	Physique approfondie
BARBIER Marie-Jeanne	Electrochimie
BARBIER Jean-Claude	Physique expérimentale C.N.R.S. (labo de magnétisme)
BARJON Robert	Physique nucléaire I.S.N.
BARNOUD Fernand	Biosynthèse de la cellulose-Biologie
BARRA Jean-René	Statistiques - Mathématiques appliquées
BELORISKY Elie	Physique
BENZAKEN Claude (M.)	Mathématiques pures
BERNARD Alain	Mathématiques pures
BERTRANDIAS Françoise	Mathématiques pures
BERTRANDIAS Jean-Paul	Mathématiques pures
BILLET Jean	Géographie
BONNIER Jean-Marie	Chimie générale
BOUCHEZ Robert	Physique nucléaire I.S.N.
BRAVARD Yves	Géographie
CARLIER Georges	Biologie végétale
CAUQUIS Georges	Chimie organique
CHIBON Pierre	Biologie animale
COLIN DE VERDIERE Yves	Mathématiques pures
CRABBE Pierre (détaché)	C.E.R.M.O.
CYROT Michel	Physique du solide
DAUMAS Max	Géographie
DEBELMAS Jacques	Géologie générale
DEGRANGE Charles	Zoologie
DELOBEL Claude (M.)	M.I.A.G. Mathématiques appliquées
DEPORTES Charles	Chimie minérale
DESRE Pierre	Electrochimie
DOLIQUE Jean-Michel	Physique des plasmas
DUCROS Pierre	Cristallographie
FONTAINE Jean-Marc	Mathématiques pures
GAGNAIRE Didier	Chimie physique

.../...

GASTINEL Noël	Analyse numérique - Mathématiques appliquées
GERBER Robert	Mathématiques pures
GERMAIN Jean-Pierre	Mécanique
GIRAUD Pierre	Géologie
IDELMAN Simon	Physiologie animale
JANIN Bernard	Géographie
JOLY Jean-René	Mathématiques pures
JULLIEN Pierre	Mathématiques appliquées
KAHANE André (détaché DAFCO)	Physique
KAHANE Josette	Physique
KOSZUL Jean-Louis	Mathématiques pures
KRAKOWIAK Sacha	Mathématiques appliquées
KUPTA Yvon	Mathématiques pures
LACAZE Albert	Thermodynamique
LAJZEROWICZ Jeannine	Physique
LAJZEROWICZ Joseph	Physique
LAURENT Pierre	Mathématiques appliquées
DE LEIRIS Joël	Biologie
LLIBOUTRY Louis	Géophysique
LOISEAUX Jean-Marie	Sciences nucléaires I.S.N.
LOUP Jean	Géographie
MACHE Régis	Physiologie végétale
MAYNARD Roger	Physique du solide
MICHEL Robert	Minéralogie et pétrographie (géologie)
MOZIERES Philippe	Spectrométrie - Physique
OMONT Alain	Astrophysique
OZENDA Paul	Botanique (biologie végétale)
PAYAN Jean-Jacques (détaché)	Mathématiques pures
PEBAY PEYROULA Jean-Claude	Physique
PERRIAUX Jacques	Géologie
PERRIER Guy	Géophysique
PIERRARD Jean-Marie	Mécanique
RASSAT André	Chimie systématique
RENARD Michel	Thermodynamique
RICHARD Lucien	Biologie végétale
RINAUDO Marguerite	Chimie CERMAV
SENGEL Philippe	Biologie animale
SERGERAERT Francis	Mathématiques pures
SOUTIF Michel	Physique
VAILLANT François	Zoologie
VALENTIN Jacques	Physique nucléaire I.S.N.
VAN CUTSEN Bernard	Mathématiques appliquées
VAUQUOIS Bernard	Mathématiques appliquées
VIALON Pierre	Géologie

PROFESSEURS DE 2ème CLASSE

ADIBA Michel	Mathématiques pures
ARMAND Gilbert	Géographie

.../...

AURIAULT Jean-Louis	Mécanique
BEGUIN Claude (M.)	Chimie organique
BOEHLER Jean-Paul	Mécanique
BOITET Christian	Mathématiques appliquées
BORNAREL Jean	Physique
BRUN Gilbert	Biologie
CASTAING Bernard	Physique
CHARDON Michel	Géographie
COHENADDAD Jean-Pierre	Physique
DENEUVILLE Alain	Physique
DEPASSEL Roger	Mécanique des fluides
DOUCE Roland	Physiologie végétale
DUFRESNOY Alain	Mathématiques pures
GASPARD François	Physique
GAUTRON René	Chimie
GIDON Maurice	Géologie
GIGNOUX Claude (M.)	Sciences nucléaires I.S.N.
GUITTON Jacques	Chimie
HACQUES Gérard	Mathématiques appliquées
HERBIN Jacky	Géographie
HICTER Pierre	Chimie
JOSELEAU Jean-Paul	Biochimie
KERCKOVE Claude (M.)	Géologie
LE BRETON Alain	Mathématiques appliquées
LONGEQUEUE Nicole	Sciences nucléaires I.S.N.
LUCAS Robert	Physiques
LUNA Domingo	Mathématiques pures
MASCLE Georges	Géologie
NEMOZ Alain	Thermodynamique (CNRS - CRTBT)
OUDET Bruno	Mathématiques appliquées
PELMONT Jean	Biochimie
PERRIN Claude (M.)	Sciences nucléaires I.S.N.
PFISTER Jean-Claude (détaché)	Physique du solide
PIBOULE Michel	Géologie
PIERRE Jean-Louis	Chimie organique
RAYNAUD Hervé	Mathématiques appliquées
ROBERT Gilles	Mathématiques pures
ROBERT Jean-Bernard	Chimie physique
ROSSI André	Physiologie végétale
SAKAROVITCH Michel	Mathématiques appliquées
SARROT REYNAUD Jean	Géologie
SAXOD Raymond	Biologie animale
SOUTIF Jeanne	Physique
SCHOOL Pierre-Claude	Mathématiques appliquées
STUTZ Pierre	Mécanique
SUBRA Robert	Chimie
VIDAL Michel	Chimie organique
VIVIAN Robert	Géographie

A MI PADRE †,
A MI MADRE,
A MI ESPOSO JORGE
Y A MI HIJO JORGE ELIAS

Je tiens à remercier,

Monsieur Claude Benzaken, qui m'a fait l'honneur de présider le jury de cette thèse.

Monsieur Philippe Jorrand, qui s'est intéressé à ce travail et a accepté de faire partie du jury.

Madame Janine Rogalski qui a bien voulu porter une attention amicale et critique à mon travail.

Monsieur Nicolas Balacheff, qui a toujours su m'encourager et me prodiguer les meilleurs conseils tout au long de ce travail. Je tiens à le remercier pour le soutien professionnel et amical qu'il m'a donné.

Madame Colette Laborde, qui a bien voulu diriger cette thèse, pour la confiance qu'elle m'a témoignée en tout moment. Le travail présenté dans cette thèse est le fruit de ses orientations et critiques toujours pertinentes. Qu'elle soit assurée de ma gratitude, mon estime et mon amitié.

Je ne voudrais pas dissocier de ces remerciements tous les membres de l'équipe de didactique des mathématiques et de l'informatique et le directeur du Laboratoire Structures Discretes et Didactique, M. Jean Marie Laborde pour l'excellent accueil qu'ils m'ont réservé et pour l'environnement favorable, sans lequel cette thèse n'aurait pas eu lieu. En particulier, je remercie Mlles. Mireille Dupraz et Iman Osta qui, en qualité de collègues d'équipe et amies ont participé, avec patience, à la "mise en français" de ce travail. De même je remercie Mlle. M. Christine Vojik pour son efficacité et son amabilité dans la réalisation de la dactylographie et des détails techniques.

Je n'oublierais pas de remercier M. André Rouchier pour l'aide précieuse qu'il m'a accordée lors de la réalisation de mon expérimentation à Orléans.

J'aimerais encore remercier tous les enseignants des Collèges qui ont bien voulu collaborer aimablement pour la mise en place des expériences ; en particulier je remercie M. Bernard Capponi du Collège du Vergeron à Moirans, ainsi que Mme. Danielle Balacheff du Collège de Domène.

Et avant de finir, je tiens à mentionner dans ce chapitre de remerciements mes amies: Mary Cisneros, Maria Blanca Ibañez, Sylvie Rogé et Orieta Santana, qui m'ont été proches dans l'étape finale de ce travail.

Ces remerciements seraient incomplets si je ne mentionne pas mes amies les secrétaires de l'I.R.E.M. de Grenoble, ainsi que tous les membres du service de reprographie de l'I.M.A.G.

Cette thèse a pu être faite grâce à la bourse de "La Fundacion Gran Mariscal de Ayacucho" (Venezuela).

TABLE DE MATIERES

INTRODUCTION

I- Une recherche en didactique de l'informatique	6
II- Les questions à l'origine de la recherche	7
III- Cadre théorique de la recherche	8
IV- Problématique de la notion d'itération	9
IV.1 Les branchements	9
IV.2 Les structures de contrôle	10
IV.3 Illustration dans quelques langages de programmation	13
V- Analyse conceptuelle de la notion d'itération	16
VI- Méthodes expérimentales	18

CHAPITRE I : UNE ENQUETE SUR LA COMPREHENSION DES EXPRESSIONS ALTERNATIVES ET REPETITIVES

I- Présentation de l'enquete

I.1 Tâche	22
I.2 Choix	22

II- Questionnaire n°1

II.1 Le questionnaire	24
II.2 Commentaires à propos des choix des exercices posés et des solutions proposées	26
II.3 Résultats et interprétations (tableaux)	28

III- Questionnaire n°2	
III.1 Le questionnaire	33
III.2 Commentaires à propos des choix des exercices posés et des solutions proposées	35
III.3 Résultats et interprétations (tableaux)	37
IV- Conclusion	40
CHAPITRE II : CONSTRUCTION D'UNE ITERATION DANS UNE SITUATION DE COMMUNICATION ELEVE-ELEVE	
I- Les choix de l'expérience	44
II- Description et déroulement de l'expérience	45
III- Analyse de la situation	
III.1 Phase émission des messages	48
III.2 Phase réception des messages	51
IV- Phase d'émission	
IV. 1 Résultats	51
IV. 2 Conclusion	59
V- Phase de réception	
V. 1 Résultats	60
V. 2 Les échanges	63
V. 3 Conclusion	67
VI- Phase de travail individuel	
VI. 1 Résultats	68
VI. 2 Exemples d'évolution phase 1- phase 5	70
VI. 3 Conclusion	71
VII- Conclusion	72

CHAPITRE III : CONSTRUCTION D'UNE ITERATION DANS UNE SITUATION DE COMMUNICATION ELEVE - MACHINE VIRTUELLE

I- Les choix de l'expérience	76
II- Description et déroulement de l'expérience	79
III- Première séance	
III.1 Fonction des instructions du langage et analyse a priori de la construction par les élèves d'un programme avec itération ...	83
III.2 Analyse des observations	84
III.3 Exemples d'élaboration d'un programme	91
IV- Deuxième séance	
IV.1-Fonction des instructions du langage	97
IV.2- Analyse des observations	97
IV.3- Exemples d'élaboration d'un programme	102
V- Conclusion	104

CHAPITRE IV : CONSTRUCTION D'UNE ITERATION DANS UNE SITUATION DE COMMUNICATION ELEVE - MACHINE REELLE

I- Choix de l'expérience	108
II- Description et déroulement de l'expérience	109
III-Analyse a priori de la construction d'un programme avec itération .	110
IV-Analyse des observations	111
V-Exemples de programmes réalisés par les élèves	113
VI- Conclusion	119

CONCLUSION	
I . Aspects communs dans la construction de l'itération	122
II. Différences dues aux caractéristiques des langages	124
BIBLIOGRAPHIE	127
ANNEXE 1 : Expérience de construction d'une itération dans une communication avec un robot virtuelle (Chap. III)	
Textes, consignes, schémas des claviers et significations des touches distribués aux élèves.....	131
ANNEXE 2 : Expérience de construction d'une itération en BASIC (Chap. IV)	
1-Transcription de la présentation par l'enseignant du problème à résoudre.....	138
2-Breve description des 6 séances antérieures.....	140

INTRODUCTION

I- Une recherche en didactique de l'informatique

II- Les questions à l'origine de la recherche

III- Cadre théorique de la recherche

IV- Problématique de la notion d'itération

IV.1 Les branchements

IV.2 Les structures de contrôle

IV.3 Illustration dans quelques langages de programmation

V- Analyse conceptuelle de la notion d'itération

VI- Méthodes expérimentales

I- UNE RECHERCHE EN DIDACTIQUE DE L'INFORMATIQUE

Avec l'introduction de l'enseignement de l'informatique dans les collèges, et plus précisément de l'enseignement de la programmation, de nombreux problèmes se posent aux éducateurs, aux enseignants et aux formateurs à propos des difficultés et des obstacles conceptuels relatifs à l'apprentissage des nouvelles notions introduites.

Les études concernant l'apprentissage des notions impliquées dans cet enseignement prennent de l'importance dans la mesure où elles sont susceptibles d'orienter les choix relatifs aux programmes au contenu et aux méthodes d'enseignement dans le domaine concerné. Ces études relèvent de la didactique de l'informatique, qui a pour objet l'étude des processus d'enseignement et d'apprentissage en situation scolaire des savoirs informatiques. Il s'agit de dégager les spécificités de ces processus liées au contenu informatique en jeu. On peut considérer que les situations d'apprentissage de l'informatique à l'école prennent place dans un système s'articulant autour de quatre éléments fondamentaux : les élèves, l'enseignant, la machine et le savoir impliqué. "Chaque acteur (ou élément) de ce système peut être étudié en tant que tel, et à ce titre il intéresse une discipline déterminée : psychologie, épistémologie, sociologie de l'enseignement, etc... Ces études spécialisées intéressent la didactique mais elles ne sauraient suffire à fournir le cadre problématique et méthodologique nécessaire à l'analyse et à la compréhension des phénomènes didactiques" (Rouchier, Samurçay, 1984).

En effet ces phénomènes ne concernent pas les éléments de ce système de façon isolée mais en interaction. Quelles sont les spécificités des notions à apprendre qui posent problème aux élèves ? Quelles représentations les élèves élaborent-ils de ces notions ? Quels sont les moyens à mettre en œuvre dans l'enseignement pour permettre une évolution de ces représentations ? ... Telles sont quelques unes des questions auxquelles veulent répondre les recherches en didactique de l'informatique.

La recherche présentée ici se situe dans cette perspective, notamment parce qu'elle se fixe comme objectif général l'étude des conceptions initiales et de leur évolution chez les élèves à propos d'une des notions fondamentales de la programmation : **l'itération.**

De par le type des langages de programmation (notons que ces langages sont les descendantes du modèle de Von Neumann, appelés par Backus

1978 : langages convetionnels) susceptibles d'être enseignés au collège cette notion est l'une des premières introduites dans l'enseignement.

L'itération a pour fonction primordiale de faciliter des tâches répétitives longues à réaliser par l'homme ; elle apparaît comme une structure indispensable dans le cas de la réalisation d'algorithmes où, une suite d'actions doit être exécutée un grand nombre de fois ou un nombre inconnu a priori. Elle se révèle alors être une solution fonctionnelle et efficace à des problèmes mettant en jeu des actions répétitives, problèmes de base de la programmation.

II- LES QUESTIONS A L'ORIGINE DE LA RECHERCHE

Différents travaux sur les problèmes d'initiation à la programmation et plus précisément ceux qui concernent la notion de boucle ont inspiré cette recherche. Ces travaux (Hoc 1977, 1979, Soloway, Ehrlich et al 1982, Pea 1984) ont dégagé les différences dans les processus d'acquisition de notions informatiques entre les "débutants" et les "non-débutants". Les difficultés présentées par les différents types de boucles dans des langages procéduraux (PASCAL)(Soloway, Ehrlich et al 1982 ; Rouchier, Samurçay et al 1984 ; Rogalski 1985) lors d'un apprentissage de la programmation ont aussi été analysées.

Notre recherche se démarque des précédentes sous deux aspects :

1) A la différence des travaux existants, elle concerne des élèves de collège (11-15 ans) qui n'ont encore jamais reçu d'enseignement en informatique du moins à l'école. Notre travail a pour objectif de dégager les conceptions initiales des élèves avant tout enseignement de la programmation, et de déterminer comment ces conceptions évoluent lorsque les élèves sont confrontés à des problèmes d'écriture d'algorithmes itératifs.

2) Elle envisage ces questions en accordant une attention spéciale au contexte de communication avec la machine. En effet, une des hypothèses fondamentales de notre recherche réside dans le fait que toute activité d'écriture d'un algorithme prend son sens dans un contexte de communication. Le programme écrit est destiné à un interlocuteur (homme ou machine) avec le but de lui faire exécuter un ensemble déterminé a priori d'actions. La conception d'un algorithme se fait en fonction des moyens disponibles pour son expression qui dépendent eux-mêmes du contexte de communication dans lequel on se place.

Dans ce contexte, l'écriture d'un programme mobilise :

- les conceptions qu'a le rédacteur du programme, du fonctionnement de l'exécutant du programme et des interprétations que fait ce dernier du programme qu'il a à exécuter (Rouchier et Samurçay 1984 p. 18).
- la connaissance et le maniement du code utilisé pour communiquer.

C'est pourquoi, l'ensemble constitué par l'opérateur (homme ou machine) et le langage de communication avec cet opérateur est pris pour variable dans les expérimentations que nous avons faites. Nous cherchons à préciser les dépendances entre les processus de construction d'un algorithme itératif et le contexte de communication avec l'itération.

Trois contextes ont été choisis, de caractéristiques très contrastées :

- l'opérateur est un autre élève, le langage de communication est la langue naturelle,
- l'opérateur est un robot virtuel, le langage de communication est un langage artificiel "ad hoc" créé par nous, de syntaxe très rudimentaire,
- l'opérateur est une machine réelle, un micro ordinateur Thomson TO7, le langage de communication le BASIC résidant dans la machine.

Les représentations que les élèves programmeurs élaboreront du fonctionnement de la machine, les contraintes du langage de communication sont susceptibles d'être très différentes d'un contexte à l'autre, et on pourrait s'étonner de l'ampleur des variations choisies ici. Il nous paraissait illusoire de déterminer l'effet de fines variations dans cette approche, alors que nous ne disposions que d'un nombre très faible de données antérieures sur le rôle joué par le contexte de communication avec la machine.

III- CADRE THEORIQUE DE LA RECHERCHE

Il y a différentes façons d'envisager les problèmes d'acquisition des connaissances et diverses théories ont pu être élaborées à ce propos au cours du temps. Notre travail se place dans une approche constructiviste de

la constitution de connaissances par le sujet apprenant.. Dans le domaine des mathématiques ou de l'informatique, l'interaction entre le sujet et les objets à la base de cette construction joue un rôle décisif lorsque l'individu est confronté à une situation problématique. "La solution de problème est la source et le critère du savoir. C'est dans la solution de problème, ou plus généralement dans le traitement de situations-problèmes que sont

élaborées les notions, et que sont abstraites les propriétés pertinentes. C'est aussi dans la solution de problème que sont éprouvées les connaissances opératoires" (Vergnaud, 1981)

Cette position théorique a deux implications très importantes :

1) à propos des connaissances mathématiques et informatiques : elles prennent leur signification dans l'ensemble des problèmes qu'elles permettent de résoudre. L'étude épistémologique des savoirs en question est un moyen d'approcher leur signification parce qu'elle met en lumière les problèmes auxquels a été confrontée la communauté scientifique et qui ont permis la construction de ces nouveaux savoirs.

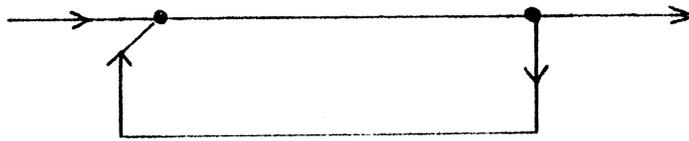
2) à propos de l'évolution du système de connaissances de l'élève : au cours de la résolution "du problème", il engage ses connaissances antérieures, les éprouve, les modifie ou les rejette éventuellement pour en élaborer de nouvelles plus adaptées à l'élaboration d'une solution (Brousseau, 1976).

Nous avons donc choisi de confronter les élèves à des situations problèmes pour lesquelles l'itération constitue un outil de résolution efficace. La construction de ces situations problèmes requiert une connaissance des problèmes qui ont été à l'origine des caractéristiques prises actuellement par les différentes structures itératives. Le paragraphe suivant essaie de dégager rapidement les caractéristiques les plus importantes de l'itération élaborées au cours de l'évolution des langages de programmation.

IV- PROBLEMATIQUE DE LA NOTION D'ITERATION

IV.1 Les branchements

Le déroulement normal du traitement d'un programme par une machine consiste à exécuter, les unes après les autres, les instructions indiquées. Pour faire exécuter plusieurs fois une même séquence d'instructions, il faudrait soit la réécrire autant de fois que nécessaire, mais cela peut être impossible ou au moins très coûteux, soit disposer de moyens pour modifier l'exécution séquentielle. La solution apportée à ce problème, dès les premiers langages de programmation, consiste à définir une instruction de branchement qui permet d'obtenir des retours en arrière ou des sauts en avant dans le programme, ce que représente le schéma suivant :

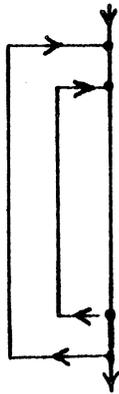


(fig 1)

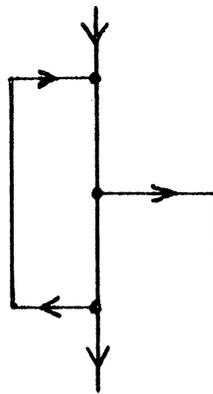
L'expression la plus courante d'un branchement est la locution GOTO (en français ALLER A) bien connue de la plupart des langages de programmation.

Un branchement permet de réaliser une itération. Mais, pour que l'exécution se termine dans un temps fini, ou autrement que sur un message d'erreur de la machine, il est nécessaire d'introduire une **condition** qui permet de contrôler l'arrêt de l'itération.

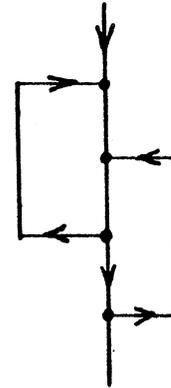
L'utilisation des branchements est a priori peu contraignante et plusieurs configurations d'algorithmes peuvent être exprimées, notamment celles traduites par les schémas suivants :



(fig. 2)



(fig.3)



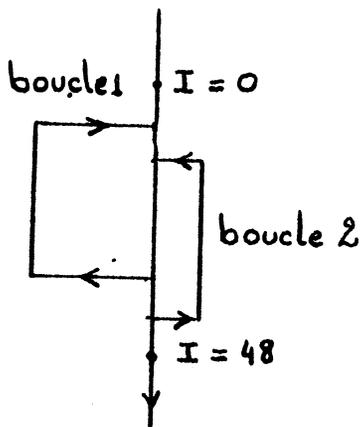
(fig.4)

IV.2 Les structures de contrôle

Au cours du développement des langages de programmation, divers problèmes ont été soulevés qui avaient trait à la conception des algorithmes, à la clarté et à la lisibilité des programmes et à leur validation.

L'étude de ces problèmes a fait évoluer la conception des langages de programmation, un programme ne doit pas être seulement correct mais aussi clair et lisible pour un utilisateur différent du programmeur. La liberté de l'utilisation des instructions de branchements (GOTO) est apparue comme un obstacle à la validation et même à la conception des algorithmes. Dijkstra, dans sa lettre écrite à l'éditeur de ACM intitulée "GOTO statements considered harmful" signale que l'utilisation de ces branchements est une des causes principales des erreurs de programmes ; de plus elle les rend difficiles à suivre ou à lire par un tiers. En effet, au delà d'une complexité vite atteinte, des configurations telles que celles présentées par la figure 4, transforment l'algorithme en un écheveau difficile à démêler. Plus fondamentalement les tentatives de validation se heurtent à la détermination du "passé" d'une variable, qui devient très difficile, voire impossible à faire dans deux boucles qui se chevauchent (Dijkstra, 1972).

Voyons l'exemple suivant:



Dans ces deux boucles enchevêtrées, la valeur de I avant d'entrer dans les boucles est zéro, la variable I est incrémentée de 3 dans la boucle 1 et de 9 dans la boucle 2

Supposons qu'à la sortie des deux boucles, I ait pour valeur 48, et essayons de déterminer à partir de cette valeur le nombre x de passages par la boucle 1 et le nombre y de passages par la boucle 2.

x et y sont solutions entières et positives de l'équation : $3x + 9y = 48$.

Or cette équation admet plusieurs solutions : les couples (1,5), (4,4), (7,3), (10,2), (13,1). Donc, dans ce cas, les seules valeurs des variables du programme à la sortie des boucles ne permettent pas de déterminer le nombre de passages par chaque boucle.

Une solution au problème de détermination des nombres de passage dans les boucles 1 et 2 consiste à utiliser une variable auxiliaire J initialisée à 1 et dont les valeurs sont modifiées à chaque passage de la boucle ;

dans la boucle 1: $J \leftarrow J \times 2$

dans la boucle 2: $J \leftarrow J \times 3$

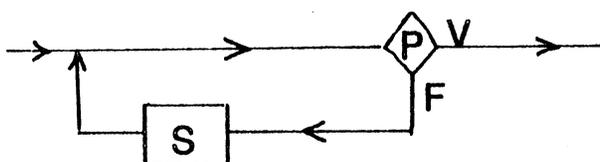
A la sortie des 2 boucles $J = 2^m \times 3^p$ m et p étant les nombres de passage respectivement dans les boucles 1 et 2.

$$J = 2^m \times 3^p$$

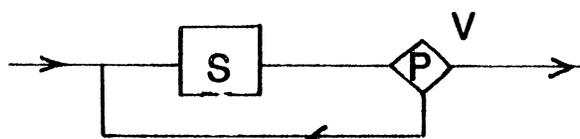
La réponse apportée par le mouvement de la programmation structurée à ces problèmes de clarté et de lisibilité consiste à utiliser des structures de contrôle n'ayant qu'une seule entrée et une seule sortie. Un ensemble est ainsi transformé en une seule instruction, unité indépendante des autres instructions dans le programme.

En ce qui concerne les structures itératives, deux structures sont introduites.

La boucle TANT QUE FAIRE



La boucle REPETER JUSQU'A CE QUE



Il faut souligner la rupture que présente la construction de telles structures avec l'instruction de branchement. L'ensemble des instructions à répéter est non seulement isolé mais devient un objet sur lequel opèrent des instructions. Alors que l'utilisation de branchement suit de près l'exécution des instructions, la distance à l'exécution augmente dans ce type de structures : ainsi le branchement n'apparaît plus de façon explicite.

Dans ces structures un constituant fondamental de l'itération est mis en évidence, nous l'appellerons **corps de l'itération**.

En fait se limiter à cette expression de la boucle (ce qui serait formellement suffisant, Böhm et Jacopini 1966) peut augmenter "inutilement" la complexité de certains algorithmes, par exemple lorsqu'on voudrait sortir prématurément du corps de l'itération (fig. 3). La possibilité d'un usage bien maîtrisé et limité de branchements de type GOTO a donc trouvé, des défenseurs (Arsac 1983, Knuth 1974) et les plus formalistes se sont ouverts à la possibilité de structures formellement redondantes (telles les D'-structures, Legard et Marcoty 1975).

L'itération REPETER ne se distingue pas seulement de TANT QUE par le mode de délimitation du corps de l'itération, mais aussi par sa signification

du point de vue algorithmique : pour la première, le corps de l'itération est toujours exécuté au moins une fois, l'examen de la condition étant fait après cette exécution, tandis que pour la deuxième, cette première exécution du corps de l'itération ne se fait pas si la condition examinée avant d'y entrer n'a pas la valeur vrai. C'est pourquoi cette expression de l'itération est considérée comme moins générale que l'expression TANT QUE (en revanche, l'effet de REPETER peut toujours être obtenu à l'aide de TANT QUE par le simple déplacement de la condition).

Une forme d'itération correspondant à un schéma de boucle du type que représente la figure 3 est actuellement reprise par un certain nombre d'ouvrages de méthodologie de la programmation ou d'algorithmique (Scholl 1979, Tennent, 1981). Il s'agit des expressions suivantes C1 ou C2 qui désignent deux séquences d'instructions :

```
ITERER
  <C1>
  <condition >
  <C1>
FIN ITERER
```

Cette itération apparaît comme plus générale que les précédentes au sens où dans le cas où C1 est vide, elle est équivalente à celle exprimée par TANT QUE, dans le cas où C2 est vide elle est équivalente à celle exprimée par REPETER. Le corps de l'itération est délimité par les "parenthèses" ITERER et FINITERER, mais cette dernière locution marque aussi un branchement inconditionnel sur le début de la boucle indiqué par ITERER; dans cette expression la condition n'est pas repérée par un marqueur. Dans le cas particulier où le nombre d'exécutions du corps de l'itération est connu a priori, la condition peut porter sur un compteur dont la valeur sera augmentée d'une unité à chacune de ces exécutions. Ce type de contrôle du déroulement d'une boucle est traduit dans de nombreux langages de programmation par une expression spécifique du type :

```
Pour <compteur> = <valeur initiale> JUSQU'A <valeur finale>
Pas <valeur du pas> FAIRE <corps de l'itération> FIN POUR
```

IV.3 Illustration dans quelques langages de programmation

L'évolution qui vient d'être tracée peut être illustrée par les formes prises par l'itération dans quelques langages.

Les langages d'assemblages, utilisant des codes mnémoniques, ne connaissent que les branchements qui provoquent le saut vers la place du programme repérée par une étiquette.

