

PRESENTATION ET ANALYSE D'ACTIVITES DE PROGRAMMATION EN LOGO

Claire DUPUIS
Marie-Agnès EGRET
Dominique GUIN
I.R.E.M. de Strasbourg

AVANT-PROPOS.

L'étude présentée s'intègre dans un ensemble de recherches de l'Institut de Recherche sur l'Enseignement des Mathématiques de Strasbourg liées à l'introduction de l'informatique dans le système scolaire. Elle a pour cadre le GRECO "Didactique et Acquisition des Connaissances Scientifiques" (du Centre National de la Recherche Scientifique) qui coordonne le travail de plusieurs équipes françaises de recherche sur la Didactique des Mathématiques et de l'Informatique ⁽¹⁾.

On distingue habituellement deux domaines de recherche dans l'**informatique éducative** :

- l'informatique "**objet**" considérée comme une discipline.
- l'informatique "**outil**" intervenant dans d'autres disciplines.

Nous nous intéressons ici essentiellement à l'informatique "**objet**", parce que nous pensons que **ces deux domaines sont liés** :

- d'une part, nous faisons l'hypothèse que "l'utilisation de l'informatique dans l'enseignement d'une autre discipline, comme les mathématiques, nécessite une alpha-

(1) On peut citer notamment C. Laborde, N. Balacheff, B. Mejjias (1985), P. Mendelsohn (1985), J. Rogalski (1988) R. Samurçay (1985).

bétisation réelle en informatique" (J. Rogalski, 1985). Certaines opérations de pensée, qui ne sont pourtant pas propres à l'informatique, peuvent avoir dans cette discipline un caractère **nécessaire** ou **explicite** : la structuration des données, l'explicitation des procédures de résolution en sont des exemples. Ce caractère explicite devrait être **producteur d'acquisitions** chez les élèves. Or les travaux des psychologues (R.D. Pea, D.M. Kurland, 1984) mettent en évidence que l'apprentissage de la programmation n'a d'influence sur le développement général de la **connaissance** que s'il est d'assez **haut niveau**.

- d'autre part, le **réinvestissement** dans d'autres domaines des acquisitions résultant d'apprentissages en informatique dépend des activités rencontrées dans cet apprentissage.

Un des résultats essentiels de la psychologie cognitive consiste à avoir mis en évidence le rôle éminent de la **résolution de problème** dans le processus d'acquisition des connaissances. Dans cette perspective, la connaissance **s'acquiert** et se **construit** par résolution de problèmes. C'est dans cet esprit que nous avons conçu nos séquences d'enseignement.

Si notre perspective est centrée sur l'informatique comme "auxiliaire de pensée", nous devons étudier le fonctionnement cognitif dans ce domaine de connaissance. L'étude de l'acquisition des concepts élémentaires propres à l'informatique et l'analyse de leur enseignement présentent donc une importance non seulement du point de vue de la connaissance didactique, mais du point de vue de la **pratique pédagogique**.

Les notions nécessaires à l'alphabétisation informatique présentent des **difficultés conceptuelles** propres dont l'enseignement doit tenir compte : nous avons donc été amenés à construire nos situations autour d'un **apprentissage**. Nous avons choisi le langage LOGO, car il est particulièrement adapté à une approche de l'activité de programmation par la résolution de problèmes grâce aux procédures qui permettent une **programmation modulaire ou structurée**.

Actuellement, il n'existe pas en France d'enseignement de base obligatoire de l'informatique. S'il n'y a pas d'enseignement, il n'y a pas de difficultés spécifiques connues, repérées, pas de conceptions spontanées identifiées :

- Quelles sont les **tâches cognitives** de l'élève dans la résolution de problème : comprendre, construire, compléter une procédure ?

- Quelles sont les **conceptions** et les **acquisitions individuelles** des élèves concernant des notions comme celle de variable et leur évolution au cours de l'enseignement ?

- Quelles sont les **difficultés spécifiques** rencontrées par les élèves dans la planification de l'action, l'acquisition et la mise en oeuvre de concepts et procédures propres à l'informatique ?

- Quels **types d'épreuve** sont nécessaires pour mettre en évidence les acquis des élèves à un moment donné de leur apprentissage ?

Nous essaierons de décrire et d'analyser l'**activité de programmation** des élèves, leurs **difficultés**, de dégager des **types d'erreurs**. Ainsi nous pourrons apporter des éléments de réponse à ces questions et formuler d'autres problèmes plus précis grâce aux résultats obtenus.

INTRODUCTION.

L'activité de programmation contraint les élèves à expliciter, par un programme écrit, leurs procédures de résolution. Cette explicitation reflète les connaissances acquises et les représentations qu'ont les élèves du fonctionnement du dispositif informatique. L'analyse des procédures de résolution implique donc la définition de critères d'analyse fondés, non sur la ressemblance du résultat graphique obtenu avec le résultat demandé, mais sur les connaissances et les représentations sous-jacentes. L'explicitation nécessaire à la programmation pourra aussi être exploitée comme moyen d'enseignement pour remédier aux difficultés apparues, grâce aux résultats tirés de l'analyse.

Nous avons donc poursuivi un double objectif :

1 - **proposer une méthode d'analyse** par la définition de critères qui s'appuient sur une analyse a priori des situations et a posteriori des procédures de résolution. L'explicitation de méthodes d'analyse nous paraît susceptible d'améliorer la communication et la comparaison des recherches expérimentales en didactique de l'informatique. "Plus que les résultats de recherche, ce sont sans aucun doute les méthodes qui peuvent être les plus utiles pour les enseignants, non pas les méthodes de "preuves" des résultats mais plutôt les **méthodes d'analyse des situations** proposées aux élèves, à la fois lorsqu'il s'agit d'introduire des notions et quand il s'agit d'évaluer des acquisitions" (J. Rogalski, 1986).

2 - **mettre en évidence les compétences acquises** par les élèves, les **difficultés** rencontrées et les **procédures mises en oeuvre** par les élèves pour essayer de les contourner.

LE CONTEXTE.

Nous travaillons avec une classe de 19 élèves (13-15 ans) de 4ème puis de 3ème. Les élèves suivent une heure hebdomadaire d'informatique en plus de leur horaire de mathématique normal. Cette heure supplémentaire obligatoire est assurée par leur professeur de mathématique. Cet enseignement n'est pas institutionnel, ce qui nous laisse entièrement libres en ce qui concerne son contenu.

Notre enseignement s'est décomposé en trois phases :

- **première phase** : enseignement de base du langage LOGO graphique et non graphique en insistant sur la **structuration** des programmes. En effet, il est possible d'écrire un programme Logo permettant de réaliser à l'écran un dessin en pilotant la tortue de façon à ce qu'elle suive fidèlement le tracé. Dans ce cas, on utilise seulement les instructions qui constituent le vocabulaire initial du langage Logo. Ce type de programme n'utilise pas, cependant, une des grandes richesses du langage Logo qui est la possibilité de structurer son programme en le décomposant en "procédures" que l'utilisateur définit lui-même : une **procédure** est une suite d'instructions associées par l'utilisateur à un mot qui devient alors le nom de la procédure. Cette décomposition est facilitée par la mise en évidence d'**invariants** (sous-figures identiques). La réalisation du projet d'ensemble implique la coordination des procédures ainsi définies (D. GUIN, J.G. HELM, 1984).

- **deuxième phase** : enseignement de la **récurtivité** c'est-à-dire, en LOGO, la solution à la plupart des problèmes non triviaux nécessitant une itération. Cet enseignement technique a pour but de donner la récurtivité aux élèves comme outil pour la réalisation de projets ultérieurs. Nous avons proposé aux élèves différents modèles afin que chacun d'entre eux puisse se construire une représentation de la récurtivité (C. DUPUIS, M.A. EGRET, D.GUIN, 1985).

- **troisième phase** : **approfondissement**. Il s'agit de proposer aux élèves des situations dans lesquelles la récurtivité peut être utilisée, pour réaliser une évaluation de leurs acquis en matière de récurtivité.

Le travail d'expérimentation comporte des **séquences didactiques** et des **problèmes tests** : les séquences et les tests sont des **situations** qui permettent, d'une part, d'aider les élèves à acquérir les bases de la programmation, d'autre part de mettre en évidence les difficultés des élèves et les conceptions mises en œuvre dans ces situations.

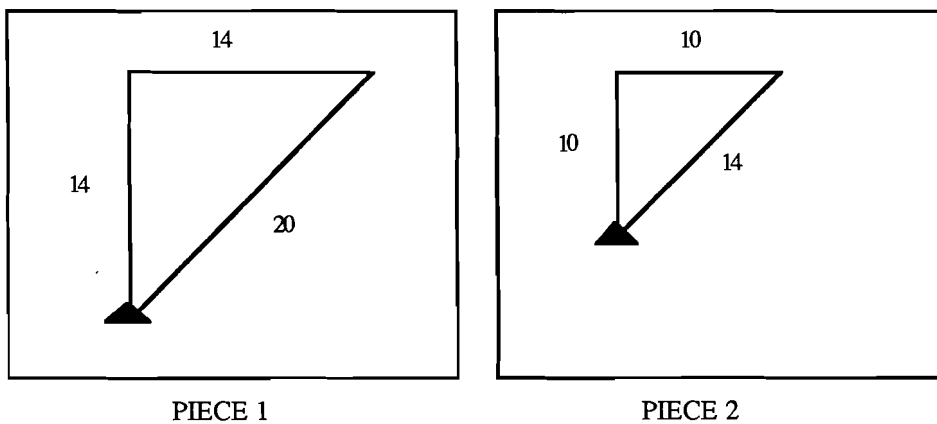
Dans cet article, nous présenterons uniquement une séquence et un test de la première phase ⁽¹⁾.

I - UNE SEQUENCE : TANGRAM.

1) Objectif de la situation et consignes :

Objectif : programmation structurée, mise en évidence d'invariants.

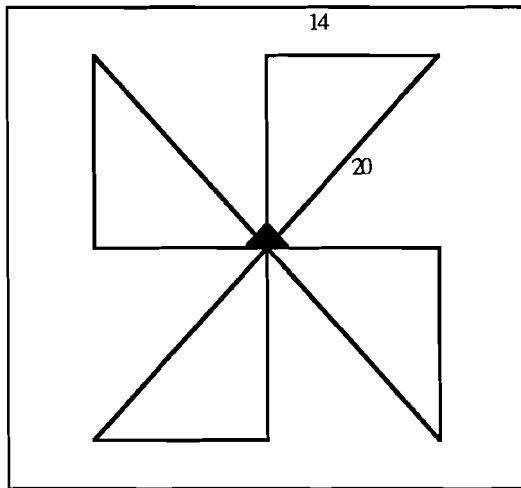
1. Ecrivez les deux programmes permettant à la tortue de tracer les deux sortes de pièces du TANGRAM :



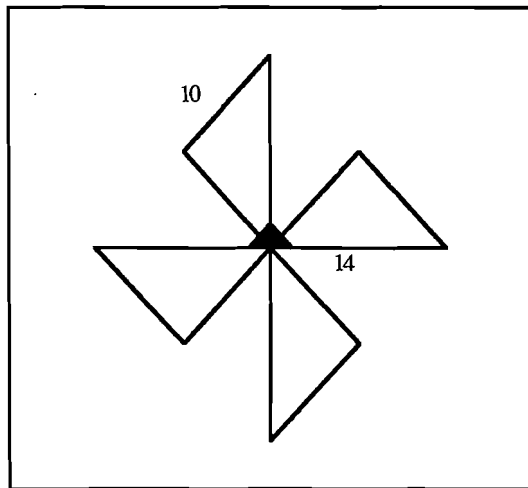
Dans les deux cas, positions initiale et finale de la tortue : origine, cap 0.

(1) La première phase de l'expérimentation est détaillée dans la brochure LOGO (C.Dupuis, M.A. Egret, D. Guin, 1987).