Une première structure itérative apparaît dans FORTRAN à côté de l'utilisation des branchements GOTO : c'est l'instruction DO qui permet l'exécution des suites d'instructions, un nombre de fois fini et connu a priori ou calculé dans le programme avant d'entrer dans la boucle.

```
DO E1 I = n1, n2, n3
    suite d'instructions
E1 CONTINUE
```

où E1 est une étiquette qui délimite la suite d'instructions
I est l'indice de parcours des valeurs (n₁, n₂)
n₁ est la valeur de départ
n₂ est la valeur d'arrivée
n₃ est le pas du parcours ; si n₃ = 1 il peut ne pas être mis.

La séquence d'instructions située entre l'instruction DO et l'étiquette E1 constitue le corps de la boucle qui est délimité aussi par l'instruction DO (n₁, n₂, n₃) et la rencontre de l'étiquette. La boucle est considérée comme finie quand la valeur du paramètre I est plus grande que la valeur assignée à n₂.

Avec ALGOL et la famille des ALGOLs sont introduites les structures itératives TANT QUE FAIRE et REPETER JUSQUA CE QUE grâce à la mise en place des délimiteurs (Début Fin) qui permettent de traiter une suite de plusieurs structures comme une unité (Bloc structuré).

En revanche, BASIC langage créé pour l'enseignement, ne connaît pas ces structures itératives et ne permet que le branchement (GOTO contrôlé par l'instruction IF THEN) et l'instruction FOR..NEXT dans laquelle il faut connaître a priori le nombre de répétitions à faire.

On donne ci-dessous, des programmes écrits en assembleur, FORTRAN et PASCAL qui calculent la somme des 100 premiers nombres entiers.

-En assembleur (PDP-11 45)

```
DEBUT: MOV #1,R0           ;place la valeur 1 dans le registre R0
        MOV #0, R1        ; et la valeur 0 dans le registre R1.
BOUC : ADD R0, R1         ;ajoute à R1 le contenu de R0.
        ADD #1, R0        ; ajoute à R0 le nombre 1.
        CMP #100, R0     ;compare la valeur de R0 au nombre 100.
        BGT BOUC        ;se branche sur BOUC si R0 est
                        ;(superieur ou) égal à 100.
        .HALT           ; fin du programme.
```

-En FORTRAN

Avec IF et GOTO :

```
INTEGER SOMME, I
SOMME = 0
I = 0
1 I = I + 1
  SOMME = SOMME + I
  IF (I.LT. 100) GOTO 1
  PRINT, SOMME
  STOP
  END
```

Avec DO:

```
INTEGER SOMME, I
SOMME = 0
DO 1 I = 1, 100
  SOMME = SOMME + I
1 CONTINUE
  PRINT, SOMME
  END
```

-En PASCAL (ces programmes calculent la somme des 100 premiers nombres pris au hasard)

Avec l'instruction FOR

```
Program Sommepr;
Var i,somme: integer;
function alea: integer; external;
Begin
  somme:=0;
  for i:=1 to 100 do somme:=somme + alea;
  writeln (somme)
end.
```

```

Avec l'instruction REPEAT
Program Sommepré1;
Var i, somme : integer;
function alea : integer; external;
Begin
  i :=1;
  somme := 0;
  REPEAT
    somme := somme + alea;
    i := i+1;
  UNTIL I > 100;
  writeln (somme)
end.

```

```

Avec l'instruction WHILE
Program Sommepré2;
Var i,somme: integer;
function alea: integer; external;
Begin
  i:=1;
  somme:=0;
  while i<=100 do
    begin
      somme:= somme + alea;
      i:= i+1
    end;
  writeln (somme);
end.

```

V. ANALYSE CONCEPTUELLE DE LA NOTION D'ITERATION

Le paragraphe précédent a mis en évidence les différents problèmes que posent la construction et l'expression d'une itération.

Pour construire une boucle, nous devons résoudre le problème de l'expression de la condition d'arrêt et celui de la détermination du corps de l'itération :

- le corps de l'itération est caractérisé par l'invariance de la séquence d'instructions qui le compose. La reconnaissance de ce corps de l'itération ne suffit pas. Comme on l'a vu, l'utilisation d'une structure itérative nécessite de le considérer comme un objet sur lequel on fait opérer une instruction. Le problème de son expression est le même que celui de la délimitation de cette séquence comme sous-ensemble des instructions du programme qui

expriment l'algorithme. Les solutions apportées sont multiples, comme on l'a vu, elles peuvent se ramener aux types principaux suivants :

TANT QUE <condition> FAIRE <corps de l'itération> FIN TANT QUE

REPETER <corps de l'itération> JUSQU'A CE QUE <condition>

- le rôle de la condition est d'assurer que l'algorithme "ne boucle pas indéfiniment". Il faut donc analyser quelles sont les variables nécessaires au contrôle de la boucle, et s'assurer qu'elles prendront les valeurs permettant de décider la fin de l'exécution. La formulation de la condition consistera alors en une expression logique formulée par ces variables.

L'expression de l'itération doit en plus comporter un **avertisseur d'itération** et des moyens de **délimiter** le corps de l'itération.

TANT QUE et REPETER sont des **avertisseurs** : ces expressions annoncent que l'on va avoir à faire à une itération. Le corps de l'itération est délimité dans le premier cas par les "parenthèses" FAIRE et FIN TANT QUE, et, dans le second cas par REPETER et JUSQU'A CE QUE.

La relation entre le corps de l'itération et la condition est double : la condition permet de contrôler la répétition du corps de l'itération, mais par ailleurs celui-ci doit permettre la mise à jour des valeurs des variables de la condition, afin d'assurer que l'exécution se termine dans un temps fini. Il faut aussi que cette condition puisse être évaluée dès la première rencontre de l'itération, et par suite, que ces variables aient une valeur initiale pertinente. C'est le problème de l'**initialisation** (il ne concerne pas seulement les variables de la condition, mais aussi éventuellement certaines variables qui apparaissent dans le traitement de l'information dans le corps de l'itération). Les choix qui seront faits ici ne sont pas purement linguistiques (Soloway et al 1982). Comme nous l'avons souligné, du point de vue algorithmique, les structures TANT QUE, REPETER, et ITERER ne sont pas identiques : place et gestion de la condition, particularités de traitement et organisation du corps de l'itération.

Pour ces raisons nous aurons à les distinguer dans l'étude des processus de conception d'algorithmes et d'apprentissage de la programmation. De plus différentes recherches (Soloway, 1982 ; Rouchier, Samurçay, 1984) ont étudié les différences conceptuelles des boucles WHILE, REPEAT et FOR. Elles ont conclu en particulier que la boucle TANT QUE est plus difficile à gérer que la boucle REPETER. En effet, la boucle TANT QUE nécessite l'anticipation de la condition d'arrêt de la boucle avant la construction des instructions du corps de la boucle. Cette anticipation ne

correspond pas aux usages spontanés d'une répétition. Les travaux de Soloway signalent aussi que les débutants rencontrent de très nombreuses difficultés dans la manipulation de la boucle FOR. (Soloway-Erich, 1982 b).

C'est la gestion de ces différents aspects de l'itération par les élèves qui sera analysée dans les situations expérimentales qui ont été construites.

VI. METHODES EXPERIMENTALES

Les méthodes expérimentales dépendent évidemment de l'approche dans laquelle on envisage les processus d'acquisition, de fonctionnement et d'évolution des connaissances.

Rappelons que nous nous situons dans une perspective constructiviste de constitution de connaissances (cf p.4). Dans le cadre de cette approche, la méthode expérimentale choisie a consisté à placer les élèves dans des situations problèmes dont la solution requiert la construction et l'utilisation d'une itération. Ces situations sont construites de façon à permettre l'explicitation des conceptions initiales des élèves, à propos de la notion étudiée, puis l'évolution de ces conceptions à travers les difficultés rencontrées tout au long de la résolution de ces problèmes.

Il ne suffit pas de placer les élèves dans une situation problématique, pour laquelle ils ne possèdent pas encore d'instruments de solution, pour que leur démarche de résolution débouche nécessairement sur la construction attendue. Le choix du problème, la distance de la solution à construire aux

connaissances disponibles chez l'élève, le choix du contexte sont autant d'éléments à déterminer pour assurer des conditions favorables à la micro-génèse envisagée.

Les notions mathématiques impliquées dans les problèmes posés sont bien connues des élèves (addition, multiplication), de façon à ce que la résolution du problème itératif ne soit pas perturbée par des difficultés d'ordre mathématique. Le problème pour les élèves porte donc sur l'écriture de l'algorithme et non sur le contenu mathématique ; c'est la planification par un algorithme d'une solution d'un problème arithmétique qu'ils connaissent déjà.

D'autre part, le choix du problème posé contribue à placer les élèves dans des conditions favorables à la mise en place d'une itération. Cette notion

constitue un instrument efficace ou même indispensable de solution des problèmes choisis : problèmes où la répétition a lieu, soit pour un grand nombre de fois, soit pour un nombre de fois inconnu a priori d'une ou plusieurs actions.

Une caractéristique des situations expérimentales adoptées favorise l'évolution des conceptions. Les élèves travaillent en binômes ; cette forme de travail présente l'intérêt de fournir un matériau d'étude constitué par les extériorisations des démarches de pensée au cours des échanges verbaux entre les partenaires. Mais elle est surtout susceptible de provoquer une évolution dans la résolution du problème grâce aux désaccords entre les élèves et aux conflits qui peuvent faciliter une prise de conscience pour chacun des partenaires de l'existence d'autres points de vue, et les amener

par là à une distanciation par rapport à leur propre production. Le dépassement d'un conflit peut dans certains cas conduire à une analyse nouvelle du problème, à une restructuration des connaissances en coordonnant à un niveau supérieur les deux points de vue contradictoires (Doise et Mugny, 1981 ; Perret-Clermont, 1979).

Dans une des expériences menées (chapitre III), une autre caractéristique participe aussi à l'évolution des conceptions des élèves : l'interaction avec l'expérimentateur.

Les dispositifs expérimentaux adoptés correspondent, comme nous l'avons déjà dit, à trois contextes différents de programmation :

-le chapitre II est consacré à une expérience réalisée en classe où les élèves ont à construire un algorithme itératif en langue naturelle destiné à un autre élève.

-le chapitre III est consacré à une expérience réalisée en dehors de la classe avec des élèves de 6e, 5e et 4e travaillant en binômes pour construire un programme avec itération dans un langage artificiel créé pour communiquer avec un robot virtuel.

- le chapitre IV est consacré à une expérience en classe de 3e où les élèves ont à effectuer un programme avec itération en BASIC pour un micro-ordinateur THOMSON TO7.

-ces expériences sont précédées (chapitre I) d'une enquête dont l'objectif est d'étudier la compréhension des expressions alternatives et répétitives par les élèves.

De la comparaison des observations de trois situations expérimentales, nous essayons de dégager dans la conclusion du travail des aspects de la

construction de l'itération par les élèves communs aux trois expérimentations. Nous cherchons à déterminer aussi le rôle du contexte de programmation sur l'élaboration de l'algorithme, en analysant les variations observées d'une expérimentation à l'autre dans les productions des élèves.

CHAPITRE I

UNE ENQUETE SUR LA COMPREHENSION DES EXPRESSIONS ALTERNATIVES ET REPETITIVES

I- Présentation de l'enquête

I.1 Tâche

I.2 Choix

II- Questionnaire n°1

II.1 Le questionnaire

**II.2 Commentaires à propos des choix des exercices
posés et des solutions proposées**

II.3 Résultats et interprétations (tableaux)

III- Questionnaire n°2

III.1 Le questionnaire

**III.2 Commentaires à propos des choix des exercices
posés et des solutions proposées**

III.3 Résultats et interprétations (tableaux)

IV- Conclusion

Dans ce chapitre, nous allons présenter les résultats d'une enquête que nous avons donnée, sous forme de questionnaire, à des élèves de collège (6ème, 5ème, 4ème). Cette enquête a pour objectif de chercher à savoir si les élèves comprennent des expressions qui indiquent des actions alternatives et répétitives, et s'ils peuvent distinguer et mettre en relation les différentes parties composant ces expressions : indicateurs de répétition ou de choix, condition de réalisation, actions à réaliser. Ces élèves n'avaient jamais reçu d'enseignement en informatique.

Dans les questionnaires, nous avons choisi des problèmes qui mettent en jeu des notions mathématiques simples adaptées aux connaissances des élèves dont les difficultés ne se trouvent pas dans la recherche de la solution mais dans l'expression de celle-ci.

I- PRESENTATION DE L'ENQUETE

I.1- Tâche

Nous énonçons un problème dont la solution consiste en l'écriture d'un algorithme. Nous proposons plusieurs solutions et nous demandons aux élèves de choisir la ou les solutions qu'ils estiment correctes et de justifier leurs choix.

I.2- Choix

Nous avons introduit des expressions classiques qui permettent d'obtenir la solution d'un problème exprimant des actions répétitives ou alternatives. Ces expressions sont :

Les alternatives :

SI condition ALORS action(s)
SINON action(s)

Les répétitives :

TANT QUE condition FAIRE action(s)
REPETER action(s) JUSQU'A CE QUE condition

Lorsqu'un élève rencontre un problème de ce type ses connaissances langagières devraient lui permettre de comprendre ces expressions, et de différencier les différentes parties de chaque expression. Ces parties sont :

Pour l'expression alternative :

- le mot SI qui indique qu'un choix va s'effectuer,
- la condition qui est une expression logique (qui prendra les valeurs "vrai" ou "faux").

- les mots ALORS et SINON qui indiquent quel choix réaliser en fonction du fait que la condition est vraie (ALORS) ou fausse (SINON).

- la ou les action(s) qui doivent s'effectuer dans le premier cas (ALORS) si la condition est satisfaite, et dans le deuxième cas (SINON) si la condition n'est pas satisfaite.

Pour les expressions répétitives :

- l'expression TANT QUE FAIRE
 - * les mots TANT QUE annoncent qu'une répétition va se produire
 - * le mot FAIRE indique les actions à réaliser si la condition est vraie

- l'expression REPETER JUSQU'A CE QUE
 - * le mot REPETER annonce qu'une répétition des actions indiquées à la suite de ce mot va se produire.
 - * les mots JUSQU'A CE QUE délimitent les actions et annoncent la condition.

- la condition ou la (ou les) action(s) ont le même sens que pour l'expression alternative.

Nous avons aussi choisi d'autres types non classiques d'expressions alternatives :

- QUAND condition ALORS action(s) SINON action(s). Le choix du mot QUAND réside dans le fait que c'est un mot fréquemment employé dans le langage courant et qu'il peut amener à des confusions entre les expressions alternatives et répétitives. Nous cherchons à connaître l'interprétation que les élèves donnent à des expressions écrites contenant ce mot.

- de la même forme que l'expression REPETER action(s) JUSQU'A CE QUE condition sans mentionner l'avertisseur "REPETER".

Nous voulions déterminer si la présence de l'expression de la condition par JUSQU'A CE QUE conduisait les élèves à interpréter l'expression comme une expression répétitive, même en absence d'avertisseur. Dans les usages courants la valeur répétitive d'une action peut en effet rester implicite, dans la mesure où la mention de JUSQU'A CE QUE n'est justifiée que s'il existe une action à répéter.

II -QUESTIONNAIRE N° 1

II.1 Le questionnaire

Nous avons distribué aux élèves le questionnaire suivant :

Exercice n° 1 :

On a un robot qui sait marcher qui comprend la voix humaine et qui sait monter les escaliers, mais quand on lui donne l'ordre "monter" il ne monte qu'une marche.

Lequel ou lesquels des ordres suivants permettent au robot de monter tout l'escalier ?

Au départ le robot est sur la première marche.

1 -TANT QUE tu n'es pas au bout de l'escalier
FAIRE : "monter"

2 -SI tu es dans l'escalier
ALORS "monter"

3 - QUAND tu es dans l'escalier, "monter".

4 - "Monter" JUSQU'A CE QUE tu sois au bout de l'escalier.

- dans chacun des cas, combien de marches a monté le robot ?
- mets le ou les numéros des ordres que tu as choisis dans la case
- Explique pourquoi tu les as choisis.

EXERCICE N°2 :

Un sac contient des boules rouges et des boules bleues. Tu donnes à un camarade des instructions pour qu'il t'écrive la couleur de la première boule qu'il tire. Parmi les instructions suivantes choisis celles (ou celle) qui conviennent.

1 - Prendre une boule JUSQU'A CE QU' on tire une boule rouge.

2 - TANT QUE la boule n'est pas rouge, FAIRE : prendre une boule.

3 - Prendre une boule.

Si la boule est rouge

ALORS écrire : "la boule est rouge"

SINON écrire : "la boule est bleue".

4 - Prendre une boule. QUAND la boule est rouge, écrire :

"la boule est rouge" SINON écrire : "la boule est bleue".

- Mets le ou les numéros des instructions que tu as choisies dans la case.

- Explique pourquoi tu les as choisies.

EXERCICE N° 3

On donne à un élève des instructions pour construire un carré. Parmi les instructions suivantes, laquelle ou lesquelles permettent à l'élève de construire correctement le carré.

1 - Trace un segment de 5 cm

Si tu n'es pas sur le point de départ

ALORS tourne à droite de 90°

trace un segment de 5 cm.

2-Trace un segment de 5 cm et tourne à droite de 90° JUSQU'A CE QUE tu arrives au point de départ du premier segment.

3-Trace un segment de 5 cm. QUAND tu n'es pas au point de départ du premier segment, tourne à droite de 90° et trace un segment de 5 cm.

4 - Trace un segment de 5 cm

TANT QUE tu n'arrives pas au point de départ du premier segment

FAIRE : tourne à droite de 90°

trace un segment de 5 cm.

- Mets le ou les numéros des instructions que tu as choisies dans la case.

- Explique pourquoi tu les as choisies.

EXERCICE N°4 :

Dans un sac il y a des boules rouges et des boules bleues. On souhaite prendre des boules une par une et on décide de s'arrêter quand on tire la première boule rouge. Parmi les ordres suivants choisis ceux (ou celui) qui te permettent toujours d'arriver à la première boule rouge et de t'arrêter quand tu l'as tirée.

- 1 - Prendre une boule. QUAND la boule est rouge, s'arrêter.
- 2 - Prendre une boule JUSQU'A CE QU'on tire une boule rouge.
- 3- Prendre une boule. QUAND la boule est bleue, recommencer au début.
- 4 - Prendre une boule.
TANT QUE la boule est bleue, FAIRE : prendre une autre boule.
- 5 - Prendre une boule. SI la boule est bleue, ALORS s'arrêter.
- 6 - Prendre une boule.
TANT QUE la boule est rouge, FAIRE : prendre une autre boule.

- Mets le ou les numéros des ordres que tu as choisis, dans la case.
- Explique pourquoi tu les as choisis.

II.2 Commentaires à propos des choix des exercices posés et des solutions proposées

Ce premier questionnaire consiste en 4 exercices pour lesquels nous donnons quatre réponses possibles pour les trois premiers et six pour le dernier. Les réponses des exercices 1,3 et 4 mettent en jeu une expression répétitive. La réponse de l'exercice 2 met en jeu une expression alternative. Parmi ces réponses l'élève devra choisir la ou les réponses qu'il considère correctes. A la fin de chaque exercice nous demandons aux élèves d'expliquer leurs choix, ce qui peut éventuellement nous apporter des informations supplémentaires.

Exercice n° 1 :

L'objectif de cet exercice est de chercher à savoir si les élèves comprennent une expression répétitive pour exprimer la solution d'un problème qui nécessite ce type de solution.

Les réponses correctes sont les n° 1 et n°4. La n°1 est une expression du

type TANT QUE FAIRE où la condition d'arrêt porte sur l'arrivée du robot

au bout de l'escalier et l'action est celle de monter. La réponse n°4 désigne aussi l'exécution d'une action : "monter" qui doit se répéter jusqu'à la vérification de la condition : "que tu sois au bout de l'escalier".

Les réponses 2 et 3 (incorrectes) n'indiquent pas une répétition. Dans les deux cas l'action de monter s'exécute une seule fois, néanmoins nous l'avons incluse comme solution possible parce que nous nous attendions à ce que les élèves utilisent des expressions alternatives pour exprimer une action de répétition (Confusion entre QUAND et TANT QUE ou entre SI et TANT QUE).

Ici, nous demandons aussi d'écrire dans chaque cas combien de marches le robot a monté. Cette question nous permettra de vérifier s'il y a bien adéquation entre la réponse à la question posée et le nombre indiqué des marches montées.

Exercice n° 2 :

Il s'agit dans cet exercice de savoir si les élèves comprennent une expression alternative pour exprimer la solution à un problème qui nécessite une solution de ce type.

Les réponses correctes sont les n°3 et n°4. La réponse n°3 est une expression alternative du type SI ALORS SINON, la réponse n°4 est une expression alternative du type QUAND ALORS SINON pour lesquelles il existe une seule action à effectuer qui dépend de la vérification ou non de la condition.

Les réponses 1 et 2 (incorrectes) expriment une action répétitive. Il se peut que les élèves interprètent ces expressions comme alternatives ou qu'ils considèrent que la solution au problème nécessite une répétition d'actions .

Exercice n°3 :

L'objectif de cet exercice est semblable à celui de l'exercice n°1. Cependant nous avons introduit une variante dans le nombre d'actions : au lieu d'une seule action comme dans l'exercice précédent, on a posé un problème qui nécessite la réalisation d'une suite de deux actions : tracer un segment de 5 cm et tourner à droite de 90°. Nous cherchons à savoir si l'introduction de cette nouvelle variante peut influencer sur les choix réalisés par les élèves.

Les réponses correctes sont les n°2 et n°4. La réponse n°2 est une expression du type : action JUSQU'A CE QUE condition . L'action est une action composée de deux actions, la condition d'arrêt est : "que tu arrives au point de départ du premier segment". La réponse n°4 est une expression répétitive du type TANT QUE avec la même condition d'arrêt et la même action composée.

De la même manière que dans l'exercice 1 nous avons inclu des réponses incorrectes (la n°1 et la n°3) pour déterminer si l'introduction de cette nouvelle variante est susceptible d'augmenter la confusion entre expression alternative et expression répétitive.

Exercice n°4 :

Le but de cet exercice est le même que celui des exercices 1 et 3 mais on a introduit une nouvelle variante : des expressions avec indicateurs de répétitions similaires mais avec des conditions d'arrêt différentes, ceci pour voir le rôle que joue la condition dans le choix des élèves.

Les réponses correctes à cet exercice sont la réponse n°2 et la réponse n°4. La réponse n°2 est une expression du type répétitive dans laquelle l'action : "prendre une boule" devra s'exécuter JUSQU'A CE QUE soit vérifiée la condition : on tire une boule rouge. La réponse n°4 est une expression du type TANT QUE où la condition : la boule est bleue doit être vraie pour exécuter l'action de prendre des boules.

Les considérations à propos de choix sous-jacents aux réponses incorrectes sont les mêmes que pour les exercices 1 et 3.

II.3 Résultats et interprétations (tableaux)

Nous avons fait passer ce questionnaire à 50 élèves répartis en deux classes : 4ème (26 élèves) et 6ème (24 élèves). Les défauts d'interprétation que nous attendions ne se sont pas révélés aussi importants que prévu.

TABLEAU 1 : Exercice N°1

Classe N° Réponse	4e	6e	Total
Sans réponse		1	1
1"tant que"	3		3
2"si alors"	1		1
3"quand"			
4"jusqu'à ce que"	1		1
1,4	17	19	36
2,4	1		1
1,2,3	1		1
1,2,4		1	1
1,2,3,4	2	3	5
TOTAL	26	24	

Réponses correctes : 1 et 4

Sans réponse : 1
 Réponse n°1 : 46
 Réponse n°2 : 9
 Réponse n°3 : 6
 Réponse n°4 : 44

Les résultats présentés dans ce tableau montrent que les élèves n'ont pas eu de difficultés pour interpréter une expression répétitive comme solution de cet exercice.

Nous pouvons penser que le choix des réponses correctes est dû au fait que la phrase "au bout de l'escalier" apparaît dans les expressions répétitives et parce que le mot "bout" évoque pour eux "avoir terminé de monter l'escalier".

De plus dans cet exercice on demande d'écrire dans chacune des réponses le nombre de marches que le robot monte. Dans la plupart des cas, les élèves ont indiqué le nombre correct de marches dans toutes les solutions proposées ce qui nous permet encore d'affirmer que les expressions TANT QUE FAIRE et JUSQU'A CE QUE sont comprises comme expression indiquant une boucle de même que les expressions conditionnelles SI ALORS SINON et QUAND ALORS SINON ne sont pas comprises comme des expressions répétitives.

Les confusions que nous nous attendions à trouver entre les expressions répétitives et alternatives ne se sont pas produites dans cet exercice.

TABLEAU 2 : Exercice N°2 :

Classe N° Réponse	4e	6e	Total
Sans réponse	1		1
1"jusqu'à ce que"			
2"tant que"		1	1
3"si alors"	5	3	8
4"quand"		2	2
1,2	1	2	3
2,3	1		1
2,4	1		1
3,4	17	14	31
1,2,4		1	1
1,2,3,4		1	1
TOTAL	26	24	

Réponses correctes : 3 et 4

Sans réponse : 1
 Réponse n°1 : 5
 Réponse n°2 : 8
 Réponse n°3 : 41
 Réponse n°4 : 36

Les résultats de ce tableau montrent que les élèves n'ont pas eu de difficultés à comprendre une expression de type alternative comme solution à ce problème . L'expression alternative du type SI condition ALORS action(s) SINON action(s) a été choisie plus souvent que l'expression du type QUAND ALORS SINON.

Comme nous le supposions a priori, un élève a écarté la réponse contenant l'expression avec QUAND parce que pour lui le mot "QUAND" indique l'idée de répétition :

"car on met si la boule est rouge cela veut dire que c'est une supposition alors que au 4 on met QUAND la boule est rouge cela veut dire qu'on tire jusqu'à ce que la boule soit rouge"

TABLEAU 3 : Exercice N°3

Classe N° Réponse	4e	6e	Total
Sans réponse	3	2	5
1"si alors"		1	1
2"jusqu'à ce que"		1	1
3"quand"		1	1
4"tant que"	8	7	15
1,3		1	1
1,4	1	1	2
2,3		2	2
2,4	8	3	11
3,4	1	3	4
1,2,3		1	1
1,2,4	1		1
1,3,4	2	1	3
2,3,4	1		1
1,2,3,4	1		1
TOTAL	26	24	

Réponses correctes : 2 et 4

Sans réponse: 5
 Réponse n°1: 10
 Réponse n°2: 18
 Réponse n°3: 14
 Réponse n°4: 38

Dans cet exercice, les réponses correctes 2 et 4 ont été les plus choisies, néanmoins les réponses sont diverses.

23 élèves (ce qui n'est pas négligeable) ont fourni une réponse erronée en choisissant une expression alternative (10 avec SI ALORS et 13 avec QUAND) pour exprimer une action de répétition. Il semblerait que les élèves se soient surtout attachés à la présence de la condition d'arrêt pour choisir les réponses, et que le fait que la répétition porte sur deux actes en séquence et non sur une seule n'ait pas joué un rôle perturbateur.

TABLEAU 4: Exercice N°4

Classe	4e	6e	Total
N° Réponse			
Sans réponse		3	3
1"quand"	2		2
2"jusqu'a ce que"		1	1
3"quand"			
4"tant que bleu"			
5"si alors"			
6"tant que rouge"			
1,2	1	6	7
1,3		1	1
2,3	1		1
2,4	1	1	2
5,6		2	2
1,2,4	14	5	19
1,3,4		1	1
1,2,3,4	4	1	5
1,2,4,6	1		1
1,2,3,4,5,6	1		1
TOTAL	26	24	

Réponses correctes : 2 et 4

Sans réponse: 3
 Réponse n°1 :37
 Réponse n°2 :35
 Réponse n°3 : 9
 Réponse n°4 :29
 Réponse n°5 : 3
 Réponse n°6 : 4

C'est cet exercice qui a posé le plus de difficultés aux élèves, même si les réponses correctes ont été choisies en grand nombre. En effet, la réponse 1, qui est erronée a été la plus choisie et le plus souvent en même temps que les 2 réponses correctes. On peut penser que les élèves ont fixé leur attention sur la condition d'arrêt : "la boule rouge" et sur l'action : "s'arrêter", plutôt que sur les indicateurs de l'action répétitive ou de l'action alternative. Nous pouvons donc parler d'une relation correcte entre la condition et l'action : "boule rouge" - "s'arrêter".

La réponse 5 a été la moins choisie parce qu'elle combine "boule bleue" et "s'arrêter" ce qui ne correspond pas à une relation correcte condition-action. Les réponses les plus choisies sont 1,2 et 4 qui contiennent des relations correctes entre la condition et les actions soit "boule rouge"et "s'arrêter" soit "boule bleue" et "continuer".

Dans ce premier questionnaire, nous n'avons pas pu voir le rôle que joue la complexité du corps de la boucle dans la compréhension de l'expression répétitive. Pour cette raison nous nous sommes proposés de donner aux élèves un deuxième questionnaire qui fait intervenir un corps avec plusieurs actions.

III-QUESTIONNAIRE N° 2

III.1 Le questionnaire

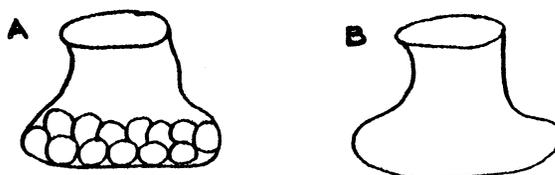
Nous avons distribué le questionnaire suivant :

EXERCICE N° 1 :

On dispose de 2 sacs, le sac A et le sac B, comme le montre le dessin ci-dessous. Le sac A contient des billes qu'on veut faire passer dans le sac B.

On donne des ordres à un robot pour qu'il le fasse. Le robot ne peut prendre qu'une bille à la fois.

Parmi les ordres suivants, choisis celui ou ceux qui permettent d'effectuer le transvasement.



- 1 - TANT QUE le sac A n'est pas vide
FAIRE : Prendre une bille dans le sac A
et la mettre dans le sac B.
- 2 - Prendre une bille dans le sac A et la mettre dans le sac B.
Recommencer.
- 3 - Si le sac A n'est pas vide
ALORS prendre une bille dans le sac A
et la mettre dans le sac B.
- 4 - Prendre une bille dans le sac A et la mettre dans le sac B
JUSQU'A CE QUE le sac A soit vide.
- 5 - QUAND le sac A n'est pas vide, prendre une bille dans le sac A
et la mettre dans le sac B.

- Mets le ou les numéros des ordres que tu as choisis dans la case

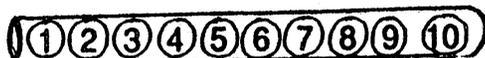
- Explique pourquoi tu les as choisis.

EXERCICE N° 2 :

On dispose d'un tube dans lequel il y a des billes numérotées de 1 à 10, rangées comme dans le dessin ci-dessous. On ne peut les sortir qu'une par une.

Une personne a effectué l'addition de certains nombres tracés sur les billes de ce tube; le résultat qu'elle a obtenu est 10.

Quelle ou quelles instructions la personne a-t-elle suivies pour obtenir ce résultat ?



1 - Prendre une bille.

Ecrire son numéro sur une ardoise.

TANT QUE le numéro de la bille n'est pas 7.

FAIRE : prendre une autre bille

ajouter son numéro au nombre écrit sur l'ardoise.

effacer le nombre écrit.

écrire à la place la somme obtenue.

2 - Prendre une bille.

Ecrire son numéro sur une ardoise.

SI le numéro de la bille n'est pas 5.

ALORS prendre une autre bille

ajouter son numéro au nombre écrit sur l'ardoise.

effacer le nombre écrit.

écrire à la place la somme obtenue.

3 - Prendre une bille.

Ecrire son numéro sur une ardoise.

Prendre une autre bille.

Ajouter son numéro au nombre écrit sur l'ardoise.

Effacer le nombre écrit.

Ecrire à la place la somme obtenue.

4 - Prendre une bille.

Ecrire son numéro sur une ardoise.

TANT QUE le numéro de la bille n'est pas 5.

FAIRE : prendre une autre bille.

ajouter son numéro au nombre écrit sur l'ardoise.

effacer le nombre écrit.

écrire à la place la somme obtenue.

5 - Prendre une bille.

Ecrire son numéro sur une ardoise.

FAIRE : prendre une autre bille

ajouter son numéro au nombre écrit sur l'ardoise.

effacer le nombre écrit.

écrire à la place la somme obtenue.

JUSQU'A CE QUE le tube soit vide.

- Quelle est la liste des nombres qui ont été écrits sur l'ardoise dans chacun des cas ?

- 1
- 2
- 3
- 4
- 5

- Mets le ou les numéros des instructions que tu as choisies dans la case.

- Explique pourquoi tu les as choisies.

III.2-Commentaires à propos des choix des exercices posés et des solutions proposées

Ce deuxième questionnaire est composé de deux exercices avec 5 réponses possibles pour chacun, les élèves doivent choisir la ou les réponses correctes de chaque exercice.

Exercice n° 1 :

Dans cet exercice, comme dans les exercices précédents, on cherche à savoir si les élèves choisissent des expressions répétitives ou alternatives pour exprimer une action de répétition. De plus dans cet exercice nous cherchons à savoir si, pour les élèves, l'arrêt est implicitement effectué dans le cas où un algorithme ne contient pas de condition d'arrêt : réponse n° 2 .

Les réponses correctes dans cet exercice sont les réponses n° 1 et 4. La réponse n° 1 est une expression répétitive construite avec **TANT QUE FAIRE** ; la condition : "le sac n'est pas vide" permet la continuation de l'exécution des actions : "prendre une bille dans le sac A et la mettre dans le sac B". La réponse n°4 nous donne les mêmes actions à faire exécuter jusqu'à ce que se produise la vérification de la condition : "sac vide".

Les réponses 3 et 5 contiennent des expressions alternatives. Nous nous attendions à ce que "QUAND" ait une valeur de répétition pour certains élèves.

Exercice n° 2 :

Dans cet exercice, en plus des objectifs visés dans l'exercice précédent, nous avons voulu poser un problème nécessitant une réponse qui prend en compte la relation entre :

- la vérification d'une condition d'arrêt,
- et l'exécution d'un corps de boucle composé de plusieurs actions.

Nous obtenons des renseignements supplémentaires grâce aux réponses données à la question : "quelle est la liste des nombres qui ont été écrits sur l'ardoise dans chacun des cas". La seule réponse correcte dans cet exercice est la réponse n°4 qui prend en compte à la fois la condition : arriver au nombre 5 et le résultat de l'addition (somme = 10).

Dans la réponse 1, la condition est " le numéro de la bille n'est pas 7", lorsque la condition est vérifiée la somme est supérieure à 10. La réponse n°2 est une expression qui conduit à l'exécution des actions du corps de l'expression conditionnelle une seule fois (la condition est vérifiée), le résultat est l'addition des deux premiers nombres $1 + 2 = 3$. La réponse n°3 contient une liste d'instructions qui effectue l'addition des nombres tracés sur les deux premières billes, le résultat est différent de 10 et il n'y a pas l'ordre de continuer à exécuter les actions. Pour la réponse n°5 il faut sortir toutes les billes du tube donc la somme est supérieure à 10.

III.3- Résultats et interprétations (tableaux)

Nous avons donné ce questionnaire à 89 élèves répartis en 4 classes : 3 classes de 4ème et une de 5ème :

TABLEAU 5: Exercice N° 1

Classe N° Réponse	4e	4e	5e	4e	Total
Sans réponse	1	1	2		4
1"tant que"					
2"recommencer"			1		1
3"si alors"					
4"jusqu'à ce que"	1	4	13	3	21
5"quand"					
1,3			1		1
1,4	4	5	1	9	19
2,4	5	1	4	3	13
3,4		2		1	3
1,2,3		1			1
1,2,4	2	1	2	3	8
1,3,4	1		1		2
1,4,5	1		1	1	3
2,4,5	1				1
1,2,3,4		1			1
1,3,4,5	2	1		1	4
2,3,4,5	1				1
1,2,3,4,5	2	2			4
TOTAL	21	19	26	23	

Réponses correctes:1 et 4

Sans réponse : 4
 Réponse n°1 : 43
 Réponse n°2 : 30
 Réponse n°3 : 17
 Réponse n°4 : 80
 Réponse n°5 : 13

Les réponses correctes 1 et 4 ont été les plus choisies mais notons que la réponse 4 a été plus choisie que la réponse 1. On peut donc en déduire que pour cet exercice l'expression avec JUSQU'A CE QUE exprime mieux

l'action répétitive pour les élèves.

Un nombre important d'élèves a choisi une expression alternative pour exprimer une action répétitive. 30 élèves ont choisi la solution au problème à

l'aide de la réponse 2 dans laquelle il y a l'ordre de recommencer mais la condition d'arrêt n'est pas indiquée. 13 élèves ont choisi l'expression QUAND pour exprimer une action répétitive.

Ces résultats nous conduisent à faire l'hypothèse que les expressions alternatives sont interprétées comme des actions répétitives et que, même s'il n'y a pas de condition d'arrêt dans une expression répétitive les élèves la vérifient implicitement.

Voici quelques exemples d'explications fournies par des élèves :

"les phrases 1,2,3,4,5 veulent dire la même chose mais exprimées différemment"

"dans la n°5, quand le robot voit qu'il reste des billes dans le sac, il continuera"

" la n°2 nous dit de recommencer et à force de faire le même travail le sac A se vide"

"j'ai choisi la n°2 parce que le robot prend une bille du sac A et la met dans le sac B et doit à chaque fois recommencer"

TABLEAU 6: Exercice N° 2

Classe N° Réponse	4e	4e	5e	4e	Total
Sans réponse	7	4	6	14	31
1"tant que n°7"		1			1
2" si alors"			1		1
3"liste intructions"	1	1	13		15
4"tant que n°5"	4	2			6
5"jusqu'à ce que tube vide"	5		2	1	8
1,2		2			2
1,4	1		1	1	3
2,4		5		3	8
3,4		2			2
3,5			1		1
4,5	1				1
1,3,5	1				1
2,3,4		1		3	4
1,2,3,4		1			1
1,2,3,4,5	1				1
TOTAL	21	19	26	23	

Réponse correcte : 4

Sans réponse : 31

Réponse n°1 : 9

Réponse n°2 : 17

Réponse n°3 : 25

Réponse n°4 : 22

Réponse n°5 : 13

Le premier résultat marquant réside dans l'absence de réponse d'élèves (31/89). Les réponses sont très diversifiées et nous ne pouvons donc pas dire qu'une réponse particulière ait été choisie par une majorité d'élèves.

On peut remarquer que les élèves ont choisi des expressions alternatives pour indiquer une répétition plus souvent que dans l'exercice précédent (17 élèves ont choisi l'expression avec SI ALORS SINON).

La solution correcte n°4 a été choisie par 22 élèves, et la réponse n°1 dont la seule différence avec la réponse n°4 consiste dans le contenu de la condition, a été choisie par 9 élèves. On peut penser que la vérification de la condition n'a pas toujours été faite par les élèves pour discriminer les réponses correctes. De même la réponse 5, qui indique une répétition mais dont la condition d'arrêt ne correspond pas à la solution correcte, a été choisie par 13 élèves pour simuler l'algorithme.

Notre deuxième objectif dans cet exercice était que les élèves simulent le fonctionnement de l'algorithme pour trouver le résultat correct, cette simulation était indispensable pour vérifier si la condition d'arrêt était remplie ou non. Nous pouvons dire que les élèves sont mis devant la difficulté de coordonner séquentiellement plusieurs actions.

La question dans laquelle nous demandions aux élèves d'écrire les nombres qui sont écrits sur l'ardoise n'a été abordée que par 9 élèves seulement. Les 80 autres élèves n'ont pas répondu à cette question. Cela nous amène à penser que l'exécution d'une action répétitive composée de plusieurs actions représentant une seule action composée (corps de la boucle) nécessite une manipulation et un entraînement important.

IV- CONCLUSION

Le premier questionnaire nous a permis de dégager certains aspects des difficultés que les élèves rencontrent à la lecture des expressions répétitives ou alternatives :

- 1° Si leur attention se porte sur le choix de la condition correcte les expressions alternatives et répétitives peuvent se confondre,
- 2° L'expression QUAND exprime dans certains cas une action répétitive,
- 3° Les indicateurs de répétition TANT QUE FAIRE , JUSQU'A CE QUE et FAIRE JUSQU'A CE QUE sont en général interprétés comme exprimant des actions de même type, mais ne sont pas choisis pour exprimer des actions alternatives. Au contraire, les expressions SI ALORS SINON et QUAND ALORS SINON peuvent être choisies dans certains cas pour exprimer une répétition.

Le deuxième questionnaire nous a permis de préciser quelques difficultés relatives à la compréhension des expressions répétitives et alternatives. Il permet d'ajouter que la compréhension d'une expression répétitive, dont le corps est une action composée de plusieurs actions présente plus de difficultés. De même l'exécution à la main de cette action composée en coordination avec une condition requise ne peut pas être réalisée une première fois par des élèves ne connaissant pas ce type d'expression.

Soulignons que les conclusions de cette enquête sont des interprétations des résultats d'une épreuve papier-crayon et que cette épreuve ne nous permet pas d'analyser les processus d'élaboration de ces résultats. Des

expériences différentes, qui seront exposées dans les chapitres suivants, ont été construites pour observer ces processus et approfondir nos interprétations.

CHAPITRE II

CONSTRUCTION D'UNE ITERATION DANS UNE SITUATION DE COMMUNICATION ELEVE-ELEVE

I- Les choix de l'expérience

II- Description et déroulement de l'expérience

III- Analyse de la situation

IV- Phase d'émission

IV. 1- Résultats

IV. 2- Conclusion

V- Phase de réception

V. 1- Résultats

V. 2- Les échanges

V. 3- Conclusion

VI- Phase de travail individuel

VI. 1- Résultats

VI. 2- Exemples d'évolution phase 1- phase 5

VI. 3- Conclusion

VII- Conclusion

Ce chapitre est consacré à une expérience située dans un contexte de communication homme-homme (élève ou groupe d'élèves vers élève ou groupe d'élèves).

Dans un premier temps, les élèves écrivent un message sous forme d'algorithme en langue naturelle destiné à d'autres élèves et devant leur permettre de résoudre une tâche donnée. Dans un deuxième temps, les élèves récepteurs interprètent l'algorithme reçu et le font fonctionner.

Nous allons nous intéresser d'une part à la construction d'un algorithme itératif permettant de résoudre un problème donné et à son expression par les élèves émetteurs, d'autre part à l'interprétation et au fonctionnement de cet algorithme par les élèves récepteurs.

I-LES CHOIX DE L'EXPERIENCE

Les choix expérimentaux fondés sur des choix théoriques, sont de différents ordres :

1- La communication s'établira entre groupes d'élèves (2 ou 3) en langue naturelle et d'une façon écrite ; nous avons choisi pour la construction du message, la langue naturelle car il s'agit d'un langage de communication sans contraintes spécifiques qui ne nécessite aucun apprentissage particulier et que les élèves manient relativement aisément. Le choix de la communication écrite exige des élèves une activité importante de planification (Hoc, 79) et de structuration de l'algorithme.

Une première partie de l'analyse portera sur les messages produits, elle concerne la structure et la forme de l'algorithme de calcul, et les relations entre les caractéristiques de la communication (élèves-élèves) et la nature du message produit. Une deuxième partie de l'analyse portera sur la réception de ces messages : elle étudiera l'interprétation du message par le groupe récepteur et sa mise en oeuvre dans l'exécution du message.

2- Le dispositif choisi est invoqué (dispositif virtuel). Ce choix permet d'éviter les manipulations matérielles et les erreurs qui leur sont attachées (fautes d'orthographe, faute de frappe..). Notre objectif d'étude est en effet centré sur la construction de l'algorithme de calcul, et ne concerne pas (du moins pour l'instant) les difficultés d'apprentissage dans l'utilisation réelle d'une machine.

3-Le problème choisi pour cette expérience est un problème de type numérique mettant en jeu une opération connue des élèves : la multiplication et sa réalisation par répétition d'additions. Le problème réside donc pour les élèves dans l'écriture de l'algorithme.

4-Le problème est choisi de façon à ce que l'itération en constitue une solution efficace et adaptée ; il s'agit en effet de calculer le produit de 2 nombres quelconques en n'utilisant que l'opération d'addition.

L'indétermination des nombres dont il faut calculer le produit requiert l'utilisation d'une itération car elle empêche de résoudre le problème par une suite déterminée a priori d'additions.

II-DESCRIPTION ET DEROULEMENT DE L'EXPERIENCE

Nous avons travaillé avec une classe de seconde du Lycée Benjamin Franklin à Orléans, comportant 32 élèves qui ont été partagés en 16 groupes de 2 élèves. **Ces élèves n'avaient pas reçu d'enseignement de l'informatique à l'école.**

L'expérience a consisté à demander aux élèves d'écrire un algorithme de calcul du produit ou du double du produit de deux entiers positifs quelconques. Cet algorithme est écrit sous la forme d'un message d'un groupe d'élèves à un autre groupe d'élèves. L'expérience se déroule en 5 phases :

1ère phase : Présentation du dispositif et de ses caractéristiques.

Nous avons distribué aux élèves le texte suivant :

*"On dispose de deux machines A et B et d'un opérateur qui les utilise.
Les spécifications des machines A et B sont les suivantes :*

Machine A : les seules touches qui fonctionnent sont :

- les touches des chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9*
- la touche + qui permet d'effectuer les additions*
- la touche = qui permet d'afficher le résultat*

Machine B : les seules touches qui fonctionnent sont :

- les touches des chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9*
- la touche +1 qui permet d'ajouter 1 au nombre affiché*
- la touche = qui permet d'afficher le résultat*

Les compétences de l'opérateur sont les suivantes :

Il ne sait que lire le message, exécuter les ordres qu'il contient, manipuler les machines A et B et lire un résultat."

L'enseignant a ajouté oralement que l'opérateur ne savait pas compter. Après avoir donné par écrit les caractéristiques du dispositif, nous les avons expliquées au tableau pour nous assurer de la compréhension, par les élèves, de la manipulation du matériel.

2ème phase.

Nous avons posé aux élèves, les problèmes P1 et P2, qui ont été distribués par écrit avec la consigne suivante :

Pour le problème (P1)

"Vous avez à écrire un message en termes d'opérations permises sur les machines A et B.

Ce message doit permettre à l'opérateur de :

CALCULER LE PRODUIT DE DEUX NOMBRES ENTIERS QUELCONQUES à l'aide des machines A et B.

Ce message sera transmis à un groupe récepteur. Quand le groupe récepteur recevra votre message, on lui donnera deux nombres entiers et il devra suivre les consignes du message, en tant qu'opérateur, pour trouver un résultat"

Pour le problème (P2), la consigne était la même, à la différence du problème qui était :

CALCULER LE DOUBLE PRODUIT DE DEUX NOMBRES ENTIERS QUELCONQUES

Parmi les 16 groupes constitués, les groupes impairs ont reçu la consigne relative au problème P1 et les groupes pairs ont reçu la consigne relative au problème P2.

L'enseignant s'est assuré auprès des groupes qu'il n'y avait pas de problèmes d'incompréhension de la consigne.

3ème phase .

Echange de messages. Les groupes sont devenus récepteurs des messages produits dans la phase précédente.

Nous avons distribué, aux groupes récepteurs, le texte suivant :

"Vous recevez un message que vous devez exécuter en prenant le rôle de l'opérateur et en vous servant des machines A et B".

L'enseignant va vous donner deux nombres entiers qui vont vous permettre de réaliser fidèlement les opérations inscrites dans le message.

Il vous est demandé d'écrire ces opérations au fur et à mesure.

Enfin, s'il ne vous est pas possible de suivre les consignes du message, on vous demande de noter les raisons qui vous en empêchent.

4ème phase : Discussion des messages.

L'enseignant choisit un algorithme "correct" pour l'expliquer et l'analyser avec les élèves. Il choisit ensuite un ou deux algorithmes "non-corrects" pour analyser les raisons de leur non-fonctionnement et pour mettre en évidence certains aspects des messages erronés.

5ème phase : Résolution individuelle d'un nouveau problème P3.

Après avoir analysé (enseignant et élèves) le fonctionnement et le non-fonctionnement des messages, nous avons donné aux élèves un troisième problème dans lequel il leur est demandé d'écrire un message qui permette le calcul de x puissance n , x et n étant deux entiers positifs quelconques.

Pour la réalisation de ce problème nous avons introduit une nouvelle catégorie de machines, la machine A' avec les caractéristiques suivantes :

Machine A' : Les seules touches qui fonctionnent sont :

- les touches des chiffres 0,1,2,3,4,5,6,7,8,9*
- la touche x qui permet d'effectuer les produits*
- la touche = qui permet d'afficher les résultats.*

Nous avons constitué un nouveau dispositif composé des machines A, A' et B et d'un opérateur qui les manipule.

Nous avons donné aux élèves des indications écrites relatives aux caractéristiques des trois machines et aux compétences de l'opérateur accompagnées de la consigne suivante :

"Vous avez comme dans l'exercice précédent à écrire un message destiné à un récepteur. Ce récepteur dispose d'autant de machines A, A' et B qu'il est nécessaire. Le message doit lui permettre, en utilisant comme opérateur les machines qui vous paraissent nécessaires de

CALCULER LA VALEUR DE x PUISSANCE n , x et n étant deux entiers positifs quelconques".

Cette phase a été réalisée par chaque élève individuellement. Il n'y a pas eu d'échanges.

III-ANALYSE DE LA SITUATION

Les analyses réalisées dans cette expérience sont fondées uniquement sur les messages écrits par les élèves parce que nous n'avons pas observé les binômes en train de travailler. Une telle observation est faite à propos d'une autre expérience exposée dans le chapitre suivant.

III.1- Phase d'émission des messages

L'itération en tant que solution du problème

Nous avons placé les élèves dans une situation comportant un problème : l'élaboration d'un algorithme de calcul du produit de deux nombres. L'itération constitue une solution optimale à ce problème du fait :

1 - des contraintes du dispositif : la machine A ne permettant que l'addition oblige à réaliser la multiplication comme une répétition d'additions. La machine B sert de compteur puisqu'elle permet d'ajouter 1 à chaque fois et permet l'arrêt de la répétition.

2 - de l'indétermination des nombres à multiplier : la consigne, que les élèves reçoivent, spécifie qu'ils doivent calculer le produit de deux entiers quelconques. L'algorithme solution ne peut être la juxtaposition d'un nombre déterminé d'additions. L'itération permet de construire un algorithme valide pour n'importe quels nombres d'entrée.

La solution du problème correspond à des itérations du type :

1 - REPETER

(certaines actions)
JUSQU'A CE QUE (condition)

2 - TANT QUE (condition) FAIRE (certaines actions)

Nous nous attendons à trouver des constructions d'itération du type REPETER (...) JUSQU'A CE QUE (...) plutôt que du type TANT QUE (.....) FAIRE (.....). En effet, l'itération du type TANT QUE (...) FAIRE (...) témoigne d'une anticipation de la condition d'arrêt qui ne correspond pas au comportement spontané des élèves (Rouchier, Samurçay 1984 ; Rogalsky 1985 ; Soloway 1982).

"L'inversion des opérations d'action et de test pour la structure de répétition "tant que /condition/ faire /actions" exige de mettre en cause le modèle spontané, on doit donc rencontrer des obstacles importants dans son acquisition " (Rogalski 1985 p. 65).

"Le fait que la boucle conditionnelle TANT QUE suppose le plan "actions du corps de la boucle / test d'arrêt" constitue nous semble-t-il un obstacle non négligeable à une conceptualisation rapide de cette structure" (Rouchier, Samurçay, 84).

Rappelons que les élèves qui participent à cette expérience n'ont pas encore suivi d'enseignement d'informatique, du moins à l'école.

Démarches de solution nécessitées par le problème

La construction de l'itération nécessite de reconnaître un ensemble d'actions à répéter, c'est à dire le corps de l'itération sur lequel on fait agir un opérateur de répétition.

Les élèves ont à utiliser des expressions qui indiquent à l'opérateur la répétition de certaines actions. Rappelons que ces opérations de répétition sont exprimées par ce que nous appelons des avertisseurs de répétition. L'algorithme étant écrit en langue naturelle, l'éventail des choix possibles pour l'expression des avertisseurs est large selon sa relation avec la condition d'arrêt : Ils peuvent être de différents types.

- 1- Ils indiquent une répétition indéterminée tels que : "ainsi de suite", "continuer comme ça".
- 2- Ils indiquent une répétition déterminée; on peut distinguer deux sous-types :
 - 2-1 : les avertisseurs relatifs à un test d'arrêt différent de celui du dispositif.

Ex: refaire cette opération autant de fois qu'un certain nombre.

2-2 : les avertisseurs relatifs au test d'arrêt de la machine B

Ex: Recommencer (les actions) jusqu'à obtenir le 2ème nombre sur la machine B.

Les élèves doivent utiliser la machine B pour le test d'arrêt et doivent exécuter des actions sur cette machine pour vérifier si la condition d'arrêt est satisfaite.

L'algorithme nécessite l'utilisation de deux machines A et B en coordination. On peut penser que deux types de problèmes se poseront aux élèves :

- l'utilisation de la machine B comme compteur
- la coordination de deux machines.

Une situation de communication

La situation dans laquelle les élèves sont placés est une situation de communication homme-homme. Dans la partie émission qui correspond à la construction de l'algorithme, le message est écrit en langue naturelle et il est adressé à un interlocuteur qui est un autre élève, ce qui laisse une marge d'imprécision aux émetteurs. Ils peuvent se permettre certains implicites dans leur message. Néanmoins, les élèves savent qu'ils construisent un message destiné à des récepteurs qui devront pouvoir le faire fonctionner. On peut donc, s'attendre à ce que les émetteurs soient conscients de la nécessité de leur expliciter les actions à réaliser, les machines, les touches, les nombres à utiliser. Dans l'analyse de l'expression de l'algorithme, les aspects suivants seront donc pris en considération :

1 - Déclaration de nombres : il s'agit de savoir si les élèves spécifient à l'opérateur, les noms qui désigneront chaque nombre et comment il le font.

2 - Indication des machines à utiliser : nous cherchons à savoir si les élèves indiquent à l'opérateur sur quelles machines réaliser chaque opération.

3 - Sortie des données : nous nous intéressons à la signalisation que les élèves utilisent pour indiquer à l'opérateur où se trouve le résultat qu'il faut obtenir.

III.2 Phase de réception des messages

L'analyse de la partie réception porte sur l'interprétation et la validation que les élèves font du message reçu ; cette validation est de type pragmatique : les récepteurs font fonctionner le programme (Rouchier, Samurçay 1984).

Les récepteurs devraient faire fonctionner le message, en prenant toujours en compte les contraintes du dispositif :

1 - Nous parlerons d'interprétation large lorsque le groupe récepteur fait fonctionner le message en utilisant d'autres compétences que celle de l'opérateur.

2 - Nous parlerons d'interprétation stricte lorsque le groupe récepteur fait fonctionner le message en utilisant les compétences de l'opérateur.

3 - Il se peut que le groupe récepteur ne puisse pas faire fonctionner le message mais qu'il puisse donner une explication du pourquoi de son non-fonctionnement.

4 - On essaiera de dégager les relations entre la nature du message produit par les élèves en tant qu'émetteurs et l'interprétation par ces même élèves d'un autre message. Par exemple des élèves émetteurs qui ne respectent pas les compétences de l'opérateur vont-ils accepter la même erreur dans le message qu'ils auront à interpréter ?

IV- PARTIE EMISSION

IV.1- Résultats

Ci-dessous, nous présentons les résultats relatifs au contenu de l'algorithme.

1- Trois groupes n'ont pas construit d'itération. Les trois messages sont constitués chacun d'une liste d'instructions dont aucune n'évoque une action répétitive.

Ex : Groupe 1 (Problème P1)

- appuyer sur une des touches les chiffres de la machine A
- prendre un chiffre inférieur de 1 au chiffre pris au départ
- appuyer sur la touche +1 sur la machine B
- faire cette opération sur la machine A
- appuyer sur la touche =
- lire le résultat

.2 - Cinq groupes ont construit une itération mais de manière implicite.

Ex : Groupe 3 (Problème P1)

- Entier x
- Entier y
- Additionner à x sa valeur y fois comme suit
 $x+x+x+x+x+x+x+\dots+x$
 y fois
- Taper =
- Lire le résultat

Dans ce message l'opérateur doit savoir interpréter la représentation symbolique $x+x+x+x+\dots+x$
 y fois

L'algorithme n'est pas rédigé en actions à effectuer sur les machines telles que taper + ou taper =; c'est à l'opérateur d'induire de ces messages les actions à réaliser.

Ex : Groupe 7 (Problème P1)

Sur la machine A

Soit x et y deux nombres,

x représente les nombres de fois qu'il faudra taper sur y . Entre chaque y , il faudra taper sur la touche +

Le dernier y sera suivi de la touche =

Comme dans le précédent message l'itération n'est pas décrite en termes d'actions séparées les unes des autres, mais ici l'opérateur doit reconstituer la suite des actions à répéter en alternance.

3- Huit groupes ont indiqué de façon explicite une itération.

3.1- Sept groupes l'ont indiquée avec le type REPETER (n fois). Dans ce type de structure répétitive le comptage est laissé à la machine (l'opérateur dans ce cas). L'instruction qui permet d'effectuer ce comptage n'est pas

incluse dans les instructions du corps de l'itération. Ce type de structure répétitive correspond, par exemple, à l'instruction qui permet d'effectuer une boucle en langage LOGO.

Ex : Groupe 11 (Problème P.1)

Saisir la calculatrice, mettre en marche l'engin.

Introduire les données suivantes :

*On appelle x la première valeur, y la deuxième
taper x*

Opération

taper +

Faire cette opération y fois en remplaçant dans la dernière opération le signe + par =

Ex : Groupe 12 (Problème P.2)

Sur machine A,

1. choisir deux nombres entiers

2. introduire l'un des deux nombres

3. appuyer sur la touche +

4. réintroduire le même nombre

5. recommencer à l'étape 3 autant de fois que l'indique le deuxième nombre

6. appuyer sur la touche = : obtention résultat

7. appuyer sur +

8. introduire le résultat obtenu en 6

9. appuyer sur =

Ex : Groupe 10. (Problème P.2)

1 Tapez sur la machine A un nombre N donné

2 Tapez sur la machine B un nombre N_1

3 Tapez sur la touche + de la machine A

4 Tapez une seconde fois le nombre N

5 Refaire 3 et 4 autant de fois que le nombre inscrit sur B

6 L'opération terminée ; tapez la touche =

7 Tapez sur la machine B le nombre obtenu

8 Tapez + sur la machine A

9 Tapez sur la machine A le nombre inscrit sur la machine B

10 Tapez la touche =

Le type de structure répétitive utilisé dans ces trois exemples correspond au type de structure répétitive (REPETER n fois), dans le deuxième exemple, l'avertisseur étant en fin de liste, nous pouvons même assimiler le type d'itération à celle qu'on peut construire avec une instruction de branchement GOTO.