2. Ecrivez les programmes qui permettent d'obtenir les dessins suivants :



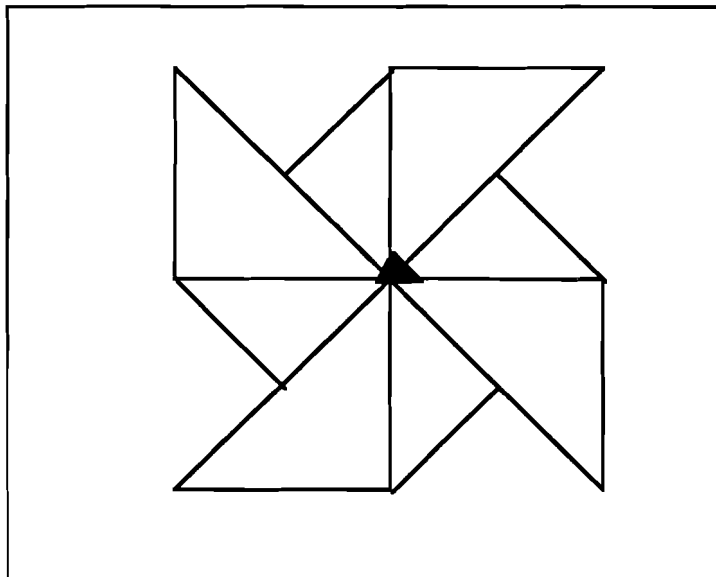
DESSIN1



DESSIN2

positions initiale et finale de la tortue : origine, cap 0.

3. Voici une figure obtenue à l'aide des pièces du TANGRAM ; écrivez le programme :



FIGURE

positions initiale et finale de la tortue : origine, cap 0.

4. Imaginez un dessin utilisant uniquement les pièces du TANGRAM. Ecrivez le programme.

2) Analyse de la situation :

Nous recherchions des situations où la structuration du programme soit ressentie non comme une contrainte supplémentaire imposée par l'enseignant, mais comme une stratégie économique de résolution de problème. Le jeu du TANGRAM est par nature "structuré". Les élèves ont pu manipuler les pièces et comprendre l'idée de structuration : "je pose un TRIANGLE1, puis un TRIANGLE2, etc.". En effet, les huit pièces élémentaires étant données (quatre petits et quatre grands triangles), les élèves n'ont plus qu'à les assembler pour former la figure : c'est un choix didactique d'éliminer la difficulté liée au choix d'invariants. De même, la consigne qui précise les positions initiale et finale de la tortue a pour but d'enlever certaines des difficultés liées au choix de coordination entre les différentes procédures.

3) Commentaires :

a) corrigé.

Le corrigé proposé aux élèves mettait en évidence les **invariants** PIECE1 et PIECE2 et les **interfaces**⁽¹⁾ POSPIECE1, POSPIECE2, POSDEPART et POSRETOUR

POUR PIECE1

AV 14 TD 90
AV 14 TD 135
AV 20 TD 135
FIN

POUR PIECE2

AV 10 TD 90
AV 10 TD 135
AV 14 TD 135
FIN

POUR DESSIN1

REPETE 4 [PIECE1 POSPIECE1]
FIN

POUR POSPIECE1

TD 90
FIN

POUR DESSIN2

POSDEPART
REPETE 4 [PIECE2 POSPIECE2]
POSRETOUR
FIN

POUR POSDEPART

TG 45
FIN

(1) Nous appelons interface une procédure (ou suite d'instructions) intermédiaire définie par l'utilisateur pour réaliser la coordination entre deux procédures.

POUR POSPIECE2

TG 90

FIN

POUR POSRETOUR

TD 45

FIN

POUR FIGURE

DESSIN1 DESSIN2

FIN

b) le pilotage de la tortue.

Les erreurs de pilotage de la tortue peuvent être qualifiées d'erreurs de jeunesse. En effet, ces erreurs ont disparu progressivement au cours de l'apprentissage. Celles qui persistent le plus longtemps sont des erreurs d'orientation relative.

c) la structuration et la décomposition en procédures.

La programmation structurée avec procédures représente une réelle difficulté pour les programmeurs débutants. Pour certains, malgré les consignes proposées, le coût d'apprentissage est tel que c'est seulement quand il ne leur sera plus possible de faire autrement qu'ils se résoudront à décomposer en procédures et à structurer.

d) la prise en compte de l'état de la tortue dans la coordination des procédures graphiques.

Nous distinguons 2 types d'erreurs :

- Les erreurs **simples d'interface**, comme des erreurs de latéralisation (par exemple, TG 45 au lieu de TD 45) ou des erreurs sur la valeur d'un angle (par exemple, TD 30 au lieu de TD 45).

- Les erreurs de **conception** comme celles qui résultent d'une conception **image** de la procédure, c'est-à-dire une identification de la procédure avec le tracé souhaité ou la figure obtenue. Cette conception est un obstacle à la programmation en LOGO. Il est difficile pour les élèves de comprendre la nécessité de préciser les positions initiale et finale de la tortue puisque le dessin obtenu est identique au modèle (cf ci-dessous II 3 b. 1).

Par exemple, pour DESSIN2, l'une des difficultés était de distinguer les changements de position à répéter (POSPIECE2) des changements de positions initiale

et finale de la tortue, puisque PIECE2 n'avait pas la même orientation que dans la première question.

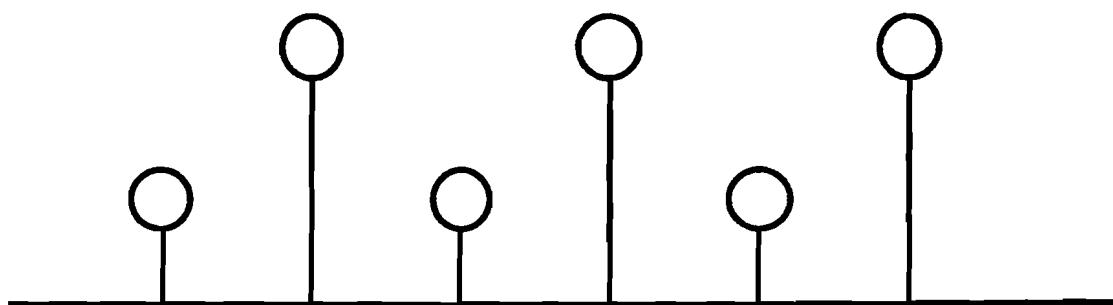
e) appréhension opératoire de la figure.