Le nombre de fois qu'il faut répéter les actions est pré-établi, c'est la machine ou dans ce cas l'opérateur qui réalise le comptage indépendamment des actions composant le corps de l'itération.

Ce type de structure répétitive trouvée dans ces exemples est dû au fait que les élèves n'ont pas utilisé la machine B comme compteur, mais qu'ils ont laissé à l'opérateur le soin de compter. Ils lui ont attribué une compétence qu'il n'avait pas.

3.2 - Un groupe a construit des itérations du type REPETER (actions) JUSQU'A CE QUE (condition)

- Ce groupe a construit une itération en utilisant la machine B comme compteur.

Ex : Groupe 9 (Problème P.1)

- *Introduire le 1er chiffre sur la machine*

- *Appuyer sur les touches +1 et = sur la machine B*

1 - *Appuyer sur + et introduire le 1er nombre sur la machine A*

a - *Appuyer sur la touche +1 et = sur la machine B.*

puis refaire l'opération 1 suivi de l'opération a

jusqu'à obtenir sur la machine B le 2ème nombre .

A ce stade appuyer sur l'égal de la machine A et lire le résultat.

Dans ce message les élèves ont construit leur itération du type REPETER (actions) JUSQU'A CE QUE (condition) . Le test d'arrêt est évalué à chaque passage de l'itération. Les actions "ajouter 1" et "taper +" modifient l'état de la machine B. L'évaluation de cet état à chaque pas de l'itération permet de comparer avec le "2ème nombre" pour pouvoir arrêter la boucle. Ces instructions appartiennent au corps de l'itération. (Résumé tableau 1)

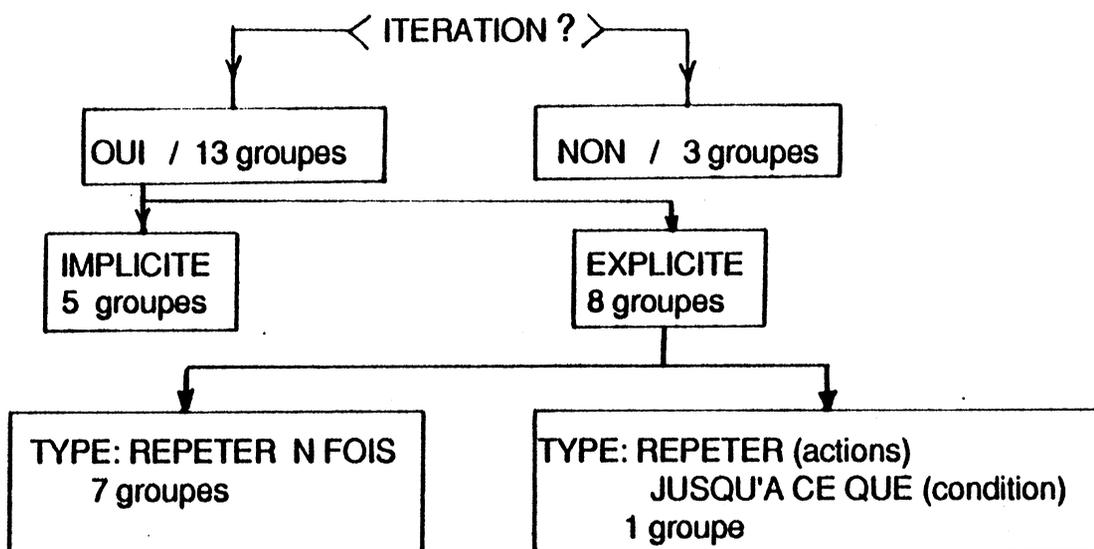


TABLEAU 1 : Répartition des messages selon l'existence d'une itération

Les aspects analysés dans la suite sont relatifs à des algorithmes avec répétitions explicites :

- Utilisation d'avertisseurs de répétition

Les élèves n'ont pas utilisé des avertisseurs pour indiquer une itération indéterminée, alors que pour exprimer une itération déterminée ils ont utilisé 2 types différents d'avertisseurs :

1. Les avertisseurs de répétition liés à un test d'arrêt différent de celui requis par le dispositif (le comptage est laissé à l'opérateur et n'est pas effectué à l'aide de la machine B).

Ex : Groupe 11 (Problème P1) page 53

Dans la phrase, "faire cette opération y fois", on peut remarquer l'avertisseur "faire-y fois" utilisé par les élèves pour exprimer l'itération. La condition d'arrêt "y fois" porte sur le comptage laissé implicitement à l'opérateur. Rappelons que l'enseignant avait indiqué à plusieurs reprises que l'opérateur ne savait pas compter.

2. Les avertisseurs de répétition liés au test d'arrêt réalisé avec la machine B comme compteur.

Ex : Groupe 9 (Problème P1) - page 54

Les élèves expriment leur itération à l'aide de l'avertisseur : "REFAIRE", en se repérant aux opérations 1 et a, en relation à la condition d'arrêt : JUSQU'A OBTENIR le 2ème nombre.

La condition d'arrêt porte sur les opérations à réaliser sur la machine B : ajouter 1, taper = , cette condition est évaluée à chaque passage de l'itération pour vérifier s'il faut arrêter ou non la répétition.

La machine B est le compteur qui contrôle l'arrêt de la boucle.

- Constitution du corps de la boucle.

Cinq groupes ont écrit un algorithme avec un corps d'itération bien délimité parfois même avec un renforcement du séquençement par une numérotation des instructions.

Ex : Groupe 13 (Problème P1)

*- Soient deux nombres entiers quelconques
le premier sera appelé "x", le second "y"*

** taper "x"*

** taper +*

*Ces deux opérations seront répétées un nombre de fois N
pour $N = y$*

** enfin, taper sur la touche =*

On aura, alors, le résultat.

Ces opérations seront effectuées avec la machine du type A.

Dans cet exemple: la phrase "ces deux opérations seront répétées...." délimite les opérations qui constitueront le corps de la boucle. Ces opérations distinguées par des étoiles, sont effectuées l'une après l'autre.

Ex : Groupe 10 (Problème P2) - page 53

Dans ce message les élèves ont numéroté les instructions qui composent le corps, la séquentialité est renforcée par la présence de l'ordre imposé par les numéros.

- Le test d'arrêt

Un seul groupe a utilisé la machine B comme compteur qui sert à arrêter la répétition des actions réalisées sur la machine A.

EX : Groupe 9 (Problème P1) - page 54

Les 12 autres groupes ont fait porter le test d'arrêt sur le comptage réalisé par l'opérateur.

- Gestion des machines

1. Dix groupes sur 16 ont écrit un algorithme en ne se servant que de la machine A.

Ex : Groupe 13 (Problème P1) - page 56

La dernière phrase de ce message mentionne que les opérations sont faites sur la machine A. En utilisant seulement la machine A les élèves changent les compétences de l'opérateur, en effet, l'arrêt de la répétition ou de l'itération est laissé au comptage de l'opérateur.

Ex : Groupe 7 (Problème P1) - page 53

Dans ce message, les élèves ont indiqué avec une phrase, au commencement de l'algorithme, que les opérations ne sont effectuées qu'avec la machine A. De même, que dans l'exemple précédent, les élèves ont changé les compétences de l'opérateur.

2. Quatre groupes ont réalisé leur algorithme en se servant des machines A et B

2.1. Un seul groupe a utilisé la machine B en coordination avec la machine A

Ex : Groupe 9 (Problème P1) - page 54

2.2. Trois groupes ont utilisé les machines A et B mais non de façon coordonnée comme il était requis par le dispositif.

Ex : Groupe 10 (Problème P2) - page 53

Dans ce message les élèves ont utilisé la machine B comme mémoire pour garder le deuxième nombre.

3. Les deux autres groupes n'ont pas indiqué les machines permettant de réaliser les opérations.

Ci-dessous nous présentons les résultats relatifs à la communication de l'algorithme :

- Déclaration des nombres

Sept groupes ont indiqué dans leur message les désignations. Ils les ont proposées par des phrases du type :

- *soient x et y deux nombres entiers*
- *appelons x le premier nombre*
- *appelons y le deuxième nombre.*

Les neuf autres groupes n'ont pas donné d'"étiquettes" aux nombres. Il les ont mentionnés au cours de l'algorithme en les repérant par des adjectifs ordinaux.

Ex : Groupe 14 (Problème P2)

1 - *Choisir un nombre entier quelconques*

2 - *Frapper ce nombre*

3 - *Appuyer sur la touche + puis refrapper le 1er nombre choisi.*

Faites cette action autant de fois que le 2ème nombre choisi.

-Indications des machines à utiliser

Nous avons déjà vu comment les élèves ont géré l'utilisation des machines. Nous nous intéressons ici à la façon dont ils ont indiqué les machines à utiliser. Cette indication n'est apparue que de deux manières différentes, soit au fur et à mesure qu'ils écrivent une instruction :

Ex : *taper sur la machine A un nombre N*
taper sur la machine B un nombre N.

soit d'une façon regroupée pour chaque machine

Ex : *Sur machine A*
1. *afficher x*
2. *taper +*
3. *afficher y*

2 groupes seulement n'ont pas indiqué à l'opérateur les machines à utiliser.

- Sortie de données

Sept groupes ont indiqué à l'opérateur où se trouve le résultat. Cette indication a été faite par des phrases du type :

Ex : *Lire le résultat sur la machine A*
Ex : *A ce stade appuyer sur l'égal de la machine A et lire le résultat.*

Les neuf autres groupes n'ont pas fourni de localisation des résultats.

IV. 2- Conclusion

Treize groupes ont utilisé l'itération pour exprimer la solution au problème mais la moitié l'a fait d'une manière explicite. La communication établie d'élève à élève ainsi que l'utilisation de la langue naturelle pour exprimer le message ont joué un rôle très important : de nombreux élèves ont exprimé la répétition d'une manière implicite.

Les types d'itérations utilisés par les élèves ont été :

1° REPETER (n fois) ; les élèves qui ont utilisé ce type d'itération ont supposé que l'opérateur savait compter, c'est pourquoi le comptage n'a pas été inclus dans les instructions du corps de la boucle.

2° REPETER ... JUSQU'A CE QUE; le test d'arrêt est réalisé à l'aide de la machine B en coordination avec la machine A (un seul groupe).

Les élèves n'ont jamais utilisé une itération du type "TANT QUE.....FAIRE". Il semble donc se dégager que ces élèves ont une difficulté à utiliser le compteur à l'intérieur de la boucle. Signalons de plus que cette observation ne semble pas en accord avec des hypothèses formulées par Janine Rogalski sur la relative facilité de l'utilisation d'un compteur pour arrêter la boucle (Rogalski, 1985). Cela nous amène à poser la question : s'agit-il d'une difficulté intrinsèque ou d'un autre fait dû à la situation présente de communication élève-élève ? En effet, nous l'avons déjà dit sans doute, le fait que l'algorithme soit à exécuter par un élève et non par une machine a

pu contribuer à l'attribution d'une compétence de comptage à l'exécutant de l'algorithme, et de ce fait rendre inutile l'utilisation d'un compteur.

Cette même expérience a été effectuée à Orléans par A. Rouchier au cours de la première séance d'une séquence d'enseignement.. Les productions des élèves ont été meilleures parce qu' un certain nombre d'élèves avaient déjà des notions d'informatiques (BASIC).

V- PHASE RECEPTION

V.1- Résultats

Les élèves qui reçoivent le message doivent le faire fonctionner en exécutant les instructions qu'il contient, et, en prenant le rôle de l'opérateur.

Treize groupes sur seize ont produit une réponse soit en exécutant le message, soit en expliquant la cause du non-fonctionnement.

- Interprétation large

Neuf groupes récepteurs ont effectué des interprétations larges :

Ex : Groupe 11 émetteur (Problème P1)

Saisir la calculatrice, mettre en marche l'engin.

Introduire les données suivantes :

On appelle x la première valeur, y la deuxième

Opération $\left\{ \begin{array}{l} \text{taper } x \\ \text{taper } + \end{array} \right.$

faire cette opération y fois en remplaçant dans la dernière opération le signe + par =.

Groupe 6 (récepteur)

$26 + 26 + 26 + 26 + 26 + 26 + 26 + 26$
 $=208$

On obtient le produit 26×8

Dans ce message avec une itération du type "REPETER (n fois)" le groupe récepteur a parfaitement interprété le comptage bien que la consigne donnée indiquait clairement qu'ils devaient se mettre à la place de l'opérateur, en respectant ses compétences. Le groupe récepteur a exécuté le message en comptant le nombre de fois qu'il fallait répéter l'opération.

Ex : Groupe 8 émetteur (Problème P2)

Soit deux nombres donnés

Opération 1 : taper le 1er nombre, autant de fois que le 2ème en appuyant sur la touche + entre chaque nombre

Opération 2 : appuyer sur la touche égal

Opération 3 : appuyer sur la touche + et réinscrire le nombre obtenu après la phase 2

Opération 4 : appuyer sur la touche = et lire le résultat.

Groupe 1 Récepteur

Avec les nombres 34 et 7

Opération 1 : 34 + 34 + 34 + 34 + 34 + 34 +

Opération 2 : = 238

Opération 3 : + 238

Opération 4 : = 476

Dans ce message, les élèves ne signalent pas sur quelle machine le récepteur doit exécuter les opérations. Néanmoins, le groupe récepteur fait fonctionner le message sans faire mention de l'oubli. Les émetteurs ont écrit pour l'opération 1 : "taper le nombre autant de fois que le 2ème, en appuyant sur la touche + entre chaque nombre". Cette opération est exécutée telle quelle par les récepteurs sur la machine A.

Ex : Groupe 7 (émetteur)

Sur la machine A

Soient x et y deux nombres

x représente le nombre de fois qu'il faudra taper sur la touche +.

Le dernier nombre y sera suivi de la touche =

Groupe 14 (récepteur)

(Les nombres donnés sont 26 et 8)

Les actions ne sont pas détaillées nous devons deviner ce qu'il se passe.

$$\begin{aligned} &8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 \\ &8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 + 8 \\ &= 208 \end{aligned}$$

Dans ce message, on voit que le groupe récepteur interprète le message d'une façon large, mais en spécifiant qu'ils le font consciemment.

Nous pouvons attribuer le fait que le comptage soit laissé à l'opérateur tant pour les émetteurs que pour les récepteurs à deux raisons :

- 1 -le message est écrit en langue naturelle
- 2-le message est adressé et est reçu par un autre groupe d'élèves.

Ces deux aspects permettent l'expression d'implicites entre les deux groupes, implicites qui ne pourraient pas être compris dans une communication élève-machine.

- Interprétation "stricte"

Nous parlons d'interprétation stricte lorsque le groupe récepteur fait fonctionner le message en se restreignant aux compétences de l'opérateur.

Aucun groupe récepteur n'a interprété de cette manière, bien qu'un groupe émetteur ait fourni un message bien-formé (utilisation de deux machines, restriction aux compétences de l'opérateur). Ce message n'a pas obtenu de réponse.

- Non-fonctionnement avec explication

Nous avons trouvé quatre messages pour lesquels les récepteurs ne font pas fonctionner l'algorithme mais donnent une explication de la cause du non-fonctionnement.

Ex : Groupe 14 (émetteur)

- 1- Choisir un nombre entier quelconque
- 2- Frapper ce nombre
- 3- appuyer sur la touche + puis refrapper le 1er nombre choisi.
Faites cette action autant de fois que le 2ème nombre indiqué (-1fois).
- 4- Appuyer sur la touche =
- 5- Noter le résultat
- 6- Appuyer sur la touche + puis retaper le résultat obtenu
- 7- Appuyer sur la touche =

Groupe 7 (récepteur)

- 1- sur quelle machine ?
- 2- a la 3ème partie il n'est pas possible de frapper le 2ème nombre (-1fois)
- 3- au 5ème, il n'est pas nécessaire de noter le résultat.

Ici, le groupe récepteur écrit des remarques aux émetteurs en expliquant la raison pour laquelle il ne peut pas faire fonctionner le message. Il est

possible de dire que ce groupe récepteur joue son rôle d'opérateur tel que la consigne l'indique. Cependant à la ligne 3 il ne fait pas de remarques à propos du comptage que doit effectuer l'opérateur, ni à propos de la non-utilisation de la machine B comme compteur. Malgré certaines remarques le groupe récepteur ne fait pas une interprétation stricte bien qu'il n'ait pas fait fonctionner le message.

Ex : Groupe 5 (émetteur)
Nombre n° 1 : x
Nombre n° 2 : y
On doit faire : x + x + x + x + x + + x
y fois puis appuyer sur la touche =
On ne doit utiliser que la machine A

Groupe 12 (récepteur)
Impossible à effectuer
n'existe pas de touche (.....)

La réponse que le groupe récepteur donne dans cet exemple ne nous permet pas de savoir si le groupe se considérait dans le rôle de l'opérateur avec ses compétences. Mais on peut supposer que l'interprétation est large parce que x + x + x + + x (avec ou sans points) n'indique pas à l'opérateur des actions de frappe sur les machines. Il semble que le seul motif de refus des récepteurs soit la présence des points de suspension.

- **Non réponse** : Nous avons trouvé 3 messages auxquels le groupe récepteur ne fournit pas des réponses.

V.2- Les échanges

Les échanges se sont effectués de la façon suivante:

Groupe 1 (émetteur)
Message sans itération

Groupe 8 (récepteur)
Non réponse

Groupe 8 (émetteur)
Message avec itération
du type REPETER (n fois)

Groupe 1 (récepteur)
Interprétation large

Groupe 2 (émetteur)
Message avec itération
implicite

Groupe 15 (récepteur)
Interprétation large

Groupe 15 (émetteur)
Message sans itération

Groupe 2 (récepteur)
Interprétation large

Groupe 3 (émetteur)
Message avec itération
implicite

Groupe 10 (récepteur)
Non fonctionnement
avec explication

Groupe 10 (émetteur)
Message avec itération
du type REPETER (nfois)

Groupe 3 (récepteurs)
Interprétation large

Groupe 4 (émetteur)
Message avec itération
du type REPETER (nfois)

Groupe 13 (récepteur)
Interprétation
large

Groupe 13 (émetteur)
Message avec itération
du type REPETER n (fois)

Groupe 4 (récepteur)
Interprétation large

Groupe 5 (émetteur)
Message avec itération
implicite

Groupe 12 (récepteur)
Non fonctionnement
avec explication

Groupe 12 (émetteur)
Message avec itération
du type REPETER (n fois)

Groupe 5 (récepteur)
Interprétation large

Groupe 6 (émetteur)
Message sans itération

Groupe 11 (récepteur)
Non réponse

Groupe 11 (émetteur)
Message avec itération
du type REPETER (n fois)

Groupe 6 (récepteur)
Interprétation large

Groupe 7 (émetteur)
Message avec itération
implicite

Groupe 14 (récepteur)
Interprétation large

Groupe 14 (émetteur)
Message avec itération
du type REPETER (n fois)

Groupe 7 (récepteur)
Non fonctionnement
avec explication

Groupe 9 (émetteur)
Message avec itération
du type REPETER.....
JUSQUE CE QUE.....
compteur machine B

Groupe 16 (récepteur)
Non réponse

Groupe 16 (émetteur)
Message avec itération
implicite

Groupe 9 (récepteur)
Non fonctionnement
avec explication

Le plus souvent les groupes ont été cohérents en passant de la phase émission à la phase réception. En effet, soit ils ont su faire fonctionner des messages présentant un moins grand degré d'implicite, soit ils ont refusé de faire fonctionner des messages présentant au contraire davantage d'implicite.

Dans la première catégorie, on peut ranger ainsi :

- les groupes 15, 1 et 6 qui n'avaient pas construit d'itération et qui ont su faire fonctionner des messages dans lesquels, soit l'itération était implicite, soit l'itération avait la forme REPETER n fois. Ils ont procédé à une interprétation large du message reçu.
- les groupes 3 et 5 qui en tant qu'émetteurs avaient construit de façon implicite une itération et ont su faire fonctionner des itérations de la forme REPETER n fois.
- les groupes 4 et 13 qui ont su faire fonctionner des messages ayant les mêmes caractéristiques que les leurs (REPETER n fois).
- les groupes 8, 11, 10 et 12, qui ayant émis un message avec la forme REPETER n fois, n'ont pas exécuté les messages sans itération (8, 11) ou avec itération implicite (10, 12).

En revanche les groupes 2 et 14 ont toléré des implicites plus grands dans les messages qu'ils avaient à exécuter que dans ceux qu'ils avaient émis. Le groupe 7 est encore plus surprenant par les exigences qu'il a manifesté

dans la phase réception (cf pp 61,62) alors qu'il avait rédigé un message avec itération implicite.

Le comportement du groupe 16 est très étonnant (mon interprétation du seul message respectant strictement les compétences de l'opérateur)mais ne peut malheureusement être expliqué faute d'explicitation de la part de ce groupe.

Illustrons ces commentaires par quelques exemples.

Groupe 4 (émetteur)

Appelons x le 1er nombre
Appelons y le 2ème nombre
1) prendre la machine A
2) afficher x
3) additionner ce nombre autant de fois que la valeur de y.
4) appuyer sur la touche =
5) soit y le résultat obtenu faire $y+y$
6) appuyer sur la touche =

Groupe 13 (récepteur)

Soit $x = 34$ et $y = 7$
a) 34
b) $34+34+34+34+34+34+34$
c) = 238
d) = $238 + 238$
e) = 476

Groupe 13 (émetteur)

soient deux nombres entiers quelconques
le 1er sera appelé "x", le 2ème "y"
* taper "x"
* taper +
Ces deux opérations seront répétées un nombre de fois "N" pour que : $N = y$
* enfin, taper sur la touche =
on aura alors le résultat
Ces opérations seront effectuées avec la machine du type A

Groupe 4 (récepteur)

$x=26$
 $y=8$
 $26+26+26+26+26+26+26+26 =208$

Le groupe 4 a écrit un algorithme avec une itération du type "REPETER (n fois)" en utilisant l'opérateur comme compteur. Les opérations ont été réalisées sur la machine A. Ce qui favorise son interprétation du message de même caractéristiques produit par le groupe 13.

Groupe émetteur 15

Machine A

- 1) afficher x
- 2) taper +
- 3) afficher y
- 4) taper +
- 5) afficher 10
- 6) taper =
- 7) passer sur la machine B
- 8) afficher le nombre obtenu sur la machine A
- 9) taper +1
- 10) taper =
- 11) lire le résultat

Groupe récepteur 2

x : 26 y : 8
 $26 + 8 + 10 = 44$ (A)
 $44 + 1 = 45$ (B)

Groupe émetteur 2

Si les deux nombres sont a et b
à l'aide de la machine A
- taper le nombre a
- ajouter à ce nombre, à l'aide du signe + b fois le nombre a
- lire et noter le résultat obtenu, appelons le c
- ajouter au résultat c, le même nombre c

Groupe récepteur 15

a = 34 b = 7
1) $34 + (34+34+34+34+34+34+34)$
2) c = 272
3) $272 + 272 = 544$

Le groupe émetteur 15 n'a pas produit un message avec itération et le groupe récepteur 2, bien qu'il ait construit un message avec itération, s'est contenté d'interpréter instruction par instruction sans faire de remarques à propos de la solution trouvée (celle-ci ne correspondant pas à la solution correcte).

V.3- Conclusion

1. Les messages en général ont été interprétés et exécutés. Sur 16 groupes récepteurs, 12 ont produit une réponse la plupart du temps dans des conditions différentes de celles requises par les contraintes du dispositif.

2. Le fait que le message soit écrit en langue naturelle a pu contribuer à des interprétations larges.

3. Les récepteurs ont été dans un premier temps émetteurs. Leur manière de formuler à ce moment-là a pu pour certains être un handicap pour interpréter des formulations différentes des leurs (groupes 7,9,10,12), mais d'un autre côté permettre à d'autres (groupes 4,13,14) de comprendre des formulations dont les caractéristiques sont identiques aux leurs, même si elles contiennent des implicites ou font appel à des compétences de l'opérateur non prévues par le dispositif.

VI- PHASE DE TRAVAIL INDIVIDUEL

La phase de production de messages individuels se produit après correction d'un exemple au tableau, où l'enseignant propose un modèle et l'efface immédiatement après l'avoir écrit

Dans cette phase, nous analyserons les aspects déjà cités dans la partie émission des échanges. Nous pouvons remarquer dans cette 5ème phase une nette amélioration de la construction des algorithmes par les élèves.

VI.1- Résultats

Les résultats de ces analyses seront exposés dans les tableaux suivant.

TABLEAUX : : Différences Phase 1 - Phase5
(échanges) (individuels)

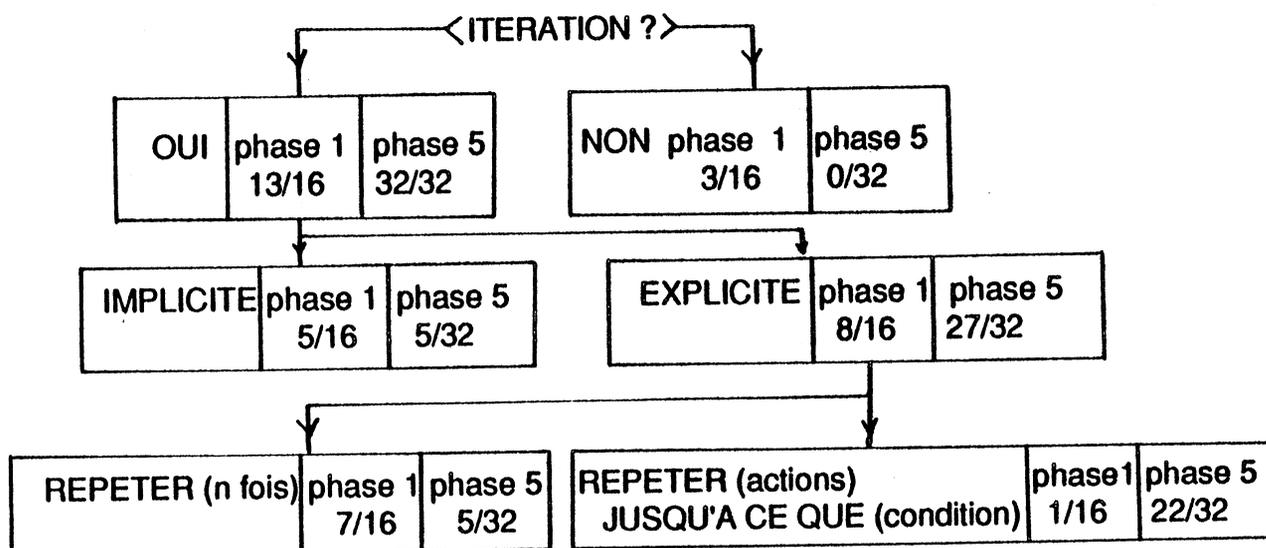


Tableau 1 : Classification des messages selon la construction de l'itération
(phase 1 - phase 5)

Nous avons pu observer une nette amélioration, dans la construction des algorithmes par les élèves, entre la phase 1 (échanges) et la phase 5 (travail individuel).

Dans la phase 1, trois groupes n'ont pas réussi à construire un algorithme itératif alors que, dans la phase 5, tous les élèves sont arrivés à le faire. Egalement dans la phase 5, l'écriture de messages avec itération implicite n'a été effectuée que par trois élèves et 22 sur 32 élèves ont construit des messages avec une itération explicite du type REPETER...JUSQU'A CE QUE

		PHASE 1	PHASE 5
AVERTISSEURS DE REPETITION	relatifs à un test d'arrêt différent de celui du dispositif compteur opérateur	6 / 16	2 / 32
	relatifs à un test d'arrêt du dispositif machine B compteur	1 / 16	22 / 32
CONSTITUTION D'UN CORPS		5 / 16	27 / 32
TEST D'ARRET sur la machine B		1 / 16	22 / 32

Tableau 2 : Différents aspects relatifs à la construction de l'algorithme

		PHASE 1	PHASE 5
DECLARATION DES NOMBRES		7 / 16	21 / 32
INDICATION DES MACHINES A UTILISER	machine A ou A' seule	10 / 16	6 / 32
	machines A ou A' et B coordonnées	1 / 16	22 / 32
SORTIE DES DONNEES EXPLICITEES		7 / 16	18 / 32

Tableau 3 : Différents aspects relatifs à la communication de l'algorithme.

VI.2 - Exemple d'évolution phase 1-phase 5

Le cas de Pierre et Laurent :

Les élèves Pierre et Laurent, en travaillant ensemble dans la première phase, ne sont pas arrivés à écrire un algorithme avec itération. Ils n'ont même pas évoqué une répétition. Leur algorithme ajoutait un premier nombre à lui-même, un deuxième nombre à lui-même puis additionnait les deux résultats. Ils n'ont fait ni déclaration de nombres ni sortie de résultats et ils ne se sont servi que de la machine A.

Par contre, dans la phase de travail individuel, après correction de l'enseignant, ces élèves ont écrit, séparément, un algorithme avec itération du type "REPETER (actions) JUSQU'A CE QUE (condition)". Ils ont utilisé de manière coordonnée les deux machines (A' et B), ont étiqueté les nombres à utiliser, ont délimité le corps de l'itération soit par accolades (Pierre) soit par numéros d'instructions (Laurent).

Pierre a initialisé des instructions et Laurent a indiqué où se trouvait le résultat.

PHASE 1 : groupe émetteur (Pierre - Laurent)

On se sert de la machine A

Taper un des deux nombres à l'aide des touches 0,1,2,3,4,5,6,7,8,9...appuyer sur la touche + puis retaper le nombre. Réappuyer sur la touche + et ensuite taper le deuxième nombre. Appuyer sur la touche + puis retaper le deuxième nombre et ensuite appuyer sur la touche =

PHASE 5 :

Pierre

X et N sont deux nombres entiers
mettre les machines A' et B en marche

sur la machine A' taper x

" " " " " B taper +1

sur machine A' taper *

sur machine A' taper x

sur machine B taper +1

sur machine A' taper =

puis recommencer 1 jusqu'à ce que B affiche N.