Dans les projets, il faut que l'élève se rende compte que la tâche "dessin avec crayon à papier" n'est pas la même que la tâche "programmation avec la tortue LOGO". Une appréhension opératoire de la figure lui est nécessaire pour structurer ses programmes : nous appelons appréhension opératoire des figures la capacité d'opérer une réorganisation perceptive adaptée à la programmation LOGO (R. DUVAL, 1988 ; C. DUPUIS, 1988).

II - UN TEST : BALLONS.

1) Présentation du test.

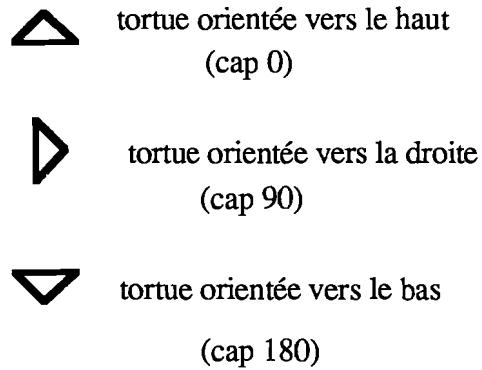
Nous vous présentons un test individuel, baptisé BALLONS, réalisé au bout de seize heures de pratique active de la programmation. Ce test avait déjà été proposé à des élèves dans des conditions d'expérimentation différentes (J. Hillel, R. Samurçay, 1985). Nous voulions comparer les résultats pour voir l'impact de notre enseignement de la programmation structurée. Les élèves ont pour consigne d'écrire le programme permettant de réaliser à l'écran un dessin semblable à celui-ci :



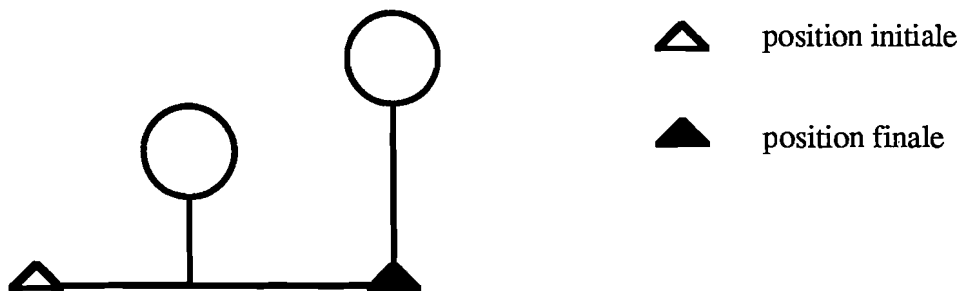
ATTENTION aux remarques suivantes :

- les distances entre les pieds des tiges sont égales.
- les cercles ont tous le même rayon.
- il y a deux longueurs de tiges.

A titre de corrigé, voici un ensemble de procédures réalisant le projet BALLONS. Pour illustrer notre propos, nous représenterons les procédures par leur exécution. Plus exactement, il s'agit d'un schéma d'exécution puisque nous faisons figurer les positions initiale et finale de la tortue alors qu'elles ne peuvent apparaître simultanément à l'écran. Dans les schémas d'exécution, nous représentons la tortue par un triangle permettant de visualiser sa position et son orientation :

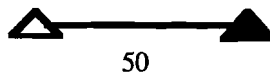



Dans cet exemple, le projet d'ensemble BALLONS est décomposé de la manière suivante




La procédure DESSIN, correspondant au schéma ci-dessus, est répétée trois fois et suivie de TRAIT :

POUR TRAIT
TD 90
AV 50
TG 90
FIN



 : position initiale

 : position finale

POUR BALLONS
REPETE 3 [DESSIN]
TRAIT
FIN

```

POUR DESSIN
INTERFACE
BALLON 10
INTERFACE
BALLON 30
FIN

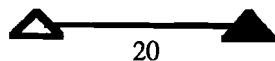
```

Le DESSIN lui-même est décomposé en deux procédures INTERFACE identiques, réalisant les déplacements horizontaux de la tortue et deux procédures BALLON :L dans lesquelles la longueur de la tige ne sera pas la même.

```

POUR INTERFACE
TD 90
AV 20
TG 90
FIN

```



 : position initiale

 : position finale

```

POUR BALLON : L
AV : L
TD 90
POLY 2 36
TG 90
RE : L
FIN

```



exemple : BALLON10

L = 10

finale confondues

La procédure POLY, qui permet d'obtenir un cercle, a un statut particulier car elle a été écrite au préalable, dans un autre contexte.

```

POUR POLY : COTE : NB
REPETE : NB [AV : COTE TG 360 /: NB]
FIN

```

Dans le test BALLONS, la figure est présentée sans indication de positions initiale ou finale de la tortue, ni de décomposition. Aucune consigne supplémentaire n'est donnée. Le choix didactique est ici différent de celui du Tangram : la **décomposition** et la **coordination** sont dans ce cas à la charge de l'élève. Nous voulons observer les stratégies de programmation des élèves et savoir, parmi les outils de programmation mis à leur disposition dans l'enseignement, ceux qu'ils choisiront d'utiliser.

2) Des critères d'analyse : pourquoi ?

1. Qu'est-ce qu'un critère d'analyse ?

L'idée de critère d'analyse, dans une recherche expérimentale en didactique, est

proche de celle de variable qualitative avec le sens que lui donnent les statisticiens en analyse des données. Une variable qualitative à modalités est une application de l'ensemble des individus dans l'ensemble des modalités. Dans notre recherche, un individu est un programme écrit par un élève. Ce pourrait être aussi un ensemble de programmes écrits par le même élève. La définition de chaque variable-critère et de ses modalités induit une partition de l'ensemble des individus et permet donc d'explicitier, au-delà des différences apparentes ou superficielles, ce que certains programmes ont en commun.