Laurent

On se sert des machines A' et B qui sont considérées en marche

X et N sont deux entiers quelconques

1 affichez sur la machine A' le nombre x

2 appuyez sur la touche *

3 tapez à nouveau le nombre x

4 sur la machine B appuyez sur la touche +1

Répéter les actions 2 3 4 dans l'ordre jusqu'à ce que le chiffre N soit affiché sur B

5 appuyer sur la touche = et lire le résultat

Avec l'exemple de ces deux élèves, nous avons voulu illustrer l'évolution qu'ils ont suivie d'une phase à l'autre. Dans la première phase nous voyons qu'ils ont une nette difficulté à écrire l'algorithme solution au problème donné, tandis que dans la phase de travail individuel, nous observons que grâce à l'aide apportée par l'enseignant qui a produit un algorithme solution modèle au tableau, ils ont pu écrire l'algorithme qui donne la solution correcte à un problème semblable à celui de la première phase.

VI.3 - Conclusion

Nous ne pouvons pas parler d'un réel apprentissage dans cette 5ème phase parce que plusieurs éléments y ont joué tels que : imitation du modèle de l'enseignant, réalisation immédiate d'un nouvel algorithme très peu de temps après les explications de l'enseignant ; ces éléments sont certainement les causes de l'amélioration de la réalisation de l'algorithme d'une phase à l'autre. Néanmoins, il faut noter que le problème P3 n'est pas le même que P1 et P2 malgré des similitudes.

Nous avons pu constater cette amélioration sur les aspects suivants:

1° Le nombre de messages avec itération implicite est réduit de moitié par rapport à la première phase (première phase 5/16 ; 2ème phase 5/32).

2° Les contraintes du dispositif ont été prises en compte dans la réalisation de l'algorithme; seulement 3 élèves sur 32 ont écrit un algorithme avec une itération du type "REPETER (n fois)" et 22 sur 32 ont fait une itération du type "REPETER.....JUSQU'A CE QUE....." en utilisant la machine B comme compteur. Ils ont utilisé donc la machine B en coordination avec la machine A. Les structures répétitives utilisées correspondent à la structure "REPETER (action) JUSQU'A CE QUE (condition)". La structure "TANT QUE

(condition) FAIRE (actions)" n'a jamais été utilisée dans cette phase.

Les messages sont écrits en termes d'opérations permises pour le dispositif (machine A', machine B, l'opérateur et ses compétences) et d'une façon structurée.

VII- CONCLUSION

Dans cette expérimentation les élèves sont placés dans une situation d'enseignement. Pendant la 1ère phase ils construisent un algorithme avec itération à l'aide de leurs connaissances initiales. Nous avons observé que:

-la construction de l'itération a été faite par un grand nombre d'élèves d'une manière implicite parce qu'ils n'ont pas utilisé correctement les contraintes du dispositif et parce que l'écriture de l'algorithme en langue naturelle le permettait.

-Les itérations utilisées par les élèves correspondent aux types: "REPETER (n fois)" et "REPETER.....JUSQU'A CE QUE". Le type "TANT QUE.....FAIRE" n'est pas apparu.

-La construction et l'utilisation d'un compteur pour arrêter l'itération n'ont été faites que par un seul binôme. Les autres ont laissé la fonction de compteur à l'opérateur, bien qu'elle ne soit pas dans ses compétences.

-Les élèves, en tant que récepteurs, ont réalisé, dans la plupart des cas, une interprétation large de l'algorithme. Ils n'ont pas utilisé d'une façon correcte les contraintes du dispositif, et l'écriture de l'algorithme en langue naturelle leur a permis d'interpréter les implicites du message.

Dans la deuxième phase les élèves reçoivent un enseignement lors de la correction des exercices et de l'élaboration d'un modèle correct de solution réalisée par l'enseignant. Dans cette phase la construction d'un nouvel algorithme, qui apporte une solution à un problème semblable au précédent, a présenté une nette amélioration du fait de la construction explicite de l'itération, de l'utilisation coordonnée de deux machines et de l'utilisation de la machine B comme compteur.

En conclusion, il semblerait que le choix des contraintes du dispositif

(langue naturelle, opérateur: élève) n'ait pas permis une construction achevée de l'itération et qu'il ait, au contraire, laissé la possibilité de nombreux implicites dans les algorithmes et dans leurs interprétations.



CHAPITRE III

CONSTRUCTION D'UNE ITERATION DANS UNE SITUATION DE COMMUNICATION ELEVE-MACHINE VIRTUELLE

I- Les choix de l'expérience

II- Description et déroulement de l'expérience .

III- Première séance

**III.1-Fonction des instructions du langage et analyse a priori de la
construction par les élèves d'un programme avec itération**

III.2- Analyse des observations

III.3- Exemples d'élaboration d'un programme

IV- Deuxième séance

IV.1-Fonction des instructions du langage

IV.2- Analyse des observations.

IV.3- Exemples d'élaboration d'un programme

V- Conclusion

Nous avons pu constater dans l'expérience précédente que les élèves, étant placés dans une situation de communication élève-élève et utilisant la langue naturelle comme langage de communication, ont construit leurs algorithmes en se permettant une certaine marge d'implicite, marges qui ne sont pas permises dans une communication élève-machine. Cela nous amène à consacrer ce chapitre à une expérience qui place les élèves dans un contexte de communication élève-machine virtuelle.

Les élèves doivent construire un programme destiné à un robot (invoqué); ce programme doit permettre le calcul de l'addition d'un certain nombre de nombres tracés sur des boules tirées aléatoirement d'une boîte. La communication s'établira grâce à un langage (créé par nous) que le robot est susceptible de comprendre.

I - LES CHOIX DE L'EXPERIENCE

Le problème

Le problème choisi consiste à trouver et à décrire un algorithme d'addition des nombres écrits sur des boules tirées aléatoirement une par une d'une boîte par un robot. Sa solution repose sur l'itération d'une opération familière aux élèves. Le caractère aléatoire des nombres à ajouter évite que les élèves effectuent l'addition à part de l'écriture de l'algorithme, ce dernier n'étant utilisé que pour faire écrire le résultat trouvé auparavant, et devenant complètement déchargé du processus répété d'addition. La solution du problème repose donc sur la construction d'une itération pour essentiellement deux raisons :

- 1° le nombre d'additions à réaliser est grand, donc le besoin d'un travail répétitif s'impose pour la rapidité du calcul,
- 2° la prise aléatoire des nombres à additionner ne permet pas l'utilisation d'une solution de type arithmétique.

Deux modalités ont été retenues qui correspondent à l'économie propre de la notion d'itération :

- 1) le nombre de boules est inconnu.
- 2) le nombre de boules tirées doit être exactement 150, la boîte en contenant un plus grand nombre.

Le dispositif

Le dispositif expérimental utilisé dans cette expérience a donc consisté à placer les élèves dans une situation problème pour laquelle un outil efficace de solution résidait dans l'écriture d'un programme comportant une boucle.

Le dispositif est composé d'un robot virtuel qui est capable de prendre les boules une par une d'une boîte, de lire le nombre inscrit sur chaque boule, de manipuler une calculatrice et d'écrire sur des ardoises magiques. Les élèves reçoivent un texte donnant l'énoncé du problème, la consigne de résolution ainsi que le langage pour communiquer avec le robot.

Le langage

Nous avons défini pour communiquer un programme au robot, un langage très rudimentaire et très étroitement adapté à la fois au problème et à l'objectif que nous avons d'étudier la construction et le fonctionnement d'une itération.

Nous avons retenu une itération du type ITERER dont on peut attendre qu'elle favorise la mise en évidence des opérations de pensées attachées à la construction du corps d'itération, à l'identification du caractère de nécessité de la condition et à sa gestion. De plus cette expression d'une boucle n'est contradictoire ni avec une stratégie donnant le primat à la condition d'arrêt (type TANT QUE), ni avec une stratégie donnant le primat au corps de l'itération (type REPETER).

Variations sur les compétences du robot

Une même modalité est donnée au même binôme (groupe de deux élèves) à deux séances, au cours desquelles les compétences du robot varient. Dans la première séance, le robot sait utiliser une calculatrice pour faire l'addition de deux nombres. Dans la deuxième séance, le robot a à sa disposition deux ardoises magiques, sur lesquelles il peut écrire avec la contrainte qu'un seul nombre à la fois peut être écrit sur une ardoise. Le robot sait ajouter dans ce cas deux nombres écrits chacun sur une ardoise différente.

Dans les deux séances, le robot sait prendre une boule dans la boîte, lire le nombre qu'elle porte et soit le taper sur la calculatrice (1^{ère} séance) soit l'écrire sur une ardoise (2^{ème} séance).

Dans la première séance la gestion de variables dans l'addition est masquée par la calculatrice; la deuxième séance introduit des problèmes supplémentaires:

- gérer trois nombres (les deux nombres à ajouter et leur somme) avec deux mémoires.

- initialiser l'itération d'une façon différente de son déroulement en régime de croisière.

Une situation d'interaction et de communication

Pour cette expérience nous avons préféré utiliser un contexte de communication avec un robot virtuel plutôt que placer les élèves face à une machine réelle. Les élèves face à une machine réelle auraient eu besoin d'un entraînement plus long pour réaliser cette expérience; avec le robot virtuel et le langage choisi pour communiquer avec lui les élèves sont capables d'apprendre le langage et aptes à réaliser l'expérience dans un temps suffisant court.

Deux types de simulation des programmes élaborés par les élèves sont possibles :

d'une part les élèves disposent de matériel (boîte, calculatrice quatre opérations et boules numérotées) permettant la simulation des actions du robot, d'autre part, l'exécution de leur programme est simulée par l'expérimentateur qui en prenant le rôle du robot devra exécuter le programme en interprétant textuellement les touches du langage.

La situation expérimentale ainsi conçue ouvre la possibilité à deux types d'interactions :

- les interactions entre élèves : les élèves devaient travailler à deux pour se mettre d'accord sur un **seul** programme. L'intérêt de ce type d'interaction a été présenté dans l'introduction de cette thèse.

- interaction avec l'expérimentateur : lorsque les élèves déclarent avoir établi un programme définitif, l'expérimentateur en simule l'exécution, comme s'il était le robot. La validation du programme assurée de cette façon est, pour les élèves, porteuse d'un nombre plus grand d'informations que celle que l'on peut attendre d'une machine réelle dont les messages d'erreurs sont en général très succincts et en petit nombre, ce qui concourt à leur polysémie. Par ailleurs la simulation met en évidence les relations entre actions et instructions habituellement occultées dans le fonctionnement d'un ordinateur. Enfin, l'expérimentateur peut commenter sa simulation ; par exemple si le corps d'une itération est vide, il déclare que le robot est contraint de répéter indéfiniment l'absence d'actions. Son rôle est donc important car, en mettant l'accent sur les éléments du programme à revoir, il engage les élèves à reprendre l'analyse du problème sur des bases plus précises que ne le ferait une machine réelle.

II- DESCRIPTION ET DEROULEMENT DE L'EXPERIENCE :

Lors de l'expérimentation les élèves travaillaient en binômes en dehors de la classe. Il s'agit d'élèves de collège (11 à 16 ans) n'ayant jamais reçu d'enseignement d'informatique.

L'expérience a été réalisée en deux séances, chacune d'environ une heure et espacées d'une semaine. Le matériel (boîte, 18 boules numérotées, calculatrice, ardoises magiques) est fourni aux élèves. 6 binômes ont réalisé la modalité 1 et 8 binômes la modalité 2.

1ère SEANCE :

L'expérimentateur distribue par écrit aux élèves le texte , la consigne et les significations des touches (cf. en annexe)

TEXTE FOURNI POUR LA MODALITE 1

On dispose d'une boîte avec des boules numérotées. On ne sait pas combien il y a de boules dans la boîte.

On désire connaître la somme de tous les nombres écrits sur les boules de la boîte.

On donne des ordres à un robot pour qu'il effectue cette somme.

Le robot est manipulé par un clavier à touches. On vous donne la liste des noms de touches et leur signification.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES NOMBRES SUR LES BOULES.

TEXTE FOURNI POUR LA MODALITE 2 :

On dispose d'une boîte avec 1000 boules numérotées de 1 à 1000.

On désire connaître la somme des nombres écrits sur 150 des boules de la boîte.

On demande à un robot d'effectuer la somme des nombres tracés sur 150 boules qu'il prend au hasard dans la boîte.

Le robot est manipulé par un clavier à touches.

On vous donne les noms des touches et leurs significations.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES 150 NOMBRES ECRITS SUR LES BOULES ;

SIGNIFICATIONS DES TOUCHES COMMUNES AUX 2 MODALITES

MARCHE : Si on appuie cette touche le robot est branché

PUB : Il prend une boule

LIS : Il lit le nombre qui est tracé sur la boule qu'il vient de prendre

TAPE : Il tape sur la calculette le nombre qu'il vient de lire

REP : Il répète indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REP

REC : Il recommence indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REC

CONT : Il continue indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche CONT

FINREP : Marque la fin de la liste des ordres à répéter

FINREC : Marque la fin de la liste des ordres à recommencer

FINCONT : Marque la fin de la liste des ordres à continuer

TAPE + : Il tape sur la touche + de la calculette

TAPE = : Il tape sur la touche = de la calculette pour obtenir un résultat

ARRET : En appuyant cette touche on débranche le robot

POUR LA MODALITE 1 :

TEST : En appuyant cette touche le robot teste si la boîte est vide ou non. Dans le cas où elle est vide, il interrompt son exécution mais il reste branché. Dans le cas où elle n'est pas vide, il continue.

POUR LA MODALITE 2 :

COMPT : Le robot a un compteur qui augmente de 1 chaque fois qu'on appuie sur la touche COMPT.

TEST N' : Si on appuie sur la touche TEST N' suivie d'un nombre, le robot regarde si la touche COMPT a le même nombre. Si la touche COMPT a le même nombre il arrête l'exécution, sinon il continue.

Les élèves disposent d'une calculatrice avec quatre opérations et d'une boîte contenant 18 boules numérotées pour les simulations. Le texte et la liste de significations des instructions sont lues à haute voix par les élèves.

L'expérimentateur s'assure que les élèves ont compris la consigne et répond à leurs questions éventuelles sur la signification des touches. Ensuite, les élèves travaillent seuls. Lorsqu'ils estiment avoir terminé leur programme, l'expérimentateur leur rappelle qu'ils peuvent le vérifier à l'aide du matériel fourni.

Si les élèves maintiennent leur programme, l'expérimentateur en simule l'exécution. S'il s'avère que ce programme ne donne pas le résultat attendu, les élèves sont invités à le remanier.

2ème SEANCE :

Cette deuxième séance a été réalisée en deux parties, une première partie consistant à mettre les élèves en contact avec les ardoises magiques pour apprendre comment le robot les manipule, et une deuxième partie consistant à résoudre le problème posé à la première séance, mais en utilisant cette fois-ci les ardoises magiques et non la calculatrice pour effectuer les additions.

1ère Partie : Manipulation des ardoises

Les élèves reçoivent le texte suivant :

On désire effectuer l'addition suivante :

$$24 + 147 + 33 + 49 + 110$$

On donne des ordres à un robot pour qu'il effectue cette somme.

Il va se servir de deux ardoises magiques pour se souvenir des nombres enregistrés dans sa mémoire.

Le robot est manipulé par un clavier à touches.

On vous donne la liste des noms des touches et leur signification.

Dans chaque ardoise on ne peut écrire qu'un seul nombre à la fois.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE L'ADDITION ECRITE AU-DESSUS.

SIGNIFICATIONS DES NOUVELLES TOUCHES INTRODUITES DANS CETTE SEANCE :

LIS : Le robot lit le nombre sur le papier.

EC 1 : Il écrit sur l'ardoise N° 1

EC 2 : Il écrit sur l'ardoise N° 2

ADD 12 : Il ajoute le nombre de l'ardoise N° 1 à celui de l'ardoise N° 2. (et il l'écrit sur une des ardoises)

EFF 1 : Il efface l'ardoise N° 1.

EFF 2 : Il efface l'ardoise N° 2.

L'objectif de la 1ère partie de cette deuxième séance consistait à initier au maniement des ardoises.

Nous pensions à priori que les élèves effectueraient l'addition en simulant le travail du robot. Ils écriraient le premier nombre de la suite sur une ardoise et les suivants sur l'autre. Après avoir ajouté les deux nombres écrits sur les ardoises, ils écriraient alors la somme de ces deux nombres sur une des deux ardoises, ce qui les conduirait à effacer l'une des ardoises. Le nombre suivant de la suite serait écrit sur l'autre ardoise laquelle aurait du être effacée auparavant. Nous pensions qu'en continuant cette démarche ils s'apercevraient qu'une ardoise sert à garder les sommes partielles et l'autre à enregistrer le nombre suivant dans la liste.

2ème partie :

Nous distribuons aux élèves un deuxième texte, donné ci dessous :

TEXTE FOURNI POUR LA MODALITE 1 :

On dispose d'une boîte avec des boules numérotées. On ne sait pas combien il y a de boules dans la boîte.

On désire connaître la somme de tous les nombres écrits sur les boules de la boîte.

On donne des ordres à un robot pour qu'il effectue cette somme.

Le robot est manipulé par un clavier à touches.

On vous donne les noms des touches et leur signification.

Le robot a devant lui la boîte avec des boules et deux ardoises magiques dans lesquelles il va écrire des nombres. Il ne peut écrire qu'un seul nombre à la fois dans chaque ardoise.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES NOMBRES SUR LES BOULES.

TEXTE FOURNI POUR LA MODALITE 2 :

On dispose d'une boîte avec 1000 boules numérotées de 1 à 1000.

On désire connaître la somme des nombres écrits sur 150 boules de la boîte.

On demande à un robot d'effectuer la somme des nombres tracés sur 150 boules qu'il prend au hasard dans la boîte.

Le robot est manipulé par un clavier à touches. On vous donne les noms des touches et leur signification.

Le robot a devant lui, la boîte avec des boules et deux ardoises magiques dans lesquelles il va écrire des nombres. Il ne peut écrire qu'un seul nombre à la fois dans chaque ardoise.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES NOMBRES ECRITS SUR LES BOULES.

SIGNIFICATIONS DES TOUCHES INTRODUITES DANS CETTE DEUXIEME PARTIE

EC 1 : Il écrit sur l'ardoise N ° 1.

EC 2 : Il écrit sur l'ardoise N ° 2.

EFF 1 : Il efface l'ardoise N ° 1.

EFF 2 : Il efface l'ardoise N ° 2.

ADD 12 : Il ajoute le nombre de l'ardoise N ° 1 à celui de l'ardoise N ° 2.

III- PREMIERE SEANCE

III.1- Fonction des instructions du langage et analyse a priori de la construction par les élèves d'un programme avec itération :

Tout programme doit commencer par la commande MARCHE qui met le robot sous tension et finit par la commande ARRET qui met le robot hors tension.

Une itération de type ITERER peut être traduite à l'aide de l'un quelconque des trois couples (REC,FINREC), (REP,FINREP), (CONT,FONCONT) et des instructions TEST (première modalité) ou TEST N° (seconde modalité). La possibilité, de laisser exprimer l'itération sous l'une ou l'autre de ces formes correspond à des expressions utilisées par les élèves lors d'une pré-expérimentation dans laquelle les élèves avaient à résoudre le même problème mais en s'adressant au robot en langue naturelle ; elle respecte des nuances de significations relevées lors de ces premières analyses ; la restriction à un vocabulaire plus étroit aurait pu gêner de façon non pertinente notre étude.

Dans le couple (REC,FINREC) le premier terme est un avertisseur d'itération, la rencontre du second terme dans le programme provoque un branchement inconditionnel vers l'instruction qui suit REC. En fait cette signification de FINREC lui est conférée par l'usage dans la définition de REC de l'adverbe "indéfiniment". On peut alors considérer que la signification de l'itération est essentiellement portée par REC, le code FINREC joue surtout le rôle de marqueur de fin de la séquence d'instructions à répéter. Ces commentaires sont valables également pour les couples (REP,FINREP) et (CONT,FINCONT).

L'instruction TEST recouvre une action complexe, elle comprend à la fois l'évaluation d'une condition et une décision de branchement dont l'effet est la sortie de la boucle. Le caractère *ad hoc* du robot apparaît dans la définition de cette instruction qui dispense de la construction de la formulation logique de la condition.

Pour l'instruction TEST N° l'élève a de plus à la mettre en relation avec l'instruction COMPT qui assure la mise à jour d'un compteur, et à indiquer l'information sur le nombre de boules à ne pas dépasser.

Les instructions TEST et TEST N° présentent une relative complexité, le choix de les exprimer comme des entités non décomposées du langage a été fait pour libérer autant que possible l'expression et la gestion de la condition des problèmes de formulation souvent difficiles qu'elles soulèvent.

Nous pensions a priori que premièrement pour construire l'itération, les élèves devraient reconnaître les instructions qui formeront plus tard le corps de l'itération, deuxièmement, que lorsque cette reconnaissance serait effectuée la nécessité d'indiquer une répétition s'imposera. Pour indiquer cette répétition les élèves devraient utiliser une de trois instructions REP-REC-CONT et délimiter les instructions à répéter (corps de l'itération) par les commandes FINREP-FINREC-FINCONT, ensuite, que le besoin d'arrêter la boucle les entraînerait à l'utiliser des instructions TEST ou TEST N° (selon la modalité)

La modalité 2 est la plus contraignante dans la mesure où elle exige de calculer la somme relative à un nombre donné (150) de boules inférieur au nombre total de boules dans la boîte (1000). Il est plus important dans cette modalité de savoir arrêter la répétition que dans la modalité 1 où elle pourrait s'arrêter d'elle-même faute de boules. Nous pensions donc a priori que l'utilisation de l'instruction TEST s'imposerait davantage dans la modalité 2 que dans la modalité 1.

III.2- Analyse des observations

Dans la plupart des cas les élèves ont pu travailler aussi longtemps qu'ils l'ont souhaité (environ 1 heure) et ont pu élaborer un programme final dont l'exécution donnait le résultat désiré. De façon précise, 14 binômes ont été observés dont quatre seulement ont été interrompus dans leur travail par des contraintes horaires; 6 binômes ont été soumis à la modalité 1 et 8 binômes à la modalité 2.

L'observation du travail des binômes a confirmé certaines de nos prévisions mais mis en évidence d'autres difficultés non prévues.

Comme nous l'attendions, l'itération a été construite comme une réponse au problème, et l'anticipation de l'arrêt dans la modalité 1 n'a guère été spontanée. La seconde modalité a, en général, nécessité un temps plus long que la première (4 binômes n'ont pas terminé leur travail) : l'usage conjoint des instructions COMPT et TEST N° a en effet soulevé des difficultés spécifiques.

En revanche les simulations des élèves n'ont pas joué à propos de la condition d'arrêt le rôle que nous leur avons attribué a priori. De plus une interprétation non prévue des instructions FINREP (resp. FINREC, FINCONT), consistant à leur accorder une fonction d'arrêt de l'itération, a gêné la mise en place de cette condition.

Le premier résultat marquant concerne le travail de construction du programme destiné au robot. La liste définitive des instructions à donner n'est élaborée qu'après maintes modifications, maintes refontes dans l'écriture du programme dues à des retours arrière dans l'analyse du problème. Le nombre des programmes intermédiaires écrits au cours de la séance est un indicateur de l'importance de ce travail de construction.

Binômes	Nombre de programmes intermédiaires	Nombre de simulations intermédiaire de l'expérimentateur
Olivier et Enrique Esthèle et Nathalie	1	0
Karine et Delphine	3	2
Isabelle et Gabrielle	4	1
Patrick et Morad	5	5
Massimo et David	5	4
Lilian et Christophe David et Hervé	1	0
Ariel et Pierre-Yves Christèle et Valérie	2	0
Catherine et Pascale	4	2
Michel et Frédéric	4	1
J. François et A. Marie	6	1
Stéphane et Nicolas	6	5

TABLEAU 1 : Nombre de programmes et de simulations intermédiaires de l'expérimentateur
(ne sont décomptés ni le programme final, ni la simulation de ce programme)

Construction et formulation de la boucle

Utilisation des avertisseurs d'itération

Plusieurs étapes ont pu être dégagées.

- *absence d'avertisseurs d'itération : la juxtaposition d'instructions simples.*

De façon générale, dans les deux modalités, la première phase de travail a consisté pour les élèves à traduire en termes d'instructions la liste des actions qu'ils effectueraient l'une après l'autre s'ils exécutaient eux-mêmes l'addition. Cette liste est plus ou moins longue :

- elle peut comporter la juxtaposition de deux ou plusieurs séquences d'instructions directes, par exemple :

MARCHE TEST PUB LIS TAPE TAPE+ PUB LIS TAPE TAPE +
PUB LIS TAPE TAPE + TEST TAPE = ARRET
Massimo, David (Modalité 1)

MARCHE PUB LIS TAPE TAPE + COMPT PUB TAPE TAPE+ COMPT
Christelle, Valérie (Modalité 2)

- elle peut ne comporter qu'une séquence

MARCHE PUB LIS TAPE COMPT
David et Hervé (Modalité 2)

Cette phase, qui est en quelque sorte une prise de contact avec le problème, a concerné 4 binômes dans la modalité 1, 5 binômes dans la modalité 2.

-mention d'un avertisseur mais absence du corps de l'itération.

L'idée d'utiliser un avertisseur d'itération apparaît après la phase précédente, lorsqu'il y a prise de conscience de l'impossibilité d'écrire la suite des séquences d'instructions à répéter, ou du coût que cela représenterait.

Cette prise de conscience a été dans certains cas la conséquence d'une simulation de l'expérimentateur. L'idée d'économie est parfois explicitée :

" REP ça ira plus vite" (Hervé)

"il faut mettre quelque chose qui fasse plus vite " (Massimo)

Christelle explique à Valérie qui ne voyait pas d'autre moyen que d'écrire 150 fois sur la feuille la même séquence : "mais ça va te prendre toute la page, c'est long, c'est pour toute la journée"

Souvent les élèves se sont contentés d'exprimer l'idée d'itération et de son arrêt en ajoutant à la suite de la (ou des) séquences(s) écrite(s) le couple REP FINREP (resp. FINREC, CONT FIN CONT), par exemple :

**MARCHE PUB LIS TAPE TAPE+ REP FINREP TAPE= ARRET
(Stéphane, Nicolas) Modalité 2**

**MARCHE TEST PUB TAPE TAPE+ PUB LIS TAPE TAPE + PUB LIS TAPE TAPE +
TEST REC FINREC ARRET
(Massimo-David) Modalité 1**

Les instructions ne sont pas utilisées ici conformément aux règles du langage de communication avec le robot, mais de façon très proche de la langue naturelle, dans laquelle la mention de la répétition peut se faire après coup par une expression telle que "continuer ainsi de suite jusqu'à épuisement des boules" ou "répéter cela 150 fois". Cette forme d'itération a été moins fréquente dans la modalité 2 (3 sur 8) dans la modalité 1 (4 sur 6), parce que plus difficilement compatible avec la contrainte de répéter exactement 150 fois une addition.

En effet, dans le cas de la modalité 1, la touche FINREP (resp. FINREC FINCONT) a été placée par les élèves après REP(resp. REC,CONT) pour arrêter l'acte de répéter sans préciser la condition d'arrêt. Bien que cette instruction possède le caractère de délimiteur du corps de l'itération, ce fait a été complètement ignoré, ce qui n'est pas surprenant du fait de l'absence du corps de l'itération. Le nom de l'instruction a pu conforter les élèves dans leur interprétation ou même y contribuer.

A cette étape, les simulations de l'expérimentateur ont joué un rôle crucial, en montrant que le robot ne répétait rien, puisque le corps de l'itération était vide. Ces simulations ont permis le passage à une phase suivante, dans laquelle l'avertisseur opère sur une séquence d'instructions qui forme le corps de l'itération.

- vers une forme achevée de l'itération

Une fois la nécessité du corps de l'itération ressentie, les élèves ont cherché à le construire. Deux grands types de construction ont pu être observés :

- construction d'un corps de boucle complet
- construction d'un corps sans condition d'arrêt, celle-ci soit n'étant pas du tout indiquée dans le programme, soit figurant en dehors de la boucle.

L'achèvement de la construction du corps de la boucle a en général été mené à bien encore grâce aux effets des simulations de l'expérimentateur.

Les trois phases qui viennent d'être dégagées n'ont pas concerné tous les binômes. Certains ont commencé le travail à la deuxième phase ou directement à la troisième. Des retours à des phases antérieures ont même eu lieu pour deux binômes : Stéphane et Nicolas, Massimo et David.

Mise en place de la condition d'arrêt

L'arrêt de la répétition a été évoqué par tous les binômes mais le problème n'a que rarement été résolu par une utilisation immédiatement correcte des instructions TEST et TEST N°. Leur emploi et leur insertion, à la place adéquate, a même en général été un des derniers problèmes à être résolu au cours de l'élaboration du programme.

Ces instructions ont souvent été placées à la suite du corps de l'itération parce que le besoin de vérifier la condition d'arrêt à chaque pas de l'itération ne s'est fait sentir qu'après coup. Cela se traduit dans la modalité 1.

- soit par l'utilisation de FINREP (FINREC ou FINCONT) comme arrêts d'itération (3 binômes). Gabrielle dit par exemple en écrivant FINCONT :

"on l'arrête de continuer et après on lui demande le résultat"

- soit par l'utilisation de TEST en tant qu'instruction permettant l'arrêt en dehors du corps d'itération. (utilisation due à une omission complète de la partie conditionnelle de cette instruction) (2 binômes).