2. Au-delà de la réussite ...

Il n'est guère raisonnable de se contenter d'observer seulement la réussite ou l'échec, sauf sur une tâche ponctuelle. D'une part, la réussite peut, en programmation comme en mathématique, être le résultat de stratégies fort différentes. D'autre part, l'échec peut avoir des causes diverses que les critères d'analyse doivent distinguer à partir des indicateurs que sont les productions des élèves.

"Les critères d'analyse ne sont pas fondés sur la proximité de la réponse avec le réalisme graphique de la réponse correcte, mais sur la nature des processus psychologiques en jeu inférée à partir des productions" (P. Mendelsohn, 1985).

3. Une analyse multi-critères.

La complexité de la tâche de programmation nécessite la **définition de plusieurs critères d'analyse**. Chaque critère qui intervient dans l'analyse reflète un des aspects de l'activité. Il faut veiller à l'indépendance a priori des critères c'est-à-dire à la possibilité d'associer n'importe quelle modalité d'un critère donné avec n'importe quelle modalité d'un autre critère pour constituer le profil d'un individu.

3) Définition de critères d'analyse.

Nous aborderons successivement les stratégies de décomposition, le traitement de la coordination, l'utilisation des procédures POLYGONE pour tracer les cercles, le traitement de la répétition, la décomposition en sous-procédures et l'utilisation de variables.

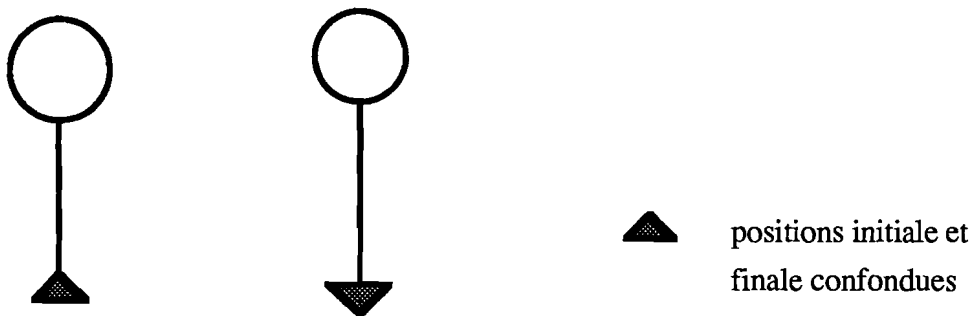
a) Stratégies de décomposition : le critère ETAT.

L'état de la tortue se caractérise par sa position et son orientation. Nous distinguerons ici trois types de prise en compte de l'état de la tortue à la fin d'une procédure graphique (ou d'une liste d'instructions) destinée à être intégrée à un projet. Ces trois types constituent les trois modalités du critère ETAT.

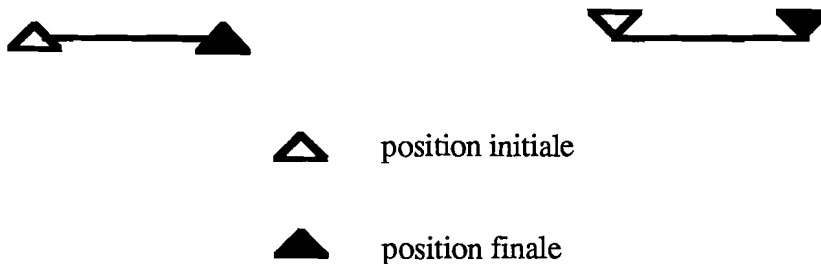
- Première modalité du critère ETAT : position standard "PS".

A la fin de la procédure, il y a un retour à une position "standard". Une position "**standard**" est une position à partir de laquelle la coordination de cette procédure avec les autres est économique (Il s'agit souvent de la position initiale). Dans ce cas, il est évident que l'élève a pris en compte l'état de la tortue à la fin de la procédure ; la coordination nécessite l'utilisation d'interfaces hors de la procédure.

Voici deux exemples de procédures de cette modalité dans la situation du test BALLONS.



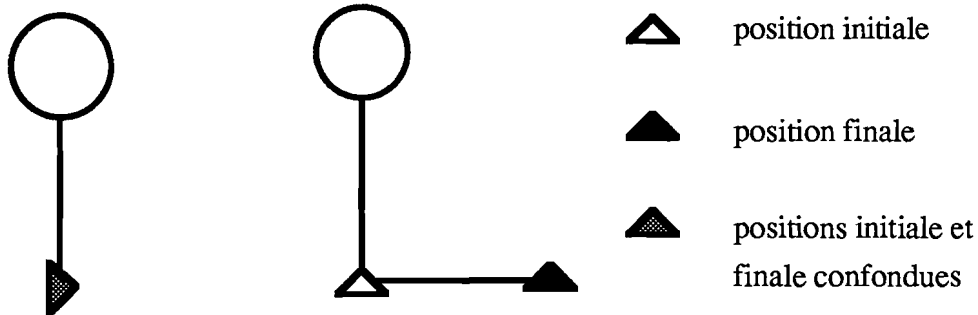
Pour réaliser le projet complet, l'élève doit dans ce cas écrire une interface. Les interfaces correspondant respectivement aux deux procédures visualisées ci-dessus peuvent être représentées ainsi :



L'utilisation de positions "standard" confère à la procédure un caractère de généralité, d'indépendance du contexte et facilite une éventuelle réutilisation dans d'autres projets.

- Deuxième modalité du critère ETAT : interface intégrée "INT".

L'interface est intégrée à la procédure. Voici deux exemples de procédures de cette modalité dans la situation du test BALLONS.

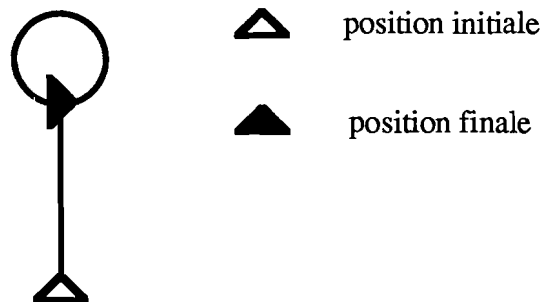


Dans ce cas aussi, il est évident que l'élève a pris en compte l'état de la tortue à la fin de la procédure. La réutilisation d'une telle procédure dans un autre contexte peut être difficile. La distinction faite ici entre deux stratégies ne signifie pas pour nous que l'une est la "bonne" et l'autre la "mauvaise" : suivant la situation, l'une ou l'autre peut être la plus économique pour résoudre un problème précis (cf. BALLONS). Mais dans une perspective de travail à long terme et d'utilisation de procédures hors du contexte où elles ont été écrites, seul le retour systématique à une position "standard" est raisonnable.

- Troisième modalité du critère ETAT : aucun retour "SANS".

Aucun retour à une position "standard" ne figure dans la procédure. Aucun mouvement de la tortue n'est effectué en plus de ceux qui sont nécessaires au tracé.

Voici un exemple de procédures de cette modalité dans la situation du test BALLONS.



b) Le traitement de la coordination.

La coordination des procédures pour obtenir le programme cherché est un problème qui se pose dès qu'il y a décomposition en procédures. La difficulté de la coordination se manifeste ici à deux reprises et de manières fort différentes : d'une part,

dans la coordination des procédures résultant de la décomposition que chaque élève a choisie dans ce contexte ; d'autre part, dans la réutilisation, pour tracer les cercles, d'une procédure écrite dans un autre contexte. Cette différence fondamentale de contexte nous a conduit à définir deux critères différents :

le critère COORDINATION, qui analyse le traitement de la coordination entre des procédures de décomposition conçues et écrites dans le contexte où elles sont coordonnées ;

le critère COORDINATION HORS CONTEXTE, qui analyse le traitement de la coordination lors de la réutilisation d'une procédure conçue et écrite dans un autre contexte. Il s'agit ici de la procédure utilisée pour tracer les cercles.

b.1 Le critère COORDINATION.

Les quatre modalités du critère coordination sont :

"0" : la coordination des procédures n'est pas réalisée.

"C" : le programme est correct.

Nous distinguerons deux types d'erreurs dans la coordination des procédures :

"ES" : le programme présente des erreurs simples d'interface, comme des erreurs de latéralisation, des erreurs sur la valeur d'un angle ou sur les relations entre des longueurs.

"EC" : le programme présente des erreurs de conception, comme celles qui résultent d'une conception "image" de la procédure.

L'élève considère la procédure comme un moyen d'obtenir un résultat et non comme une suite d'instructions modifiant (éventuellement) l'état de la tortue. Les premiers programmes des élèves reflètent tous cette conception qui est à rapprocher de leur propre expérience du dessin à la main : pour tracer un trait, il est inutile de savoir où et dans quelle position s'est arrêté le crayon après le tracé précédent. Cette conception ne peut être observée qu'au moment où l'élève veut coordonner les procédures qu'il a choisies d'écrire. Deux situations sont caractéristiques de cette conception :

- l'élève n'écrit aucune instruction pour coordonner ses procédures alors qu'il n'a pas intégré l'interface aux procédures.

- l'élève écrit une interface supposant un retour à une position "standard" alors qu'il n'a pas fait de retour à une position "standard".

En principe, le problème de la coordination des procédures ne se pose pas si l'élève écrit un programme sans aucune procédure ou sans instruction telle que REPETE, qui résulte d'une structuration ; mais cette attitude ne s'observe qu'au début de l'apprentissage de la programmation ou pour des programmes très simples. De sorte que, pour ne pas multiplier les codages, nous utiliserons le critère coordination, par extension dans ce cas rare.

b.2 Le critère COORDINATION HORS CONTEXTE (CHC).

Pour tracer des cercles (approximation par des polygones réguliers), les élèves disposent de deux procédures écrites dans leurs cahiers :

POLYGONE1 : longueur du côté : nombre de côtés et
POLYGONE2 : longueur du côté : angle.

En effet, nous avons travaillé, lors d'une séance précédente, à l'écriture de procédures permettant de tracer un polygone régulier quelconque, puis un cercle. Tous les élèves ont utilisé l'une ou l'autre des procédures POLYGONE pour tracer les cercles dans la situation BALLONS.

Mais POLYGONE1 et POLYGONE2 sont deux procédures qui ont été conçues, écrites, et exécutées pour répondre à une tâche particulière : le dessin d'un polygone, puis d'un cercle isolé. Les élèves n'avaient pas eu l'occasion de les intégrer dans d'autres projets. Il semble bien que, dans ce cas, la conception "image" de la procédure, c'est-à-dire l'identification du cercle avec la procédure POLYGONE, domine chez tous les élèves. Ceci se repère aux faits que :

- Dans la première écriture de leurs programmes BALLON, aucun élève n'a écrit d'interface entre les procédures de la tige et du cercle. Leurs cercles étaient donc tous tangents à la tige, puisque la position initiale et finale de la tortue est tangente au cercle dans les deux procédures POLYGONE1 et POLYGONE2.



- Au moment de l'exécution, la réaction générale a été la surprise. Les élèves ne s'étaient pas posés la question de la coordination à cet endroit alors que tous se préoccupaient de coordination entre leurs procédures BALLON.

Ces problèmes de prise en compte de l'état de la tortue à la fin de la procédure et de coordination ne sont donc pas exactement de même nature que ceux qui se posent entre deux procédures qui ont été conçues et écrites pour être coordonnées. La coordination d'une procédure écrite dans un autre contexte peut nécessiter un retour en arrière sur la procédure et un enrichissement de la conception de cette procédure. C'est pourquoi le problème des cercles est traité à part. Tous les autres critères sont observés en dehors de la procédure éventuellement utilisée pour tracer les cercles. Les modalités du critère CHC figurent dans l'annexe 1.

c) Un critère pour la répétition.

L'élève peut choisir d'utiliser ou non l'instruction REPETE. S'il l'utilise, il doit déterminer la séquence à répéter et le nombre de répétitions. Mais là encore il a un choix car le traitement de la répétition peut se faire de la manière suivante :

liste d'instructions puis REPETE [la même liste d'instructions].