Exemple :

MARCHE REP PUB LIS TAPE TAPE + FINREP TEST TAPE = ARRET
(3ème programme intermédiaire de Karine et Delphine)

On retrouve là le comportement spontané du locuteur dans toute forme de communication qui consiste à s'adresser à l'interlocuteur comme s'il partageait les mêmes compétences et le même savoir.

L'élève exécutant lui-même les actions ne ressent pas le besoin de vérifier à chaque prise l'état de la boîte ; l'arrêt interviendra naturellement lorsqu'il n'y aura plus de boules. Il suffit donc de demander au robot d'arrêter l'itération à la fin du programme, au moment où la boîte est vide.

La compréhension complète des instructions TEST et TEST N° a probablement été difficile, car elle exigeait non seulement la reconnaissance de la partie conditionnelle de ces instructions mais aussi la coordination entre la condition et son double effet (poursuite de l'exécution ou arrêt).

L'exemple fourni par un programme intermédiaire de Massimo et David illustre cette difficulté de coordination. Ces deux élèves ont en effet écrit l'instruction TEST à deux endroits du programme avec deux significations différentes, la première instruction TEST ayant pour but de faire continuer l'exécution, la deuxième de l'arrêter.

**MARCHE TEST PUB LIS TAPE TAPE + PUB LIS TAPE
TAPE + PUB LIS TAPE TAPE + TEST TAPE = ARRET**

Dans la modalité 2, la complexité est encore plus grande parce que la coordination interne de l'instruction TEST N° dépend elle-même de la coordination entre TEST N° et COMPT. Cette double coordination n'a été comprise rapidement que par deux binômes.

Les autres ont soit négligé l'utilisation d'une des deux instructions, soit utilisé TEST N° 150 de façon indépendante de COMPT en dehors du corps de l'itération avec une signification d'arrêt (de la même façon que TEST a été utilisée dans la modalité 1).

Trois binômes ont réussi à coordonner à la fin de leur travail les instructions COMPT et TEST N° 150, après des simulations de l'expérimentateur et des oppositions de point de vue (Christelle et Valerie, Michel et Frédérick, ce dernier cas sera présenté ensuite). Les trois autres ne l'ont pas fait.

	Juxtaposition d'instructions simples	Mention de l'avertisseur, absence du corps	corps		
			sans test	non coordination du corps et du test	coordination du corps et du test
MODALITE 1					
Patrick et Morad Massimo et David	1 → 1,2,3 → 5 →	2 → 4 →	3,4,5 →		6 →
Olivier et Enriqne Karine et Delphine Esthèle et Nathalie Isabelle et Gabrielle	1 → 1,2 →	2 → 1,2 → 3,4 →	3 →		6 → 2 → 4 → 1,2 → 5 →
MODALITE 2					
Catherine et Pascale	1 →	2 →		3,4 (inachevé)	
Stéphane et Nicolas	1 →	2 →	3,4,5 →		6 (inachevé)
J. François et A. Marie	1 → 5 →	6 (inachevé)		2,3,4	
David et Hervé Christelle et Valérie	1 → 1 →	2 (inachevé)			2 →
Michel et Frédéric Ariel et Pierre-Yves Lilian et Christophe	1 →		1 → 1 →	2,3,4 →	5 → 2 → 2 →

TABLEAU 2 : Succession des types de programmes écrits par les élèves au cours de leur travail dans la 1ère séance.

(Les programmes sont désignés dans le tableau par leur rang dans l'ordre dans lequel ils ont été écrits).

III.3- Exemples d'élaboration d'un programme

Le cas de Patrick et Morad (Modalité 1)

Il illustre le travail de construction qui a eu lieu dans la plupart des binômes, et qui a débouché après l'écriture de nombreux programmes intermédiaires sur un programme final correct. Il atteste aussi du rôle important de l'expérimentateur dans l'évolution du travail.

Après une courte phase de juxtaposition d'instructions simples (PUB LIS TAPE TAPE +), Patrick et Morad prennent conscience de l'aspect répétitif des actions du robot.

- P : "cest ça PUB LIS TAPE TAPE+ et après il recommence .
PUB LIS TAPE TAPE+.... il recommence tout ça"
M : il recommence les ordres qui sont marqués
il répète c'est ça c'est REP....il répète indéfiniment"
P : "oui, et après FINREP marque la fin et après TAPE= et
après tape ARRET".

Le programme écrit est alors :

(P1) MARCHE PUB LIS TAPE TAPE+ REP FINREP TAPE= ARRET

Ils ont mentionné l'avertisseur d'itération REP mais n'ont pas construit le corps. L'expérimentateur leur propose de vérifier leur procédure en la simulant avec 10 boules dans la boîte. Ils effectuent alors une simulation qui consiste à exécuter l'algorithme selon le fonctionnement qu'ils imaginent mais qu'ils n'ont pas su exprimer dans le langage de communication avec le robot. C'est là encore une bonne illustration de la faiblesse des feed-back apportés par les propres simulations des élèves.

P : "après REP.... REP on a mis. Il répète indéfiniment,
ça veut dire qu'il fait toujours ça qu'il prend le
nombre qu'il marque toujours alors TAPE+....
alors TAPE+ après après FINREP après TAPE =
et après ARRET".

Le programme P1 jugé correct est proposé à l'expérimentateur qui simule une exécution en indiquant que le robot ne fait rien entre REP et FINREP puisqu'il n'y a aucune instruction entre ces deux instructions. Cette simulation provoque la construction du corps de la boucle. La condition d'arrêt n'est toujours pas évoquée.

Le programme écrit est le suivant.

(P2) MARCHE PUB LIS TAPE TAPE+ REP PUB LIS TAPE TAPE+
FINREP TAPE= ARRET

Il est à nouveau proposé à l'expérimentateur. Celui-ci montre que le programme "boucle", en faisant la même séquence d'actions de plus en plus vite. Patrick exprime son étonnement : "oui, mais on a marqué FINREP".

L'expérimentateur intervient pour rappeler le rôle de marqueur de fin de boucle que possède FINREP. Patrick et Morad écrivent ensuite deux programmes intermédiaires, soumis à chaque fois à la simulation de l'expérimentateur, en ayant comme stratégie d'enlever les instructions qui semblent procurer des ennuis, c'est à dire tantôt REP, tantôt FINREP. L'idée d'écrire 10 fois la même séquence pour calculer la somme des nombres portés par 10 boules est même évoquée par Morad. Mais ils en arrivent finalement à réécrire le programme P2 avec un corps de boucle sans condition d'arrêt.

L'expérimentateur simule le programme en continuant d'essayer de prendre des boules dans la boîte après qu'elle ait été vidée. Cela permet la prise de conscience par Patrick de la nécessité de la touche TEST.

P : "il faut mettre TEST
en mettant TEST il dit si elle est finie
elle est finie et il s'arrête on met TEST 1 à après TAPE"

Le programme final est alors écrit :

MARCHE PUB LIS TAPE REP TAPE+ PUB LIS TAPE TEST FINREP TAPE= ARRET
(la structure est ici celle de REPETER... JUSQU'A CE QUE).

Le cas de Michel et Frédéric (modalité 2)

Ce cas est un exemple de la difficulté de coordination des instructions COMPT et TEST 150. En effet, très rapidement, Michel et Frédéric décident de mettre en place une itération avec construction d'un corps. Mais le problème "Comment arrêter l'itération au bout de 150 répétitions ?" n'est résolu qu'à la suite d'une intervention de l'expérimentateur.

Frédéric pense d'abord à TEST 150 en tant qu'instruction d'arrêt, et Michel à FINREP. Ce désaccord aboutit à un programme de compromis dans lequel l'arrêt est demandé à la fois par FINREP et TEST N°149. La place de TEST N°149 dans le corps de la boucle n'est pas motivée. Cette instruction avec la signification qu'elle a alors pour les élèves aurait pu se trouver après FINREP. La boucle ne porte que sur COMPT qui est interprétée par les élèves comme la demande d'addition de deux nombres.

MARCHE PUBLIS TAPE REP COMPT TEST N°149 FINREP TAPE=ARRET

Ce programme est d'ailleurs rapidement modifié en plaçant TEST N°149 en dehors du corps, parce que "TEST ne doit pas être répété" (Michel), puisqu'il ne possède qu'une valeur d'arrêt.

A l'étape suivante, Michel et Frédéric parviennent à la compréhension de COMPT comme compteur, mais un désaccord s'établit de nouveau entre eux. Frédéric reste attaché à son interprétation de TEST N° comme instruction d'arrêt qui empêche d'appréhender la relation entre TEST N° et COMPT, alors que Michel commence à la percevoir.

M : "et le moment où il doit s'arrêter, on le sait qu'avec le TEST, hein"

F : "attends on est où là"

M : "si il a le même nombre que COMPT il s'arrête"

F : "non je crois que c'est là FINREP. Attends hein tu vas voir TEST N°"

M : "tu le mets après ?"

F : "ouais parce que pour pas que ça répète ça"

M : "ouais mais là"

F : "ça c'est la fin de la liste"

M : "ça va répéter combien de fois ? Comment on peut savoir comme ça va répéter ? Ca répète, ça répète, ça répète"

F : "et ben ça c'est la fin de la liste à répéter" il s'agit de FINREP),
"mais ça c'est ce qui va arrêter de répéter" (il s'agit de TEST N°)

Pour l'instant Frédéric ne tient pas compte de l'argument décisif de Michel, sur la nécessité de savoir à quel moment le nombre de répétitions est atteint, car la question ne se pose pas encore pour lui.

Un nouveau programme, avec un corps d'itération encore incomplet malgré des ajouts et avec TEST N°, en dehors du corps est proposé à la simulation de l'expérimentateur. Celui-ci montrant que le programme ne fonctionne pas, à cause du caractère incomplet du corps, Michel et Frédéric décident de recommencer.

Le même débat s'instaure, Michel explicitant le lien entre COMPT et TEST N°, Frédéric voulant mettre TEST après "la fin des choses à répéter". C'est alors que l'expérimentateur rappelle la signification de l'instruction TEST. Cette intervention apporte un soutien institutionnel fort au point de vue de Michel. Frédéric reconnaît le caractère conditionnel de TEST.

F : "et ouais ça compare[...] Alors il faut comparer à chaque fois.
C'est bon ce que tu disais."

Frédéric et Michel terminent ensuite sans problème l'écriture du programme.

le cas de Catherine et Pascale (Modalité 2)

Ce binôme n'a pas pu arriver au bout de son travail en raison du grand nombre de difficultés auxquelles il s'est heurté dans la construction du programme. Le rôle de l'expérimentateur a été très important pour permettre des évolutions, mais il aurait fallu davantage de temps pour que le travail du binôme débouche sur un programme correct.

Catherine et Pascale commencent par écrire une liste d'instructions (juxtaposition d'instructions simples). La touche COMPT signifie pour elles "ajouter les nombres". Pascale a l'idée de la répétition mais Catherine n'en voit pas le besoin.

P : "regarde il y a REP"
C : " on a pas besoin de répéter"
P : "regarde il y a REC aussi"
C : "on est pas obligé de le mettre"
P : "regarde on en a besoin"

Pascale lit à nouveau les significations des touches, et dit à Catherine qu'il faut utiliser les trois (REP, REC, CONT), parce que le robot doit "continuer à recommencer".

Catherine prolonge la liste des instructions en s'aidant du schéma du clavier. Elle a tendance à inclure dans le programme les nouvelles touches qu'elle lit au fur et à mesure sur le clavier.

Le programme suivant est écrit :

(P1) : MARCHE PUB LIS TAPE COMPT PUB LIS TAPE COMPT REC COMPT FINREC
FINREP TAPE + TAPE= TEST N' ARRET

Lorsqu'elles ont terminé, l'expérimentateur leur propose de vérifier par elles-mêmes leur procédure, en se servant des dix boules de la boîte. Elles effectuent une simulation sans se servir des boules malgré les incitations de l'expérimentateur à faire la simulation avec les boules. Elles considèrent que leur programme est correct et le donnent à l'expérimentateur. Il simule à son tour avec 10 boules et en indiquant la signification de l'instruction COMPT et celle de l'instruction REC suivie de CONT.

Les élèves essaient un nouveau programme et décident qu'il faut utiliser CONT pour que le robot continue à réaliser la liste d'instructions précédemment écrite.

P : " tu mets CONT (elle lit la signification)....
ah oui comme ça, il nous écoute encore"

La touche COMPT est toujours utilisée pour ajouter le nombre de boules.

Le programme est alors écrit :

(P2) : MARCHE PUE LIS TAPE COMPT CONT FINREP FINREC FINCONT TAPE+
TAPE= TEST N' ARRET

Du fait que l'expérimentateur a simulé leur programme jusqu'à l'instruction CONT elles recopient le reste du programme sans le modifier. Elles font une vérification orale de leur propre programme en le considérant correct (on remarque le faible rôle de leur propre vérification sur l'évolution de leur programme ainsi que la réticence à utiliser le matériel). L'expérimentateur simule à nouveau et explique que les instructions FINREP-FINREC-FINCONT ne peuvent pas être utilisées toutes les trois et explique leurs relations avec REP-REC et CONT. Il continue à simuler le reste du programme en faisant la remarque que l'instruction TEST N' doit être accompagnée d'un nombre, sinon le robot ne comprendra pas la signification de l'instruction. Elles écrivent un nouveau programme en éliminant les instructions FINREP et FINREC et en ajoutant 5 après TEST N' tout à fait au hasard.

Le programme suivant est alors écrit:

(P3): MARCHE PUB LIS TAPE COMPT CONT FINCONT TAPE+ TAPE=
TEST N'5 ARRET

Elles ont construit un programme avec un corps vide (mention d'un avertisseur mais absence du corps de l'itération). Elles vérifient le programme à la main; le considérant comme correct, elles le remettent à nouveau à l'expérimentateur. La vérification de l'expérimentateur porte cette fois-ci sur les instructions CONT et FINCONT; il indique que le robot ne continue rien parce que il n'y a pas d'instructions entre ces deux instructions.

Il y a un nouvel essai ; pour la première fois elles font la simulation au fur et à mesure qu'elles construisent leur programme ; elles ont une discussion sur la signification de l'instruction COMPT. Catherine explicite la signification de la touche COMPT.

C: "dans le compteur c'est marquer 1
donc c'est le premier machin"

P: "alors... bah.... c'est bon COMPT..."

Ensuite, après l'instruction CONT elles écrivent TAPE+ et TAPE= pour répéter seulement l'addition et le résultat.

C: "comme ça il va ajouter tout le temps....
puis après on aura le résultat"

Elles finissent leur programme en mettant TEST N'1. Rien ne nous permet de savoir pourquoi.

Le programme suivant est alors écrit:

(P4): MARCHE PUB LIS TAPE COMPT CONT TAPE+ TAPE=
FINCONT TEST N'1 ARRET

La fin de la séance les contraint à arrêter leur travail.

IV-DEUXIEME SEANCE

IV.1- Fonction des instructions du langage

Le problème nouveau présenté ici réside dans l'utilisation des ardoises, dans le 1^{ère} séance la gestion de variables était masquée par la calculette. L'utilisation de deux ardoises pour écrire trois nombres nécessite la mise en place d'une initialisation différente du corps de l'itération ainsi que l'utilisation d'une place mémoire pour une valeur qui va changer chaque fois.

Les instructions EC1 et EC2 permettent l'écriture des nombres lus et de la somme des nombres écrits sur les ardoises. On s'attend à des difficultés dans la gestion de la place mémoire, un nouveau nombre ne peut être écrit que sur une ardoise effacée à l'aide de la touche EFF1 pour l'ardoise 1 et EFF2 pour l'ardoise 2. Avec l'utilisation de ces instructions les élèves devront effectuer une première gestion (initialisation) des trois nombres différente de celle qui devra être effectuée dans le corps de l'itération.

Les élèves devront lire un premier nombre et l'écrire sur l'ardoise 1, ensuite lire le nombre suivant et l'écrire sur l'ardoise 2, effectuer l'addition de ces deux nombres, effacer l'ardoise 1 et écrire ce troisième nombre obtenu par l'addition sur l'ardoise 1.

Dans le corps de l'itération les élèves devront seulement écrire les instructions de lecture et d'écriture du nombre suivant et l'écriture du nombre résultat de l'addition.

Les considérations faites à propos des autres instructions pour l'analyse a priori de la 1^{ère} séance restent les mêmes pour cette deuxième séance.

IV.2- Analyse des observations :

L'observation du travail des binômes dans cette deuxième séance a mis en évidence de nouvelles difficultés relatives à l'initialisation de l'itération. Cette initialisation constitue une opération difficile même pour des élèves d'un plus haut niveau de connaissance dans la construction des boucles (Samurçay R., 1985) Cependant la gestion des variables (place mémoire) n'a pas soulevé de problèmes grâce à la simulation directe de cette gestion sur les ardoises. La 1^{ère} partie de la 2^{ème} séance (manipulation des ardoises) a certainement contribué aussi à faciliter cette gestion.

La construction d'une itération a été rapidement envisagée par tous les élèves qui avaient terminé correctement la 1^{ère} séance (7 binômes).

Ils ont utilisé le même raisonnement pour construire l'itération que lors de la 1ère séance. Lorsqu'ils se sont aperçus que cette stratégie ne fonctionnait pas pour le problème dans ces nouvelles conditions (avec les ardoises magiques au lieu de la calculette), ils ont effectué un retour aux premières phases de la première séance (juxtaposition d'instructions simples) puis ont procédé à une nouvelle analyse du problème et ont reconstruit l'itération à partir des nouvelles données.

Les solutions élaborées par les binômes dans cette deuxième séance dépendent évidemment de leurs productions lors de la première séance. C'est pourquoi nous rendons compte de leur travail dans la deuxième séance en fonction de leur résultat à la séance précédente :

1-Pour l'ensemble des binômes qui ont terminé la 1ère séance : trois catégories peuvent être distinguées :

1.a. - Ceux qui avaient réalisé le programme final de la 1ère séance sans initialisation (Ariel et Pierre-Yves, Enrique et Olivier, David et Hervé, Esthèle et Nathalie, Lilian et Christophe).

1.b.- Ceux qui avaient fini leur dernière procédure à la première séance en laissant, involontairement, une initialisation (Patrick et Morad, Isabelle et Gabrielle).

Ces derniers ont fait une transposition directe de leur programme, donc leur nouveau programme a bien fonctionné lors de leur propre vérification et de celle de l'expérimentateur.

1.c.- Trois binômes ayant réalisé un programme correct lors de la première séance n'ont pas achevé leur programme lors de la deuxième séance. (Massimo et David, Karine et Delphine, Michel et Frédéric). Ces trois binômes présentent la caractéristique commune d'avoir eu du mal à construire un programme à la première séance.

Revenons à la catégorie 1.a. Dans cette catégorie l'évolution du travail des élèves a connu les phases suivantes :

1ère phase :

Transcription de leur programme final de la 1ère séance en commençant par l'utilisation de l'avertisseur de répétition.

Ex : David et Hervé

Production final 1ère séance :

MARCHE REC TAPE+ PUB LIS TAPE COMPT TEST N° 10
FINREC TAPE = ARRET

1er essai 2ème séance :

MARCHE REC PUB LIS EC1 COMPT PUB LIS EC2 COMPT
ADD12 EF1 EC1 EFF2 TEST N° 150 FINREC ARRET

EX : Ariel et Pierre-Yves

Production finale 1ère séance :

MARCHE REP PUB COMPT LIS TAPE TAPE + TEST N° 150
FINREP TAPE = ARRET

1er essai 2ème séance

MARCHE REP PUB COMPT LIS EC1 PUB COMPT LIS EC2 ADD12
EFF1
EFF2 EC1 TEST N° 150 FINREP ARRET

2ème phase:

La simulation qu'ils effectuent leur montre que ce programme ne peut pas fonctionner parce que à chaque fois, il y a trois nombres à écrire sur deux ardoises (PUB, PUB, ADD12). Ils essaient de modifier leur programme en changeant la place des instructions d'écriture et d'effacement sur les ardoises mais se heurtent à l'obstacle qu'il y aura toujours un nombre qui ne pourra pas être écrit sur une ardoise parce qu'elle n'est pas libre. Les modifications réalisées sur leurs programmes sont de la forme: EC1 par EC2, EFF2 par EFF1 ou vice versa. Après plusieurs essais sans trouver une solution au problème, ils simulent eux mêmes en revenant à la première stratégie utilisée dans la première séance (juxtaposition d'instructions simples) ce qui leur permet de prendre conscience de la nécessité d'une initialisation.

3ème phase:

Cette prise de conscience les conduit à faire un nouvelle analyse du problème

David: "Il faut marquer REP ici,....parce qu'il ne recommencera ça puisque c'est le début"

La solution élaborée repose sur le changement de place de l'avertisseur de répétition.

Exemple David et Hervé

Production finale 2ème séance:

MARCHE PUB COMPT LIS EC1 REC PUB COMPT LIS
EC2 ADD12 EFF1 EFF2 EC1 TEST N°150 FINREC ARRET

2.- Ceux qui n'ont pas terminé le travail à la première séance (4 binômes).

Les problèmes soulevés dans la deuxième séance requièrent un certain niveau qui doit être atteint dans la première. Il est donc normal que les élèves qui ne sont pas arrivés à élaborer un programme correct à la première séance aient eu plus de difficultés que les autres binômes à franchir celles de la deuxième séance.

Stéphane et Nicolas

2ème séance inachevée construction d'un corps sans test ; oubli de mention de l'avertisseur ; (FINREP à la place de REP)

Christelle et Valérie

2ème séance inachevée

Catherine et Pascale ; Jean-François et Anne-Marie n'ont pas participé à la 2ème séance.

	Juxtaposition d'instructions simples	mention de l'avertisseur absence de corps	Corps			
			sans test	non coordination du corps et du test	coordination du corps et du test	
					avec init	sans init
Modalité 1						
Patrick et Morad					avec init 1,2	
Massimo et David	1 →	→ 2 (inachevé)				
Olivier et Enriquer	← 2				→ 3	1 →
Karine et Delphine	1					
Esthèle et Nathalie	← 2				→ 3	1 →
Isabelle et Gabrielle					1,2	
Modalité 2						
Stéphane et Nicolas				1,2,3 (inachevé)		
David et Hervé	← 2				→ 3	1 →
Christelle et Valérie	1	→ 2,3 (inachevé)				
Michel et Frédérick				← 2,3,4		1 →
Ariel et Pierre-Yves	← 2				→ 3	1 →
Lilian et Christophe	← 2				→ 3	1 →

TABLEAU 3 : Succession des types de programmes écrits par les élèves au cours de leur travail dans la 2ème séance

Les programmes sont désignés dans le tableau par leur rang dans l'ordre dans lequel ils ont été écrits

IV.3-Exemples d'élaboration d'un programme

Le cas d'Olivier et Enrique (Modalité 1)

Ils élaborent un programme de la même manière que pour leur production finale de la 1ère séance (catégorie 1.a). Leur première idée consiste à utiliser l'avertisseur de répétition REP :

Olivier : Il faut mettre REP déjà.....

Après l'instruction de mise en marche du robot, ils construisent le corps de l'itération en coordination avec le test d'arrêter de la boucle et pour terminer, ils écrivent le délimiteur FINREP et ARRET.

Le programme est alors écrit :

(P1) : MARCHE REP TEST PUB LIS EC1 PUB LIS EC2 ADD12 EFF1 EC1 EFF2
FINREP ARRET.

Ils jugent leur programme P1 correct et ils le donnent à l'expérimentateur sans faire de vérification.

La simulation faite par l'expérimentateur leur montre que pour le deuxième tour de la boucle le robot ne pourra pas écrire sur l'ardoise le nombre 1 puisqu'il y a déjà un nombre écrit. Ils essaient de changer les instructions EC1 par EC2 ou EFF1 par EFF2. Las d'avoir tenté tous ces changements, ils veulent changer de stratégie et essaient une nouvelle façon de procéder..

E : Il fallait faire encore PUB, LIS.

Enrique revient à la juxtaposition d'instructions simples, il ajoute une nouvelle séquence d'instructions et en même temps il la simule.

Programme Intermédiaire : MARCHE TEST PUB LIS EC1 PUB LIS EC2 ADD12 EC1
EFF2 PUB LIS EC2 ADD12.....

En exécutant sur une liste plus longue de séquence d'instructions, Enrique prend conscience qu'il faut changer la place de l'instruction REP (problème d'initialisation).

E : " Non, c'est pas comme ça.....On ne peut pas mettre REP au début c'est impossible parce qu' après sur 1, il faudra mettre le résultat. Alors c'est pas REP au début, on le mettra à un autre moment, on met REP ici... c'est ça qu'on va répéter."

Leur programme final est alors écrit :

(P2) : MARCHE TEST PUB LIS EC1 PUB LIS EC2 REP ADD12 EFF1
EC1 EFF2 TEST PUB LIS EC2 FINREP ARRET.

Le cas de Karine et Delphine (Modalité 1)

Elles commencent par écrire la liste d'instructions..... Certaines idées retenues dans la première séance (catégorie 1.c) apparaissent tout de suite

D : "Il faut qu'il teste.... Il faut qu'à chaque fois il teste si la boîte est vide ou pas."

Elles continuent leur liste d'instruction et au bout de la deuxième séquence, Karine exprime la nécessité d'utiliser l'avertisseur de répétition.

K : "Attends, attends, il faut mettre REP.... Il faut qu'il répète."

Néanmoins, l'utilisation d'un avertisseur de répétition n'est pas évidente pour Delphine :

D : "Mais non..... là ça va pas, il peut pas parce que regarde, il faut qu'il marque sur l'ardoise pour savoir un résultat, donc..... si à chaque fois il va marquer sur l'ardoise 1....."

Delphine ne voit pas comment, en utilisant l'instruction REP, elle va pouvoir éviter d'écrire sur une ardoise déjà occupée. Elle arrive à convaincre Karine.

K : "Mais oui, s'il fait REP, il va répéter tout ça... et s'il fat sur l'ardoise 1, il faut écrire un seul nombre.. après il va réécrire une autre et il faut marquer qu'un seul chiffre donc on peut pas."

Elles écrivent néanmoins leur longue liste à plusieurs reprises. Karine fait des commentaires pour indiquer le besoin d'utiliser un avertisseur de répétition.

K : "Il faut mettre REP je te dis....on va aller jusqu'à demain."

Au bout d'un moment la liste devient tellement longue qu'elles sont fatiguées d'écrire, elles demandent à l'expérimentateur si elles peuvent simuler avec moins de boules. Le programme reste réduit à la solution pour le nombre de boules que contient la boîte.

Le programme écrit est alors :

(P1) : MARCHE TEST PUB LIS EC1 TEST PUB LIS EC2 ADD12EFF1 EC1 TEST
PUB LIS EFF2 EC2 ADD12 EFF1 EC1 EFF2 TEST PUB LIS EC2 ADD12 EFF1
EC1 EFF2 TEST PUB LIS EC2 ADD12 EFF1 EC1 TEST PUB LIS EC2 ADD12
EFF1 EC1 EFF2 TEST PUB LIS EC2 ADD12 EFF EC1 EFF2 TEST ARRET

Le programme P1 n'a pas pu être vérifié par l'expérimentateur parce que son écriture a nécessité l'heure entière.

Nous n'avons pas présenté des exemples de la modalité 2 parce que les catégorisations faites correspondent aux deux modalités et que les exemples précédents sont les plus parlants.

V-CONCLUSION

De l'observation des conduites des élèves dans la première séance de l'expérience peuvent être dégagés les points suivants qui nous paraissent importants pour la programmation :

- la mise en place de l'itération dans l'écriture d'un programme est effectivement le résultat d'une construction faite d'étapes au cours desquelles la signification et l'expression de l'itération évoluent.
- les instructions qui ont soulevé le plus de difficultés aux élèves sont celles qui nécessitent une articulation avec d'autres instructions ou une coordination entre les composantes internes de l'instruction. Il s'agit des avertisseurs d'itération, des marqueurs de fin d'itération et des conditions d'arrêt.
- le rôle du langage dans l'évolution des productions des élèves est apparu clairement : le passage des programmes avec simple mention de l'itération sans corps d'itération à une forme plus achevée de l'itération a été rendu possible par les contraintes du langage.
- les représentations qu'élaborent les élèves du fonctionnement du robot sont apparues en particulier lors des leurs difficultés à anticiper la vérification de la condition d'arrêt à chaque pas de l'itération.

- Les simulations jouent un rôle important dans la dynamique de la résolution du problème. Elles confirment ou infirment des projets de programme, provoquent des prises de conscience (celle de la condition d'arrêt par exemple). Mais l'effet des simulations faites par l'expérimentateur est plus important que celui des simulations des élèves qui ne peuvent lire leur programme sans prendre en compte le sens qu'ils lui conféraient lorsqu'ils l'élaboraient. La validation apportée par la simulation de l'expérimentateur est plus efficace parce que plus externe.

-La deuxième séance a ajouté des nouveaux problèmes, celui de l'initialisation de l'itération et celui de la gestion des mémoires. Les élèves, face à ces difficultés et après l'échec constaté d'un premier programme transposé directement de la première séance, retournent à leurs premières écritures de programmes sans itération. Ce n'est qu'ensuite qu'ils reconstruisent une itération en fonction des nouvelles données.

-Le langage utilisé a été construit pour les fins de l'expérimentation. Puisque son influence sur l'évolution des programmes des élèves a pu être mise en évidence, nous allons étudier ensuite les effets de variations possibles du langage. C'est pourquoi l'expérimentation suivante se place dans un contexte de communication avec une machine réelle en langage BASIC.