Ce type d'écriture de la répétition, que nous avons observé au début de l'apprentissage, est encore présent dans ce test : il persiste donc assez longtemps. Il correspond au modèle spontané que les élèves ont de la répétition : **faire une action et recommencer.**

Dans un contexte un peu différent, C. Laborde, N. Balacheff, et B. Mejias (1985) ont aussi observé cette forme de répétition qu'ils rapprochent 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".

Les modalités du critère REPETE figurent dans l'annexe 1.

d) Deux critères d'utilisation.

SP : Utilisation d'au moins une (sous-) procédure.

"OUI"

"NON"

Ce critère ne recoupe pas les autres, puisque l'élève peut choisir d'écrire tout son programme dans une seule procédure, quitte à écrire plusieurs fois une même suite d'instructions. L'absence de (sous-) procédure est rarement associée à la réussite d'ensemble d'un projet complexe.

VAR : Utilisation d'au moins une variable pour désigner une longueur.

"OUI"

"NON"

Si une variable est utilisée pour désigner une longueur, par exemple la longueur de la "tige", il est possible de réutiliser la même liste d'instructions pour les deux sortes de ballons.

4. Mise en oeuvre de ces critères.

Les résultats de l'analyse suivant ces critères sont présentés dans le tableau de l'annexe 2.

a) Pertinence des modalités des critères.

Certaines modalités des critères, possibles a priori dans cette situation, n'ont pas été observées. La modalité SANS retour à une position standard du critère ETAT n'est pas observée, tous les élèves ayant pris en compte la position finale de la tortue suivant l'une ou l'autre des stratégies position standard ou interface intégrée. Il n'y a pas non plus d'erreurs de conception dans la coordination des procédures. Ce phénomène peut être considéré comme un résultat positif de l'enseignement, car l'absence de retour à une position standard et la conception "image" de la procédure ont été observées chez ces mêmes élèves au cours de l'apprentissage.

Deux modalités du critère REPETE a priori possibles (XC et FC) sont absentes mais si l'on considère séparément les deux codes, toutes les modalités sont présentes. L'élève qui n'a pas utilisé l'instruction REPETE, tout en ayant un programme correct, a décomposé en procédures et préfère écrire trois fois les noms de ses procédures.

De fait, la situation n'était pas suffisamment complexe pour qu'il soit absolument nécessaire, pour réussir, d'utiliser des variables, des procédures ou même l'instruction REPETE. Pour les variables, il est clair que leur utilisation, qui reste difficile pour les élèves, n'était pas significativement plus économique que l'écriture de procédures sans paramètre.

b) Profils des individus.

Parmi les huit élèves pour lesquels la coordination des procédures est correcte, l'instruction REPETE est bien utilisée et les cercles sont bien placés, on trouve toutes les combinaisons possibles des critères VARIable et ETAT : certains ont utilisé des variables, d'autres n'en ont pas utilisées ; les uns ont écrit des procédures avec retour à une position "standard", les autres avec interface intégrée. Mais tous ces élèves ont utilisé au moins une sous-procédure.

Si l'on ne tient pas compte de la position du cercle par rapport à la tige, quatre élèves de plus ont écrit un programme "correct". Mais parmi eux, une élève n'a pas utilisé de sous-procédure et un autre n'a pas utilisé l'instruction REPETE.

Si nous croisons les critères SP : utilisation d'au moins une (sous-) procédure et coordination restreint aux modalités correcte ou non-correcte, nous obtenons :

		Coordination correcte	
		Oui	Non
SP	Oui	11	4
	Non	1	3

La **décomposition** du problème en procédures est liée à la réussite d'ensemble puisque sur douze élèves qui ont réussi la coordination, onze d'entre eux ont utilisé des procédures. Cependant la liaison n'est pas statistiquement significative au seuil habituel de 5 % puisque la probabilité, à marges constantes, d'obtenir un élève (ou zéro) ayant réussi sans utiliser de (sous-) procédure est 0,117 soit près de 12 %.

CONCLUSION.

A l'issue de cette première phase, nous pouvons affirmer que les élèves ont acquis une certaine maîtrise du pilotage de la tortue et que les éléments de base d'une programmation graphique structurée sont en place. La structuration des programmes est devenue une méthode efficace de résolution de problèmes.

Nous avons pu pendant cette phase analyser l'activité de programmation des élèves. Pour cela, nous avons :

- dégagé des **types d'erreurs**,
- défini des **critères d'analyse** en vue d'une **évaluation** contrôlée,

- mis en évidence les compétences acquises,
- commencé à **expliquer** d'une part les procédures mises en oeuvre par les élèves pour contourner les difficultés et d'autre part les **représentations** qu'ils ont du fonctionnement du système informatique ⁽¹⁾.

Plus généralement, nous pensons que ce travail d'analyse est indispensable dans tout enseignement d'informatique pour :

1) Construire des séquences d'enseignement de manière à ne faire apparaître que très progressivement les **difficultés** mises en évidence dans l'analyse.

2) Evaluer les acquis des élèves grâce à des tests.

et les mathématiques ?

Durant cette première phase, nous avons repéré des difficultés liées à l'utilisation des **variables**, difficultés qui ne sont pas propres à la programmation LOGO mais qui sont observées aussi en mathématiques au même niveau scolaire dans les problèmes de mise en équation ⁽²⁾.

Un enseignement sur les fonctions en troisième, utilisant la programmation LOGO, a été expérimenté avec les élèves ayant participé au travail décrit ici. Cet enseignement a été reconduit cette année avec des élèves n'ayant reçu qu'une quinzaine d'heures d'enseignement informatique. Il nous a révélé l'impossibilité d'utiliser des activités de programmation dans l'enseignement des mathématiques avec des élèves n'ayant pas atteint un niveau suffisant de programmation ⁽³⁾.

Par conséquent, c'est seulement après une réelle alphabétisation en informatique que l'on peut aborder le problème de l'aide apportée par les activités de programmation dans l'enseignement des mathématiques.