CHAPITRE IV

CONSTRUCTION D'UNE ITERATION DANS UNE SITUATION DE COMMUNICATION ELEVE-MACHINE REELLE

I- Choix de l'expérience

II- Description et déroulement de l'expérience

**III-Analyse a priori de la construction d'un programme
avec itération**

IV-Analyse des observations

V-Exemples de programmes réalisés par les élèves

VI- Conclusions

A la différence des expériences précédentes, ce chapitre concerne le début d'une première expérimentation qui place les élèves dans un contexte de communication avec une machine réelle.

Cette expérience a eu lieu au Collège de Moirans avec des élèves de 3ème dans le cadre d'un atelier d'informatique.

Il est proposé aux élèves d'effectuer un programme qui permet le calcul de la somme de 100 nombres pris au hasard. Il s'agit donc du même problème que pour l'expérience avec le robot (chapitre III), mais cette fois ci les conditions de réalisation du problème sont adaptées aux connaissances des élèves acquises après six séances d'un cours d'introduction à la programmation en langage BASIC avec des micro-ordinateurs THOMSON T07.

I - LES CHOIX DE L'EXPERIENCE :

-Le langage :

Le langage utilisé est le BASIC, en fait nous n'avons pas eu le choix de ce langage : il était imposé par le type de micro ordinateur existant dans les établissements dans lesquels nous pouvions travailler. Néanmoins, ce langage convient très bien à notre propos, étant donné que nous nous intéressons, dans cette expérience,

1° aux difficultés rencontrées par les élèves, pour construire un programme avec itération, lorsqu'ils manient un langage de communication strict et de haut niveau avec l'ordinateur.

2° aux difficultés relatives aux liens entre l'apprentissage de ce langage et la mise en place de l'itération.

-Les conditions de réalisation de l'expérimentation :

Les élèves travaillent par deux. Ils doivent taper sur le micro-ordinateur un programme qui permette le calcul demandé. Leurs échanges sont enregistrés, ce qui nous permet d'obtenir des renseignements plus précis sur le processus de construction de leur programme. Enfin ils peuvent demander l'aide de l'enseignant.

Nous avons choisis ces conditions parce que nous pensons que l'évolution de ce processus est due :

aux interactions entre élèves comme nous l'avons déjà expliqué dans l'introduction de notre travail.

et à la différence de l'expérience précédente:

-à l'interaction avec la machine : lorsqu' un premier programme est tapé à la machine, cette dernière fournit un résultat, ou en cas d'incorrection du programme, répond par des messages d'erreurs de type syntaxique, ou des messages d'erreurs relatifs à l'exécution. On peut penser a priori que la rencontre par les élèves de ces messages les incitera à modifier leur programme.

-à l'interaction avec l'enseignant : il se peut que les interactions entre élèves et avec la machine ne soient pas suffisantes pour qu'ils construisent un programme correct ; en particulier les élèves peuvent ne pas comprendre certains messages d'erreurs. Les explications de l'enseignant sont susceptibles de les aider à comprendre le type d'erreur signalé par la machine ou le motif de non exécution du programme. L'enseignant joue donc un rôle important dans le processus de construction de leur programme.

II - DESCRIPTION ET DEROULEMENT DE L'EXPERIENCE

L'enseignant donne d'abord la consigne du problème à toute la classe et l'explique (cf. annexe 2). On cherche à calculer la somme de 100 nombres pris aléatoirement d'une boîte qui contient 1000 boules numérotées. Ensuite est distribuée la feuille avec le programme suivant qui apparaîtra sur les écrans:

```
0 *****
1 DIM TB (1000)
5 PRINT "POUR CONTINUER ENTREE"
6 A$ = INKEY$
7 A$ = INKEY$ : A = RND : IF A$ = " " THEN 7
8 *****
999 END *****
1000 B = INT (1000*RND (1) + 1)
1010 IF TB (B) = 1 THEN GOTO 1000
1020 TB (B) = 1
1030 RETURN
1040 *****
```

L'enseignant explique que pour tirer un nombre de la boîte il suffit d'appeler le sous-programme en écrivant GOSUB et le N° de ligne où il commence, c'est à dire GOSUB 1000. L'écriture dans le programme de l'instruction GOSUB 1000 permet l'obtention d'un des 1000 nombres de la boîte.

L'enseignant explique les instructions du programme écrit sur la feuille :

- la ligne 1 est la création de la boîte
- les lignes 1000 à 1040 exécutent un sous-programme qui effectue le tirage des boules.
- les lignes 5-6-7 exécutent un remaniement de la boîte (comme le mouvement de boules du loto).

L'enseignant explique que les élèves doivent utiliser les lignes 9 à 998 pour construire leur programme. Ensuite, les élèves sont mis par groupes de 2 pour travailler leur programme avec le micro-ordinateur.

A la différence des expériences précédentes, il est nécessaire pour les élèves d'avoir appris certains éléments de la programmation en langage BASIC et avec le micro-ordinateur THOMSON T07. Nous avons choisi la septième séance pour réaliser cette expérience parce qu'à cette date les élèves disposent des éléments nécessaires à la construction de l'itération, en particulier de l'instruction IF THEN et de l'instruction GOTO (description schématisée du déroulement de chacune des séances précédentes, cf.annexe 2).

III-ANALYSE A PRIORI DE LA CONSTRUCTION D'UN PROGRAMME AVEC BOUCLE

Les résultats des expériences précédentes ainsi que la proximité du langage BASIC des procédés naturels de construction d'une itération nous avaient amené à penser a priori que les difficultés qui apparaîtraient dans l'écriture du programme avec boucle seraient les suivants :

-le comptage des boules tirées et l'addition des nombres : à la différence de l'expérience avec le robot, les élèves sont ici confrontés à de nouvelles difficultés : la mise en place d'une variable cumulative et la notion d'affectation ($S \leftarrow S+B$ pour l'addition, $I \leftarrow I + 1$ pour le comptage).

Ces deux instructions recouvrent deux opérations, l'addition de la valeur de la variable et d'un nombre donné (1 dans le cas du comptage des boules, B dans le cas de l'addition des nombres), et l'affectation du résultat trouvé à la même variable. Une des difficultés de gestion de ces instructions réside dans l'existence de deux valeurs différentes de la même variable à l'intérieur de la même instruction. Contrairement aux usages mathématiques la variable possède ici une signification dynamique.

La variable cumulative pour compter les boules (incrémentations de 1 à chaque fois) ne devrait pas être de la même complexité que la variable cumulative pour additionner les nombres (Samurçay, 1985) parce que cette dernière requiert de plus la manipulation d'une autre variable (nombre tiré).

- l'utilisation d'une instruction de branchement: Lorsque les élèves sont confrontés à la nécessité d'indiquer à la machine l'exécution répétée des instructions déjà écrites, ils devront utiliser l'instruction GOTO. A priori on peut supposer que l'utilisation de cette instruction posera des problèmes, soit parce qu'ils ne penseront pas à l'utiliser pour résoudre le problème, soit parce qu'ils ne sauront pas s'en servir puisque en particulier ils devront indiquer le numéro de ligne sur lequel doit se faire le branchement.

- expression de la condition d'arrêt et coordination avec l'instruction de branchement: L'instruction GOTO ne contient pas en elle même l'expression d'une condition d'arrêt. Cette condition d'arrêt doit s'exprimer à l'aide d'une autre instruction (IF (condition) THEN). On peut, donc, penser que les problèmes apparaîtront d'une part, au niveau du besoin et de la façon d'exprimer l'arrêt (utilisation du IF et expression adéquate de sa condition), et d'autre part lors de la coordination des deux instructions IF et GOTO.

VI - ANALYSE DES OBSERVATIONS :

Nous avons travaillé avec 5 binômes qui sont arrivés à construire un programme dont l'exécution donnait le résultat désiré. Ils ont travaillé environ 2 heures.

L'observation des binômes nous a permis de dégager certaines difficultés dans la construction d'une itération utilisant un langage évolué.

Après avoir essayé des solutions de type arithmétique (ex: multiplication par 100 de la valeur de la première boule tirée) qui se sont révélées inadaptées, les élèves ont tous cherché à construire une itération.

Ce qui a joué le rôle le plus important dans l'évolution de la construction de leur programme a été l'interaction avec l'enseignant. Ce dernier a apporté une aide importante aux élèves pour franchir les obstacles qui les empêchaient d'avancer dans l'écriture de leur programme. En faisant référence à des exercices déjà traités, il a donné des conseils aux élèves qui leur ont permis de sortir des situations dans lesquelles ils se trouvaient arrêtés.

Les démarches suivies par les élèves ont été les suivantes:

La première solution imaginée par les élèves a été de lire toutes les boules pour pouvoir les additionner ensuite. Cette solution est rendue impossible par le fait qu'il faut prévoir une variable pour garder chacun des nombres écrits sur les boules. Les élèves sont ainsi amenés à changer de stratégie.

Le nouveau problème qui apparaît se situe donc au niveau :

- 1) de la construction d'une variable qui garde les valeurs des sommes intermédiaires (variable cumulative),
 - 2) de l'expression de chacune des additions intermédiaires qui modifient la valeur de la variable cumulative (au fur et à mesure que le nombre porté par la dernière boule tirée est ajouté à la somme des précédents (affectation)).
- Nous avons d'ailleurs prévu ce type de problèmes dans l'analyse a priori des difficultés des élèves (cf : analyse a priori)

Ce problème n'est résolu dans la plupart des cas que grâce à l'aide apportée par l'enseignant, qui renvoie à un problème semblable déjà résolu au cours des séances précédentes, ou donne tout simplement la solution (Josiane et Béatrice).

Le problème qui se pose ensuite est de dire à la machine de répéter ces instructions. Contrairement à ce qu'on pouvait penser la possibilité d'utiliser l'instruction GOTO pour résoudre ce problème est apparue en général rapidement et sans l'aide de l'enseignant, les élèves s'étant eux mêmes souvenus de problèmes précédemment résolus. La seule utilisation de l'instruction GOTO a entraîné des difficultés pour la suite de l'écriture du programme; au moment de faire tourner le programme, les élèves se sont aperçus qu'il ne s'arrêtait pas (ou qu'il ne pouvait s'arrêter qu'avec l'aide de la touche Contrôle C).

Le nouveau problème à résoudre est devenu : comment faire pour arrêter le programme ? La solution leur est apparue en deux temps : Il faut pouvoir compter les boules tirées et il faut pouvoir utiliser ce comptage pour arrêter le programme.

Pour pouvoir compter, ils ont rencontré plusieurs problèmes: création d'un compteur à l'aide d'une variable qui sera incrémentée lors de chaque tirage, insertion de l'instruction qui effectue ce comptage à l'intérieur du corps de la boucle. L'idée de la création d'un compteur leur est apparue naturellement mais l'expression de ce compteur et l'insertion de la gestion de ce compteur à l'intérieur de la boucle n'ont pu être faites que grâce aux suggestions de l'enseignant.

Pour pouvoir arrêter le programme, ils ont rencontré les problèmes suivants: expression d'une condition à l'aide de la variable utilisée pour le comptage, insertion de cette condition à l'intérieur de l'instruction IF, coordination de l'instruction IF avec l'instruction de branchement GOTO.

L'utilisation de l'instruction IF pour arrêter la boucle a été dans un premier temps dissociée de l'instruction GOTO (celle-ci occupait une autre ligne

dans leurs programmes) et liée à l'instruction STOP (IF Condition THEN STOP). La liaison de ces deux instructions n'a pu être effectuée dans un deuxième temps (IF condition THEN GOTO) que grâce à l'aide apportée par l'enseignant.

V - EXEMPLES DE PROGRAMMES REALISES PAR LES ELEVES

V.1-Le cas de Josiane et Béatrice:

La première difficulté rencontrée est l'utilisation de l'instruction GOSUB 1000 pour tirer les boules. L'observateur a dû réexpliquer la fonction de GOSUB. Lorsqu'elles comprennent qu'avec cette instruction elles peuvent tirer les boules, la nouvelle difficulté est relative au tirage de 100 boules. Elles proposent de recopier l'instruction GOSUB 100 fois.

B: "Alors, il va falloir faire comme ça (écrire GOSUB 1000) 100 fois si l'on veut qu'elle nous les sorte toutes !"

Après avoir constaté qu'écrire 100 fois GOSUB 1000 est un long travail, de nouvelles suggestions apparaissent.

B : "écris 100 x GOSUB 1000 ou
J : " tu fais GOSUB 1000 = A et après A x 100

Comme elles n'arrivent pas à trouver une solution, elles appellent l'enseignant. Celui-ci leur suggère de faire l'addition de 2 boules et il leur suggère aussi d'afficher le nombre pris (PRINT B, B étant, dans le programme, la variable qui contient le nombre tiré).
A ce moment leur programme est le suivant:

```
10 GOSUB 1000  
20 PRINT B
```

La difficulté d'addition des nombres subsiste encore. Josiane manifeste son embarras parce qu'un nouveau nombre écrit sur B va effacer l'ancien, elle ne pourra donc jamais les additionner. Elles ont recours à nouveau à l'enseignant qui leur donne la solution : Il écrit $C = C + B$ (C étant la variable choisie pour garder la somme).

Une solution étant partiellement apportée au problème des additions, la nouvelle difficulté est de faire ces additions 100 fois. Elles retiennent différentes solutions.

B : "J'ai compris.....Il faut mettre 30 IF C = C + B THEN GOTO 10."

La proposition de Béatrice n'est pas comprise par Josiane qui insiste en disant qu'il faut dire à la machine de faire cela 100 fois. Béatrice lui explique qu'avec cette ligne la machine va s'arrêter dès qu'elle aura additionné 100 boules. Pour Josiane c'est plutôt IF C = C+B THEN STOP qui permettra cet arrêt.

Elles aboutissent au programme suivant :

```
10 GOSUB 1000
20 PRINT B
30 IF C = C + B THEN GOTO 10
```

Elle le font tourner et le programme ne s'arrête pas. La nouvelle difficulté est alors : comment faire pour arrêter le programme ? Elles font appel à l'enseignant qui suggère de créer une boîte pour compter les boules en expliquant sa construction : $D = D + 1$ (D est la variable choisie pour compter le nombre de boules tirées). Il leur fait changer également la ligne 30 par $C = C + B : GOTO 10$. Josiane insiste qu'il faut s'arrêter après avoir compté 100 boules, en écrivant la ligne 40 IF D = 100 THEN STOP. Pour elles c'est bien la condition d'arrêt de la boucle mais elles ne l'insèrent pas dans le corps de la boucle.

Leur nouveau programme est alors :

```
10 GOSUB 1000
15 D = 1 + D
20 PRINT B
30 C = C + B : GOTO 10
40 IF D = 100 THEN STOP
```

Elles font tourner le programme. Il tourne indéfiniment. Elles appellent à nouveau l'enseignant qui leur explique que l'instruction 40 n'est pas dans la boucle et il leur suggère de la mettre dedans en écrivant : 25 IF D = 101 THEN STOP. Elles terminent leur programme en affichant la somme.

Le programme final est donc :

```
10 GOSUB 1000
15 D = 1 + D
20 PRINT B
25 IF D = 101 THEN STOP
30 C = C + B : PRINT " SOMME" C : GOTO 10
35 PRINT C
40 IF D = 100 THEN STOP
```

L'exécution de ce programme sur l'écran donne la valeur des boules tirées et la valeur des sommes intermédiaires, la somme finale étant affichée lors du dernier tirage. Les lignes 35 et 40 ne sont jamais exécutées dans ce programme mais Josiane et Beatrice ne l'ont pas remarqué puisque leur programme affiche les résultats attendus.

V.2-Le cas de Brigitte et Florence :

De la même manière que dans le cas précédent, leur première idée de solution est de faire 100 fois GOSUB 1000 pour sortir les 100 nombres et les additionner ensuite. Elles essaient d'écrire GOSUB 100 fois.

B : " Il faut que la ligne 10 tu l'écrives 100 fois... et tu les additionnes après."

Florence donne une autre suggestion :

F : "Il faut écrire GOSUB 1000 x 100"

Brigitte suggère alors d'écrire PRINT B pour connaître le nombre tiré (c'est sur la variable B que le sousprogramme GOSUB sort la valeur du nombre tiré) et d'ajouter ce nombre à 100.

Leur premier essai de programme est :

```
10 GOSUB 1000
20 PRINT B
30 B + 100
```

Elles font tourner le programme et celui-ci ne donne pas l'addition des 100 boules. Elles font alors appel à l'enseignant. Celui-ci leur explique l'utilisation d'une variable qui leur permettra de conserver la somme chaque fois qu'elles tirent une nouvelle boule en suggérant à Brigitte de changer la ligne 30 par $S = S + B$.

Elles modifient leur programme :

```
10 GOSUB 1000
20 PRINT "S =SOMME"
30 S = S + B
```

Lorsqu'elles le font tourner, rien n'apparaît sur l'écran.

Pour voir la somme à l'écran elles ajoutent la ligne 40 avec l'instruction PRINT S.

Le problème de l'addition d'une nouvelle boule est résolu mais elles n'arrivent toujours pas à additionner exactement 100 boules, elles vont chercher à surmonter cette nouvelle difficulté.

Elles font des suggestions :

F : "Il faut écrire $S = S + 100$ B"

B : "Il faut continuer à écrire GOSUB 1000".

La solution n'apparaît pas. Elles ont alors recours à des problèmes déjà résolus sur des feuilles distribuées par l'enseignant, ce qui leur donne l'idée du GOTO. Pour Brigitte le GOTO ne s'écrit pas tout seul. Il faut l'écrire avec un IF, elle se rappelle qu'il faut écrire SI (quelque chose) alors GOTO 30. Mais elle n'arrive pas à se le rappeler et elles décident d'écrire une nouvelle ligne 50 GOTO 30 pour répéter l'addition.

Lorsqu'elles font tourner le programme celui-ci ne s'arrête pas. Elles doivent utiliser la touche CONTROL C pour l'arrêter. La nouvelle difficulté rencontrée est alors : comment arrêter l'addition après avoir tiré exactement 100 boules ?

Florence suggère d'utiliser STOP mais elle ne sait pas comment. Brigitte pense qu'il faut écrire IF $S = 100$ THEN GOTO 30. Elles font appel à l'enseignant qui leur suggère de créer un compteur, cette modification est reprise par Brigitte :

B : "Si regardes....maintenant il nous faut une boîte... par exemple qui compte les boules pour arriver à 100.... il nous faut une boîte, un compteur"

F : "Alors...écris 60 PRINT C"

B : "Tu vois par exemple si dans C il y a plus de 100 boules il faut qu'elle s'arrête "

Alors elles écrivent

```
70 IF C = 100 THEN STOP
```

A ce stade le programme est :

```
10 GOSUB 1000
20 PRINT "S = SOMME"
30 S = S + B
40 PRINT S
50 GOTO 30
60 PRINT C
70 IF C = 100 THEN STOP
```

Elles le font tourner et il ne s'arrête pas. Elles appellent à nouveau l'enseignant. Celui-ci leur fait remarquer que premièrement il faut faire GOTO 10 pour prendre de nouvelles boules, deuxièmement que le programme ne s'arrête pas parce qu'on n'arrive jamais à la ligne 70 et troisièmement qu'il apparaît 0 à chaque fois qu'elles impriment C parce qu'il n'y a rien dans cette boîte.

Grâce à cette aide elles font des modifications à leur programme. Elles ajoutent la ligne 45 PRINT B pour connaître le nombre de la boule tirée, elles changent la ligne 50 par la ligne 80 et écrivent GOTO 10. Néanmoins, le problème du compteur n'est résolu qu'avec l'aide d'un camarade qui leur dit d'écrire $C=C + 1$. Elles écrivent à la ligne 60 $C = C + 1$ au lieu de PRINT C.

Au moment de faire tourner le programme elles s'aperçoivent qu'elles n'ont pas la somme. Elles ajoutent 100 PRINT "S = SOMME TOTALE" ; 110 PRINT S et changent la ligne 70 par 70 IF C > 100 THEN GOTO 100

Le programme final est alors :

```
8 *****
10 GOSUB 1000
20 PRINT " S = SOMME"
30 S = S + B
40 PRINT S
45 PRINT B
60 C = C + 1
70 IF C > 100 THEN GOTO 100
80 GOTO 10
100 PRINT " S = SOMME TOTALE"
110 PRINT S
999 END *****
```

L'exécution de ce programme sur l'écran donne pour chaque tirage l'affichage d'une formule de présentation (S = SOMME) et les valeurs de la somme intermédiaire et de la dernière boule tirée.

A la fin est affichée la formule S= SOMME TOTALE suivie du résultat final.

A la suite nous montrons les programmes définitifs des autres trois binômes.

CHRISTOPHE et DAVID

```
8 *****
10 X = X + 1
20 GOSUB1000 : PRINTUSING "####" ; B;
30 A = A+B
60 IFX < 100 THEN GOTO 10
70 PRINT " LA SOMME EST" A
999 END*****
```

JO et MARYLENE

```
8 *****
10 GOSUB 1000
20 PRINTUSING "####" ; B;
30 S = " SOMME"
40 S = B+S
50 C = C+1
60 IF C = 100 THEN GOTO 70
65 GOTO 10
70 PRINT S
999 END *****
```

FABRICE ET STEPHANE

```
8 *****
10 IF x > 100 THEN GOTO 50
15 GOSUB 1000 : PRINTUSING "####" ; B ;
20 C = C+B
30 X = X+1
40 GOTO 10
50 PRINT : PRINT : COLOR 2 : PRINT C : COLOR 1
999 END *****
```

VIII - CONCLUSION

- Pour arriver à construire l'itération comme solution au problème les élèves sont passés par les étapes suivantes :

1-ils ont essayé d'écrire plusieurs fois les instructions qui permettent le tirage des boules pour les additionner ensuite.

2-ils ont dû passer par la création d'une variable cumulative (affectation) chaque fois qu'ils tirent des boules.

3-à cause du travail fastidieux (écriture d'instructions plusieurs fois) ils ont eu recours au GOTO,

4-l'utilisation d'un branchement inconditionnel (GOTO sans condition d'arrêt) a pour effet que le programme tourne indéfiniment. Cela les a amenés à chercher une solution pour arrêter le programme

5-ils ont dû créer, pour arrêter l'itération, un compteur qui devait être évalué lors de chaque exécution du corps de l'itération.

6- Ils ont dû lier l'instruction IF, utilisant une condition portant sur la valeur du compteur , avec l'instruction de branchement GOTO.

- En général, le passage d'une étape à l'autre n'a pu être franchi que grâce à l'intervention de l'enseignant. Ni l'interaction entre les élèves, ni l'interaction avec la machine n'ont contribué à l'évolution de la construction du programme, contrairement à ce que nous avons pensé.

-Nous n'avons pas pu observer de difficultés relatives à l'initialisation de la boucle. En effet, les élèves n'ont pas eu besoin d'initialiser les variables à utiliser dans le corps de l'itération (en BASIC une variable à laquelle on n'a pas assigné de valeur vaut zéro).

- Comme nous l'avions prévu à priori, les élèves ont rencontré des difficultés pour exprimer l'addition des boules, le comptage des boules et pour mettre en relation la condition d'arrêt et l'instruction de branchement. Cependant, contrairement à ce que l'on pensait, ils n'ont eu que relativement peu de difficultés à utiliser l'instruction GOTO pour indiquer la répétition.

La difficulté la plus difficile à franchir a été l'utilisation de la variable compteur à l'intérieur du corps de l'itération.

-Dans cette expérience, la construction de l'itération a présenté plus de difficultés que dans les expériences précédentes du fait de l'utilisation des structures imposées par le langage BASIC et du fait que la gestion des variables revenait ici aux élèves alors que dans la modalité 1 de l'expérience avec le robot elle n'existait pas et que dans la modalité 2 elle était plus facile à exprimer grâce au langage ad hoc du robot.

CONCLUSION

**I- Aspects communs dans la construction de l'itération
aux trois expérimentations.**

**II- Différences dues aux caractéristiques des langages
et des situations de programmation.**

Les situations expérimentales présentées dans les chapitres précédents ont confronté les élèves à un problème d'écriture d'un algorithme itératif dans trois contextes de programmation différents. La comparaison de la construction de l'itération par les élèves dans ces trois contextes expérimentaux permet de mettre en évidence des caractéristiques de cette construction communes aux trois situations. Elle permet aussi de repérer les variations que présentent les productions des élèves et leur élaboration d'un contexte à l'autre. Nous cherchons à analyser ces variations en fonction des variations du contexte.

I- ASPECTS COMMUNS DANS LA CONSTRUCTION DE L'ITERATION AUX TROIS EXPERIMENTATIONS

On peut dégager une suite d'étapes qu'a suivie en général la construction de l'itération par les élèves. Le déroulement même des étapes et la rapidité du passage d'une étape à la suivante ont différé suivant les expérimentations.

-Découpage en actions élémentaires: un premier résultat marquant concerne le découpage en actions élémentaires. Il n'a pas eu lieu de façon spontanée dans les deux cas où il n'était pas imposé par le langage de communication avec la machine. En effet, le seul découpage immédiat en actions élémentaires a eu lieu dans le cas de la communication avec le robot où elles étaient codées sous forme d'instruction du langage.

Les premières tentatives d'écriture de l'algorithme sont globales. On écrit une seule suite d'additions pour exprimer l'addition répétée du nombre inscrit sur la calculette A (chapitre II, cas du groupe 3 qui a écrit : -additionner à x sa valeur y fois comme suit : $x+x+x+x+\dots+x$
y fois)

ou on tire toutes les boules et on réalise ensuite l'addition des nombres inscrits sur les boules (cas de Brigitte et Florence, dans le chapitre IV où elles proposent d'écrire 100 fois l'instruction GOSUB 1000 pour sortir les 100 nombres et les ajouter ensuite).

Les découpages en actions élémentaires nécessitent une analyse du problème qui doit prendre en compte les compétences de l'opérateur. Il s'agit en même temps de déterminer et de découper en opérations permises le processus que devra effectuer l'opérateur pour passer d'un état initial à un état final. C'est une opération conceptuelle complexe qui se fait tout au long de l'élaboration de l'algorithme et non au seul commencement du travail.

Les représentations que l'élève programmeur se fait du fonctionnement de l'opérateur jouent évidemment un rôle important dans la prise de conscience que l'opérateur ne sait faire que certaines opérations bien déterminées. Cette prise de conscience n'est pas immédiate et le premier comportement des élèves au début de l'activité consiste à attribuer à l'opérateur les compétences qu'ils possèdent eux-mêmes.

-Construction d'une liste d'instructions à répéter : Cette opération de découpage en actions élémentaires partielles a donné lieu, dans tous les cas, à une liste ordonnée d'instructions à répéter, mais qui n'a pas été complète d'emblée : il a toujours manqué des instructions qui ont été masquées par la façon dont l'élève a découpé l'algorithme, et par le fait que certaines nécessités de la réalisation de la tâche n'apparaissent que plus tard dans la construction de la solution. C'est un trait commun à toutes les expériences, indépendamment du langage utilisé : en BASIC, l'instruction ($c = c+1$) pour effectuer le comptage n'est apparue qu'après plusieurs découpages possibles, et manifestement à l'étape où est apparu le besoin d'arrêter la répétition, donc, de compter le nombre d'additions réalisées. De même, c'est à cette même étape qu'est apparue dans l'expérience du robot, l'insertion de la touche TEST N°.

-L'usage d'avertisseurs de répétition : les contraintes du problème posé (grand nombre ou nombre inconnu a priori d'actions à répéter) ont conduit, dans les trois expériences, à une prise de conscience de la nécessité de répéter la liste d'instructions (impliquée par le découpage de l'algorithme). Il s'ensuit l'utilisation d'un avertisseur qui n'est pas conditionnel ; l'élève se rend compte de la nécessité de répéter les actions partielles, mais la condition d'arrêt n'est pas exprimée. Ce dernier aspect n'est pas contradictoire avec l'existence d'une prise de conscience de la nécessité d'arrêter la répétition, mais cette idée reste implicite ; on peut penser que les élèves attribuent à la machine une intelligence autonome qui permettra d'arrêter la répétition au bon moment.

Ce caractère implicite de la condition d'arrêt était apparu dans le cas des questionnaires (chapitre I) où une quantité considérable d'élèves choisissent une expression répétitive sans condition d'arrêt pour solution d'un problème itératif.

-La mise en place de la condition d'arrêt : les caractéristiques du dispositif expérimental et particulièrement la validation de l'algorithme ont contribué à ce que la nécessité d'explicitement une condition d'arrêt s'impose (la répétition infinie de l'exécution en BASIC, la simulation de l'expérimentateur, à la place du robot, d'une répétition qui ne s'arrête jamais).

C'est à cette étape que les élèves ont exprimé explicitement cette condition. Cette explicitation a d'abord été mise en place sans tenir compte du fait qu'elle doit être vérifiée à chaque répétition. Là encore, tout se passe comme si les élèves attribuaient à la machine la capacité intrinsèque de cette vérification. On a pu le voir par exemple dans l'écriture d'un algorithme en langue naturelle où les algorithmes écrits supposaient que la machine (opérateur élève) possédait des compétences non prévues par le dispositif et que la vérification se réalisait à la fin de la répétition.

De même dans l'expérience de communication avec le robot, l'instruction TEST a en général été placée dans l'algorithme après plusieurs simulations montrant que ce dernier bouclait. Souvent elle a été mise dans un premier temps en dehors du corps de l'itération après l'itération elle-même, comme si la vérification de la condition ne se faisait qu'une fois la répétition terminée.

Ces difficultés rencontrés par les élèves, dans deux étapes concernant l'usage d'avertisseurs de répétition et la mise en place de la condition d'arrêt peuvent se rapprocher des stratégies erronées détectés par Pea (1984) dans les expériences qu'il a menées dans d'autres contextes et avec d'autres langages, stratégies développées par les élèves, comme si la machine possédait des puissances interprétatives et intelligents.