(1) Ce travail est développé dans (C. Dupuis, 1988 ; C. Dupuis, D. Guin, à paraître).

(2) C. Dupuis, D. Guin, 1988 ; C. Dupuis, 1988 ; C. Dupuis, D. Guin, à paraître.

(3) D. Guib, 1988.

ANNEXE 1 : Modalités des critères CHC et REPETE.

Coordination Hors Contexte :

"0" - Aucun cercle n'est tracé .

"C" - Les cercles sont correctement placés aux bouts des tiges : l'interface est correcte.

"T" - Les cercles sont tangents aux tiges : l'interface est absente.

"A" - Les cercles ne sont ni tangents, ni corrects : l'interface est incorrecte.

REPETE un nombre de fois [une liste d'instructions] :

Nous allons définir un critère codé sur deux colonnes, le premier code indique si le nombre de répétitions est exact, le second si la liste d'instructions répétée est correcte ou non.

premier code :

"0" : aucune instruction REPETE n'est utilisée .

"C" : l'instruction REPETE est utilisée et le nombre de répétitions est correct.

"F" : l'instruction REPETE est utilisée et le nombre de répétitions est faux.

"X" : l'instruction REPETE est utilisée mais avec une structure du type : [liste d'instructions] puis REPETE [la même liste d'instructions].

deuxième code :

"0" : pas de liste.

"C" : la liste d'instructions répétée est correcte .

"F" : la liste d'instructions répétée est fautive .

Ceci fait que l'on peut obtenir les codes **00, CC, CF, FC, FF, XC et XF** pour ce critère.

ANNEXE 2

Les réponses analysées ici sont les programmes modifiés par les élèves après une exécution.

CRITERES :	CHC	SP	VAR	ETAT	COordination	REPète
ELEVES :						
STé + DAV + KLA	C	OUI	NON	PS	C	CC
SEB	C	OUI	OUI	PS	C	CC
ALI + CAR + NAT	C	OUI	NON	INT	C	CC
MAR	C	OUI	OUI	INT	C	CC
SAR	T	OUI	NON	INT	C	CC
SAB	A	OUI	NON	PS	C	CC
CYR	C	OUI	OUI	PS	C	00
ALE	T	NON	NON	INT	C	CC
FLO	T	NON	NON	INT	ES	CF
GIL	T	NON	NON	INT	ES	XF
KAR	C	OUI	OUI	INT	ES	CF
RAC	T	OUI	NON	PS	ES	CF
REG	T	OUI	OUI	PS	ES	FF
CHR	T	OUI	NON	INT	0	00
MAG	0	NON	NON	INT	ES	CF

Dans ce tableau, nous avons regroupé dans une même ligne les élèves qui correspondaient aux mêmes modalités des critères ; cela ne signifie pas que leurs programmes étaient absolument identiques. Lorsque l'élève ne définit pas formellement de procédure mais utilise un REPETE [une liste d'instructions], le critère ETAT est observé sur la liste d'instructions.

BIBLIOGRAPHIE.

DUPUIS C., (1988) : Appréhension des figures et choix de codage en programmation Logo, *Séminaire de Didactique des Mathématiques et de l'Informatique*, IMAG, Grenoble.

DUPUIS C., EGRET M.A., GUIN D. (1985) : Récursivité et Logo 1 : Préexpérimentation, *Brochure I.R.E.M de Strasbourg*.

DUPUIS C., EGRET M.A., GUIN D. (1987) : Logo 3. Programmation structurée : Présentation et Analyse de situations, *Brochure I.R.E.M. de Strasbourg*.

DUPUIS C., EGRET M.A., GUIN D. (1988) : Pour une analyse multi-critères de programmation en Logo, pp 111-130, *Annales de Didactique et de Sciences Cognitives*, vol .1, Strasbourg.

DUPUIS C., GUIN D. (1988) : "Découverte de la récursivité en LOGO dans une classe", *Actes du premier colloque franco-allemand de Didactique des Mathématiques et de l'Informatique*, La Pensée Sauvage, F-38002 Grenoble.

DUPUIS C., GUIN D. (à paraître) : Gestion des relations entre variables dans un environnement de programmation LOGO.

DUVAL R. (1988) : " Pour une approche cognitive des problèmes de géométrie en termes de congruence", *Annales de Didactique et de Sciences Cognitives* , N° 1, pp 57 - 74.

GUIN D. (1988) : LOGO, to find out the different registers of the function notion, *ICME 6*, Budapest, July 27 - August 3 .

GUIN D., HELM J.G. (1984) : LOGO - 2 : Rapport d'expérimentation en CM2, *Brochure I.R.E.M de Strasbourg*.

HILLEL J., SAMURCAY R. (1985) : Analysis of a LOGO environment for learning the concept of procedures with variables, *Research supported by Quebec Ministry of Education*, FCAC Grant EQ 2539

LABORDE C., BALACHEFF N., MEJIAS B. (1985) : Genèse du concept d'itération : une approche expérimentale, *Enfance*, Vol. 2/3, pp. 223 - 239.

MENDELSON P. (1985) : L'analyse psychologique des activités de programmation chez l'enfant de CM 1 et CM 2, *Enfance*, Vol. 2 / 3 , pp. 213 - 221.

PEA R.D., KURLAND D.M. (1984) : On the cognitive effects of learning computer programming, *New Ideas Psychol.*, Vol. 2, N° 2, pp. 137-168.

ROGALSKI J. (1985) : Alphabétisation informatique, problèmes conceptuels et didactiques, *Bulletin de l'APMEP*, N° 347, pp. 61-74.

ROGALSKI J. (1986) : Pour une pédagogie de l'informatique, *Enseignement public et informatique*, N° 42, pp. 105 - 109.

ROGALSKI J. (1988) : "Acquisition de structures conditionnelles : effet des prérequis logiques et des représentations du dispositif informatique", *Annales de Didactique et de Sciences Cognitives*, N° 1, pp 131 - 152.

SAMURCAY R. (1985) : Signification et fonctionnement du concept de variable informatique chez des élèves débutants, *Educational Studies in Mathematics*, 16, pp. 143 - 161.