II- DIFFERENCES DUES AUX CARACTERISTIQUES DES LANGAGES ET DES SITUATIONS DE PROGRAMMATION

- Suivant les dispositifs expérimentaux que nous avons utilisés, l'opérateur chargé de l'exécution du programme a pu être un autre élève, un opérateur humain simulant un robot ou un micro-ordinateur. De part les compétences attribuées par les élèves à ces opérateurs, des différences importantes sont apparues dans la construction des algorithmes. La communication en langue naturelle, suivant en cela l'usage commun, laissait une plus grande place à des implicites que la communication avec le robot (simulé) ou avec le micro-ordinateur ne tolère pas. Ainsi, un compteur n'a été construit que par un seul binôme d'élèves dans l'algorithme d'addition réitérée en langue naturelle (chapitre II), le comptage étant laissé naturellement à l'élève opérateur chargé d'exécuter l'algorithme.

- La présence ou l'absence de feedbacks sur les productions des élèves a lui aussi joué un rôle important. Comme on pouvait s'y attendre, ceux dont l'origine est extérieure aux élèves (simulation du robot, ou exécution par le

micro-ordinateur) ont eu un rôle moteur dans l'évolution des algorithmes construits. Mais la simulation des programmes par les élèves eux-mêmes a été sans effet, car de façon naturelle ils mettaient en œuvre leurs conceptions du fonctionnement défini par la consigne.

Cependant, les feedbacks apportés par la machine elle-même ne sont pas toujours féconds. Dans le cas où ils consistent en un calcul qui boucle indéfiniment, l'interprétation par les élèves de cette situation (inattendue pour eux) les a conduits à reconsidérer leur algorithme en posant le problème de l'arrêt de l'itération. Mais lorsque ces feedbacks consistent en messages d'erreur, les élèves peuvent très rarement leur donner une signification leur permettant d'évoluer. Surtout dans les cas de premières tentatives, où la distance à un algorithme effectivement exécutable peut être très grande. Ici l'enseignant a joué un rôle décisif en expliquant à ces élèves la signification du message d'erreur, voire le fonctionnement du micro-ordinateur.

- l'ensemble des instructions disponibles et la structure des langages, en interaction avec les conceptions qu'ont les élèves des compétences de l'opérateur, tiennent une place déterminante soit par la disponibilité de certaines instructions (par exemple : COMPT,...), soit par la présence de structures de contrôle évoluées.

Cependant, cette présence ou cette disponibilité ne suffisent pas pour que d'emblée les élèves utilisent ces outils dans leur construction de l'algorithme comme on a pu le voir dans le cas de l'expérience avec le robot, où le programme définitif a été élaboré après plusieurs essais et refontes des programmes intermédiaires.

- Aussi dans le cas de l'utilisation du langage BASIC comme de la langue naturelle, le compteur n'apparaît pas immédiatement. D'abord parce que les élèves considèrent implicitement que l'opérateur (le micro-ordinateur ou l'opérateur humain) utilisera automatiquement un compteur. Mais même lorsque le besoin en sera reconnu, la construction par les élèves de ce compteur présente beaucoup de difficultés. Dans le cas de BASIC l'intervention de l'enseignant est nécessaire, en particulier pour gérer une variable cumulative.

- L'itération, en BASIC, ne pouvait être obtenue qu'à l'aide d'un branchement éventuellement associé à une instruction conditionnelle. Cela a nécessité une construction complexe, notamment pour la délimitation du corps de l'itération.

En langue naturelle en revanche, les possibilités offertes d'une expression moins explicite, ont largement occulté ces difficultés : le contrôle de l'itération était à la charge de l'opérateur.

Le robot présente un couple d'instructions (par exemple REP, FINREP) dont l'utilisation conjointe permet d'indiquer une itération et d'en délimiter le corps. Cette possibilité, sans offrir le cadre d'une structure forte (comme par exemple REPETER... JUSQU'A CE QUE), permet cependant, par ses mnémoniques évoquant la répétition, de marquer l'existence d'une structure itérative en tant que telle. En fait l'utilisation de ce couple d'instructions n'a été possible qu'au terme d'un travail important de construction dans lequel FINREP a pu n'être associé à REP qu'après coup. De plus FINREP, compris comme "fin de répétition" a constitué dans certains cas un obstacle à la mise en place d'une condition d'arrêt.

Ces résultats soulèvent la question de ce qui aurait pu être fait par les élèves en ayant les contraintes d'un langage structuré du type Pascal : la délimitation du corps de l'itération et l'expression de la condition organiquement liées à l'instruction While... Leur traitement dès les premiers moments de la construction d'une itération paraît inévitable, au moins en première analyse. En fait ce que nous avons obtenu montre que l'actualisation des valeurs en jeu dans la condition au cours du calcul, ne va pas de soi. Le problème de la signification de cette condition reste entier, elle ne peut être que le résultat d'une construction que les contraintes de structuration d'un langage ne garantissent pas. Il y a là un développement nécessaire à notre travail dans un avenir immédiat.

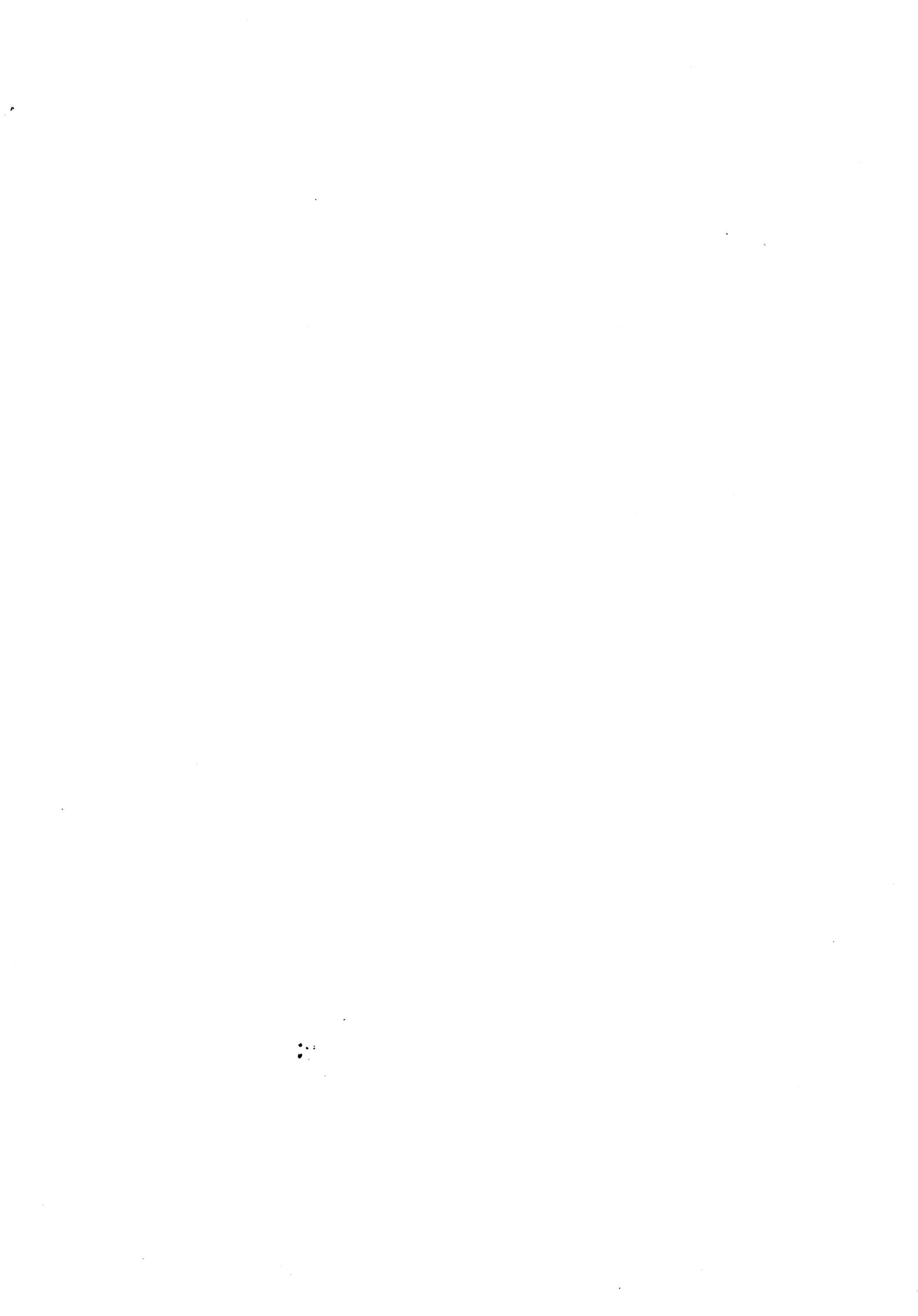
BIBLIOGRAPHIE

- ARSAC J. (1982) : Premières leçons de programmation - Paris : CEDIC .
- ARSAC J. (1983) : Les bases de la programmation - Paris : Bordas.
- BACKUS J. (1978) : Can programming be liberated from the von Neumann Style? A functional style and its algebra of programs : Communications of ACM - Volume 21, number 8.
- BÖHM C. JACOPINI G. (1966) : Flow Diagrams Turing Machines and Languages with only two Formulation rules. Communication of the ACM Vol. 9 n° 5 pp. 366-371.
- BONAR J., SOLOWAY E. (nov. 1982) : Uncovering Principles of Novice Programming : Research Report n° 240 - University of Massacusetts.
- BROUSSEAU G. (1976) : Les obstacles épistémologiques et les problèmes en mathématiques : Recherches en Didactique des Mathématiques - Vol 4, n° 2 (1983).
- DIJKSTRA E.W. (1968) : GO TO Statement Considered Harmful : Communications of the ACM - Vol 11, n° 3 , pp. 147-148.
- DIJKSTRA E.W. (1972) : Notes on Structured Programming. in Dahl Dijkstra et Hoare: Structured Programming - New-York : Academic Press.
- DOISE W., MUGNY G. (1981) : Le développement social de l'intelligence - Paris : Inter Editions.
- HOC J.M. (1979) : Le problème de la planification dans la construction d'un programme informatique : Le Travail humain - 42 , 2 , pp.245-260.
- HOC J.M. (1982) : L'étude psychologique de l'activité de programmation, une revue de la question : Technique et Science Informatique - 1982 , 1 5, pp.383-392.

- KNUTH D.E. (1974) : Structured Programming with GO TO statement : ACM Computing Surveys - Vol 6, n°4, pp. 261-301.
- LEGARD F. MARCOTTY M. (1975) : A genealogy of control structures : Communications of the ACM - Vol 18, n° 11, pp. 629-639.
- PEA R.D. (1984) : Language-Independent Conceptual "bugs" in Novice Programming : Technical Report n°31 . New-York : Bank street college of education center for children & technology
- PERRET-CLERMONT A.N. (1979) : La construction de l'intelligence dans l'interaction sociale - Berne : P. Lang.
- ROGALSKI J. (1984) : Enseignement de l'informatique en seconde: une expérience. Séminaire de didactique des mathématiques et de l'informatique n° 50 IMAG
- ROGALSKI J. (1985) : Alphabétisation informatique : Problèmes conceptuels et didactiques. APMEP bulletin n° 347 pp 61 - 74
- ROUCHIER A., SAMURCAY R. (1984) : Concepts informatiques et programmation, Une première analyse en classe de seconde des lycées : Rapport de recherche, Projet E.A.O. 696 A.B.C. C.N.R.S.-E.H.E.S.S.et I.R.E.M. d'Orleans.
- SAMURCAY R. (1985) : Learning Programming : An Analysis of looping strategies used by beginning students : For the Learning of Mathematics - 5-1. PLM Publishing Association, Montréal Québec, Canada.
- SCHOLL P.C. (1979) : Vers une programmation systématique : étude de quelques méthodes techniques et outils : Thèse d'état, Grenoble - Université 1.
- SOLOWAY E., EHRLICH K., BONAR J., GREENSPAN J. (1982 a) : What do novices know about programming ? : In B. Schneiderman and A. Badre (Eds) Directions in Human-Computer Interactions Ablex Publishing Co.
- SOLOWAY E., BONAR J., EHRLICH K. (1982 b) : Cognitive strategies and looping constructs : an empirical study à paraître in the Communications of the ACM.

TENNENT R.D. (1981) : Principles of programming Languages
Londres : Prentice/Hall.

VERGNAUD G.(1981): Quelques orientations théoriques et
méthodologiques des recherches françaises en didactique des
mathématiques : Recherches en Didactique des Mathématiques -
Vol. 2, n°2 , pp. 215-229.



ANNEXE 1

EXPERIENCE DE CONSTRUCTION D'UNE ITERATION DANS UNE COMMUNICATION AVEC UN ROBOT VIRTUEL (CHAP III).

Texte, consignes, schémas des claviers et signification de touches distribués aux élèves.

1ère SEANCE

TEXTE FOURNI POUR LA MODALITE 1

On dispose d'une boîte avec des boules numérotées. On ne sait pas combien il y a de boules dans la boîte.

On désire connaître la somme de tous les nombres écrits sur les boules de la boîte.

On donne des ordres à un robot pour qu'il effectue cette somme.

Le robot est manipulé par un clavier à touches. On vous donne la liste des noms de touches et leur signification.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES NOMBRES SUR LES BOULES.

SCHEMA DU CLAVIER :

MARCHE	PUB	LIS	TAPE
TEST	REP	REC	CONT
FINREP	FINREC	FINCONT	TAPE +
TAPE =	ARRET		

SIGNIFICATIONS DES TOUCHES:

MARCHE : Si on appuie cette touche le robot est branché

PUB : Il prend une boule

LIS : Il lit le nombre qui est tracé sur la boule qu'il vient de prendre

TAPE : Il tape sur la calculette le nombre qu'il vient de lire

TEST : En appuyant cette touche le robot teste si la boîte est vide ou non. Dans le cas où elle est vide, il interrompt son exécution mais il reste branché. Dans le cas où elle n'est pas vide, il continue

REP : Il répète indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REP

REC : Il recommence indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REC

CONT : Il continue indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche CONT

FINREP : Marque la fin de la liste des ordres à répéter

FINREC : Marque la fin de la liste des ordres à recommencer

FINCONT : Marque la fin de la liste des ordres à continuer

TAPE + : Il tape sur la touche + de la calculette

TAPE = : Il tape sur la touche = de la calculette pour obtenir un résultat

ARRET : En appuyant cette touche on débranche le robot

TEXTE FOURNI POUR LA MODALITE 2 :

On dispose d'une boîte avec 1000 boules numérotées de 1 à 1000.

On désire connaître la somme des nombres écrits sur 150 des boules de la boîte.

On demande à un robot d'effectuer la somme des nombres tracés sur 150 boules qu'il prend au hasard dans la boîte.

Le robot est manipulé par un clavier à touches.

On vous donne les noms des touches et leurs significations.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES 150 NOMBRES ECRITS SUR LES BOULES ;

SCHEMA DU CLAVIER

MARCHE	PUB	LIS	TAPE			
COMPT	REP	REC	CONT			
FINREP	FINREC	FINCONT	TAPE +			
TAPE =	TEST N°	ARRET				
0	1	2	3	4	5	6
7	8	9				

SIGNIFICATIONS DES TOUCHES :

MARCHE : Si on appuie cette touche le robot est branché

PUB : Il prend une boule

LIS : Il lit le nombre qui est tracé sur la boule qu'il vient de prendre

TAPE : Il tape sur la calculette le nombre qu'il vient de lire

COMPT : Le robot a un compteur qui augmente de 1 chaque fois qu'on appuie sur la touche COMPT

REP : Il répète indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REP

REC : Il recommence indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REC

CONT : Il continue indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche CONT

FINREP : Marque la fin de la liste des ordres à répéter

FINREC : Marque la fin de la liste des ordres à recommencer

FINCONT : Marque la fin de la liste des ordres à continuer

TAPE + : Il tape sur la touche + de la calculette

TAPE = : Il tape sur la touche = de la calculette pour obtenir un résultat

TEST N° : Si on appuie sur la touche TEST N° suivie d'un nombre, le robot regarde si la touche COMPT a le même nombre. Si la touche COMPT a le même nombre il arrête l'exécution, sinon il continue.

ARRET : En appuyant cette touche on débranche le robot.

2ème SEANCE

TEXTE FOURNI POUR LA 1ère PHASE

On désire effectuer l'addition suivante :

$$24 + 147 + 33 + 49 + 110$$

On donne des ordres à un robot pour qu'il effectue cette somme.

Il va se servir de deux ardoises magiques pour se souvenir des nombres enregistrés dans sa mémoire.

Le robot est manipulé par un clavier à touches.

On vous donne la liste des noms des touches et leur signification.

Dans chaque ardoise on ne peut écrire qu'un seul nombre à la fois.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE L'ADDITION ECRITE AU-DESSUS.

SCHEMA DU CLAVIER :

MARCHE	LIS	EC 1	EC 2			
EFF 1	EFF 2	ADD 12	ARRET			
0	1	2	3	4	5	6
7	8	9				

SIGNIFICATIONS DES TOUCHES :

MARCHE : Si on appuie cette touche le robot est branché.

LIS : Le robot lit le nombre sur le papier.

EC 1 : Il écrit sur l'ardoise N° 1

EC 2 : Il écrit sur l'ardoise N° 2

ADD 12 : Il ajoute le nombre de l'ardoise N° 1 à celui de l'ardoise N° 2.

EFF 1 : Il efface l'ardoise N° 1.

EFF 2 : Il efface l'ardoise N° 2.

ARRET : En appuyant cette touche, on débranche le robot.

TEXTE FOURNI POUR LA MODALITE 1 DE LA DEUXIEME PHASE :

On dispose d'une boîte avec des boules numérotées. On ne sait pas combien il y a de boules dans la boîte.

On désire connaître la somme de tous les nombres écrits sur les boules de la boîte.

On donne des ordres à un robot pour qu'il effectue cette somme.

Le robot est manipulé par un clavier à touches.

On vous donne les noms des touches et leur signification.

Le robot a devant lui la boîte avec des boules et deux ardoises magiques dans lesquelles il va écrire des nombres. Il ne peut écrire qu'un seul nombre à la fois dans chaque ardoise.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES NOMBRES SUR LES BOULES.

SCHEMA DU CLAVIER:

<i>MARCHE</i>	<i>PUB</i>	<i>LIS</i>	<i>TEST</i>
<i>REP</i>	<i>REC</i>	<i>CONT</i>	<i>EC 1</i>
<i>EC 2</i>	<i>EFF 1</i>	<i>EFF 2</i>	<i>ADD 12</i>
<i>FINREP</i>	<i>FINREC</i>	<i>FINCONT</i>	<i>ARRET</i>

SIGNIFICATIONS DES TOUCHES :

MARCHE : Si on appuie cette touche le robot est branché.

PUB : Le robot prend une boule.

LIS : Il lit le nombre qui est tracé sur la boule qu'il vient de prendre.

TEST : En appuyant cette touche le robot teste si la boîte est vide ou non. Dans le cas où elle est vide, il interrompt son exécution mais il reste branché. Dans le cas où elle n'est pas vide, il continue.

REP : Il répète indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REP.

REC : Il recommence indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REC.

CONT : Il continue indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche CONT.

EC 1 : Il écrit sur l'ardoise N° 1.

EC 2 : Il écrit sur l'ardoise N° 2.

EFF 1 : Il efface l'ardoise N° 1.

EFF 2 : Il efface l'ardoise N° 2.

ADD 12 : Il ajoute le nombre de l'ardoise N° 1 à celui de l'ardoise N° 2.

FINREP : Marque la fin de la liste des ordres à répéter.

FINREC : Marque la fin de la liste des ordres à recommencer.

FINCONT : Marque la fin de la liste des ordres à continuer.

ARRET : En appuyant cette touche on débranche le robot.

TEXTE FOURNI POUR LA MODALITE 2 DE LA 2ème PHASE:

On dispose d'une boîte avec 1000 boules numérotées de 1 à 1000.

On désire connaître la somme des nombres écrits sur 150 boules de la boîte.

On demande à un robot d'effectuer la somme des nombres tracés sur 150 boules qu'il prend au hasard dans la boîte.

Le robot est manipulé par un clavier à touches. On vous donne les noms des touches et leur signification.

Le robot a devant lui, la boîte avec des boules et deux ardoises magiques dans lesquelles il va écrire des nombres. Il ne peut écrire qu'un seul nombre à la fois dans chaque ardoise.

EN VOUS METTANT D'ACCORD, ECRIRE LA LISTE DES TOUCHES QUE VOUS DEVEZ APPUYER POUR QUE LE ROBOT EFFECTUE LA SOMME DES NOMBRES ECRITS SUR LES BOULES.

SCHEMA DU CLAVIER :

MARCHE	PUB	LIS	COMPT			
REP	REC	CONT	TEST N°			
FINREP	FINREC	FINCONT	EC 1			
EC 2	EFF 1	EFF 2	ADD 12			
ARRET						
0	1	2	3	4	5	6
7	8	9				

SIGNIFICATIONS DES TOUCHES :

MARCHE : Si on appuie cette touche le robot est branché.

PUB : Le robot prend une boule.

LIS : Il lit le nombre qui est tracé sur la boule qu'il vient de prendre.

COMPT : Le robot a un compteur qui augmente de 1 chaque fois qu'on appuie sur la touche COMPT.

REP : Il répète indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REP.

REC : Il recommence indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche REC.

CONT : Il continue indéfiniment les ordres qui sont marqués sur les touches que vous appuyez à la suite de la touche CONT.

TEST N° : Si on appuie la touche TEST N° suivi d'un nombre, le robot regarde si la touche COMPT a le même nombre. Si la touche COMPT a le même nombre il arrête l'exécution, sinon il continue.

FINREP : Marque la fin de la liste des ordres à répéter.

FINREC : Marque la fin de la liste des ordres à recommencer.

FINCONT : Marque la fin de la liste des ordres à continuer.

EC 1 : Le robot écrit sur l'ardoise N° 1.

EC 2 : Il écrit sur l'ardoise N° 2.

EFF 1 : Il efface l'ardoise N° 1.

EFF 2 : Il efface l'ardoise N° 2.

ADD 12 : Il ajoute le nombre de l'ardoise N° 1 à celui de l'ardoise N° 2.

ARRET : En appuyant cette touche on débranche le robot

ANNEXE 2

EXPERIENCE DE CONSTRUCTION D'UNE ITERATION EN BASIC (CHAP. IV).

Transcription de la présentation par l'enseignant du problème à résoudre.

B : Le problème qu'on va traiter aujourd'hui c'est le suivant Vous imaginez qu'on a une espèce de caisse ou un carton dans lequel on va mettre des boules, par exemple des boules du Loto. Mais simplement on va s'en servir tout à fait comme dans le loto... Ces boules, comme celles du Loto portent un numéro dessus elles sont numérotées de 1 jusqu'à 1000..... Vous avez compris la situation. Vous avez un grand carton dans lequel vous avez mille boules avec chacune dessus un numéro. Naturellement au niveau de la machine vous ne verrez pas de boules, la seule chose que vous apercevrez c'est le numéro qu'il y a dessus... Vous êtes capable d'imaginer que c'est le nombre d'une boule qui a été tirée.. Alors ensuite ce que l'on va vous demander c'est après un tirage de 100 boules, de dire qu'elle est la somme des nombres qui était sur les 100 boules... Je répète, vous avez une machine.... Vous allez prendre des boules là-dedans.... Vous allez en tirer un certain nombre, on vous demandera : quelle est la somme des nombres qui est inscrite sur les boules tirées ?.... Vous comprenez la question?

On vous a préparé un petit morceau de programme qui va vous aider... Sur la feuille que je vous ai donnée vous avez des numéros... La ligne 0 c'est un petit tiret de petits points qui montre le début du programme et vous avez un petit bout de programme qui va jusqu'à la ligne 8 comprise et dans ce petit bout de programme si vous voulez, vous avez la préparation... On pourrait considérer qu'on est en train de préparer la caisse avec des boules et ensuite vous avez un bout de programme de 999 à 1040. Dans ce petit programme qui est jusqu'à la ligne 1040, vous avez si vous voulez, presque une machine à tirer les boules... C'est à dire quand vous voudrez tirer une boule vous demanderez à la machine de faire ce qu'il y a dans ce petit bout de programme, d'accord ?..... et voilà.... Comment ça va se passer ? Mettons que dans votre programme à un moment donné vous ayez besoin de sortir une boule. Vous allez tirer une boule dans la caisse..... Comme nous n'avons pas 1000 boules dans un carton on le fait faire par la machine Dessous on lui fait faire une simulation, d'accord ? Vous voulez tirer une boule ? que ça soit au début, ou bien en cours du travail, vous taperez sur la machine dans votre programme GOSUB 1000. Ca veut dire à la machine : allez au seul

programme qui est à la ligne 1000 et puis elle va à la ligne 1000 et puis elle revient et à ce moment là elle aura choisi une boule qui s'appellera B. C'est pour ça que j'ai mis ici, pour sortir une boule, vous tapez dans votre programme GOSUB 1000 et à ce moment là vous disposez d'une boule qui s'appelle B qui est la dernière boule que la machine vient de tirer... Vous comprenez comment ça fonctionne ?

Elève: les boules sont tirées au hasard ?

B : oui, elles sont tirées au hasard. C'est à dire que si vous faites un tirage une première fois, la boule suivante sera différente.

Est-ce-que vous avez une question ?

Marie-Hélène tu t'es posée une question à propos de la notion de sous-programme oui ?.... Si tu veux... on peut considérer le petit bout de programme qui est de la ligne 999 à la ligne 1040. Comme un sous-programme qui est mis quelque part dans le programme et qui peut être appelé... C'est à dire, chaque fois qu'on en a besoin il suffit de dire à la machine GOSUB 1000 à ce moment là elle fait le petit bout de programme qui est indiqué et elle revient à l'instruction qui suit à GOSUB 1000.... c'est à dire quand elle a fait le travail elle revient juste à la fin de cette instruction là... Donne, tu peux t'en servir... C'est pas du tout pareil qu'un GOTO parce que quand tu as un go-to dans une ligne, mettons que tu as go-to ici GOTO 100. La machine va à la ligne 100 et puis elle continue à la suite, tandis que quand elle voit GOSUB 1000.... si tu fais GOSUB 1000.... la machine fera ce programme là et elle reviendra à la ligne mettons 20 qui sera la suivante.... Quand elle rencontrera ce qu'il y a à la fin du sub-programme RETOUR... Tu sors à la fin de ton programme la dernière instruction c'est RETOUR (RETURN)... Ca veut dire que lorsqu'elle rencontrera ça, elle retournera jusque là... Donc, pour vous en fait ça veut dire simplement ça Chaque fois que vous voulez une boule vous faites GOSUB 1000 et à ce moment là vous disposez d'une boule qui s'appelle B.... C'est à dire que dans une variable.. dans une boîte qui s'appelle B vous avez le numéro de la boule tirée... Donc, ce n'est pas difficile pour vous de tirer une boule.

Alors ce petit programme vous n'êtes pas obligés de le taper. Debrouillez-vous pour qu'il n'aille pas plus loin que la ligne 999, ce qui normalement ne devra pas trop vous poser de problèmes, d'accord ? Vous écrivez votre programme à partir de la ligne 10 et à partir de la ligne 10 jusqu'à la fin de votre programme (qui n'ira pas si loin) et on vous rentrera que lorsque vous aurez fini ce petit programme qu'on a vu sur les cassettes ,.... d'accord ?

E : Ah ! est-ce qu'on peut rentrer un programme par dessus un autre programme ?

B : Par dessus un autre programme, oui. Bon, voilà cela vous convient ?..... C'est à dire le premier morceau de programme c'est en fait en peu pour mettre à 0 le compteur de tirage et puis de la ligne 1000 à 1040, c'est ce que vous appelez vous quand vous voulez tirer une boule d'accord...? Alors il ne vous reste plus qu'à faire ce que je vous ai demandé. Il s'agit de faire un programme qui permettra de trouver la somme des 100 premières boules tirées, d'accord ? Bon, voilà, je vous laisse faire et passer dans la salle d'à côté.

BREVE DESCRIPTION DES 6 SEANCES ANTERIEURES

1ère séance : - brève description du fonctionnement de la machine
et de ses composantes

Clavier

Ecran

Micro Processeur

Mémoire vive

Mémoire morte

RAM

- instruction

chaîne de caractères

- affichage PRINT

variables

- différences -mode immédiat

-mode programme

- instructions RUN - LIST

2ème séance : - rappel sur différence entre PRINT "X" et PRINT X

- instruction INPUT, variable de lecture

- reconnaissance symbolique et signification

d'erreurs

- résolution de problèmes

3ème séance : - continuation et résolution des problèmes antérieurs.

4ème séance : - graphique

5ème séance - instruction conditionnelle
IF [condition] THEN
- opérateurs de comparaison
- instruction GOTO
- résolution de problèmes

6ème séance : - résolution d'un problème : construction d'une horloge

AUTORISATION DE SOUTENANCE

DOCTORAT 3ème CYCLE, DOCTORAT-INGENIEUR, DOCTORAT USMG

Vu les dispositions de l'arrêté du 16 avril 1974,

Vu les dispositions de l'arrêté du 5 juillet 1984,

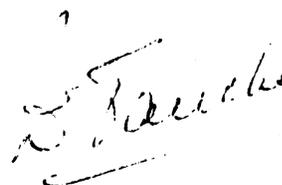
Vu les rapports de M. *me. COLETTE LABORDE*.....

M.

M. *MEJIAS Epouse DAXOUB, Beatriz*..... est autorisé
à présenter une thèse en vue de l'obtention du *Doctorat de 3ème cycle*..
Mathématiques Appliquées.....

Grenoble, le *17 07 84*.....

Le Président de l'Université Scientifique
et Médicale



M. TANCHE

