

**UNIVERSITE LOUIS PASTEUR  
IREM**

**Institut  
de Recherches  
sur l' Enseignement  
des Mathématiques  
10 rue du Général Zimmer  
67084 STRASBOURG CEDEX**

## **LOGO**

**3 . Programmation Structurée :  
Présentation et Analyse de situations .**

**Claire DUPUIS  
Marie-Agnès EGRET  
Dominique GUIN**

**Ce travail a été mené dans le cadre du GRECO Didactique et acquisition des  
connaissances scientifiques , du Centre National de la Recherche Scientifique .**

**Siège administratif : 54 Bld . Raspail 75270 PARIS CEDEX 06**

**1987**

## Avant - propos

Cette brochure contient la description et l'analyse d'une expérimentation menée avec des élèves de quatrième sur un enseignement de base du langage LOGO orienté vers l'apprentissage de la programmation structurée . Il a été suivi par une introduction de la récursivité ( C. Dupuis , M.A. Egret , D. Guin , à paraître ) .

Les enseignants pourront trouver , s'ils le souhaitent , des thèmes d'activités dans la partie " Présentation des différentes séances " ( p. 6 à 32 ) avec leurs objectifs et nos commentaires . La partie " Présentation des différents tests " ( p. 33 à 68 ) permet d'estimer les acquis et les difficultés des élèves au cours de cet apprentissage .

## Introduction

Cette étude s'intègre dans un ensemble de recherches de l'IREM de Strasbourg liées à l'introduction de l'informatique dans le système scolaire .

L'informatique éducative comprend deux domaines de recherche :

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

Nous nous intéressons ici essentiellement à la première approche : les éléments de notre étude sont tournés vers l'informatique " **objet** " , parce que nous pensons que **ces deux domaines sont liés** :

### Notre hypothèse :

*Nous faisons l'hypothèse que l'utilisation de l'informatique dans l'enseignement d'une autre discipline , comme les mathématiques , nécessite une alphabétisation réelle en informatique ( J. Rogalski , 1985 ) .*

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 .

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** .

Dans ce document , nous présentons un enseignement de base orienté vers l'apprentissage de la programmation structurée qui a été suivi par une introduction de la récursivité d'autant plus indispensable en LOGO qu'il n'existe pas d' itération conditionnelle . Elle fera l'objet d'un compte-rendu dans la brochure suivante ( C. Dupuis , M.A. Egret , D. Guin , à paraître ) .

Le **changement de l'enseignement des mathématiques** , et de son **contenu** , à partir d'une **intervention de l'informatique** , nous semble nécessiter une étude sur les concepts **informatiques** de variable , de récursivité , de procédure **en relation** avec les concepts **mathématiques** pour avoir une certaine maîtrise sur les **représentations** des objets mathématiques que les élèves se construiront . 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** .

Le travail de préexpérimentation ( C. Dupuis , M.A. Egret , D. Guin , 1985 ) a été conduit sur des **séquences didactiques** et des **problèmes tests** : le but était de repérer des **situations favorables** au recueil d'**observables significatifs** pour l'étude des représentations et des acquisitions des élèves . Il s'agit de construire **des situations** nous permettant de mettre en évidence des **critères d'analyse** des productions : **analyse des coordinations et de la planification de l'action** dans la tâche particulière qu'est la programmation . Ces situations sont des conséquences de ce **choix didactique** .

### Nos objectifs :

A) **La construction de situations** destinées à produire chez l'élève l'**acquisition des connaissances significatives** dans le domaine de **la programmation structurée graphique** .

B) **L'analyse des procédures de résolution** , des difficultés des élèves et des conceptions mises en oeuvre dans ces situations .

Ces deux points sont indissociables : il est , en effet , inconcevable de construire sans analyser ou d'analyser sans construire .

La spécificité des **contenus** de connaissance et le caractère propre des **situations d'enseignement** , en particulier en classe , constituent sans doute les traits particuliers de l'approche **didactique** . Il s'agit d'une rupture importante par rapport à certaines approches **cognitivistes** qui n'accordent pas assez d'importance aux **contenus** de connaissance et au groupe **classe** . L'analyse des **apprentissages** en situation scolaire nécessite l'étude des **interactions** du système formé par les élèves , le maître et la connaissance .

Les notions nécessaires à **l'alphabétisation informatique** présentent des **difficultés conceptuelles** propres dont l'enseignement doit tenir compte : nous avons donc été amenées 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 problème grâce aux procédures qui permettent une **programmation modulaire ou structurée** .



Actuellement , il n'existe pas au collège d'enseignement de base 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 .

Cette recherche a été menée dans le cadre du GRECO " Didactique et Acquisition des Connaissances Scientifiques " du CNRS qui coordonne le travail de plusieurs équipes de recherche sur la Didactique des Mathématiques et de l'Informatique ( N. BALACHEFF , C.LABORDE , P. MENDELSON, J. ROGALSKI , A. ROUCHIER , R. SAMURCAY) . Une présentation des travaux du groupe Didactique de l'informatique a été faite au colloque du GRECO ( A. Rouchier , R. Samurçay, 1987 ) .

## Conditions de l'expérimentation

L'expérimentation est menée avec une classe de 4<sup>ème</sup> composée de 19 élèves de 14 ans , avec la participation de l'enseignante de mathématiques .

Les élèves provenant de deux classes de 5<sup>ème</sup> différentes ont tous " fait du LOGO " graphique , mais le nombre d'heures passées devant un micro-ordinateur varie beaucoup d'une classe à l'autre . Seuls , quatre élèves ont des éléments de programmation en BASIC :

- L'une d'entre eux ( MAG ) a un micro-ordinateur chez elle .
- Une autre ( CHR ) a programmé en colonie .
- Les deux derniers ( SEB et CYR ) ont programmé chez des copains .

La salle informatique comprend 7 MICRAL 80 22 G , et nous travaillons avec un langage LOGO qui ne **connait que les nombres entiers ...!!** ( Matériel Education Nationale ..).

Les séances ont généralement lieu le samedi en fin de matinée . Il est important de souligner que , malgré cet horaire peu propice en général à la concentration , les élèves n'ont jamais fait de difficultés pour travailler . Ils ont , sans aucun doute , eu du plaisir à participer à cette expérience .

## Présentation des différentes séances

Les sept séances présentées constituent la première phase de notre expérimentation. Rappelons notre projet ( C. Dupuis , M.A. Egret , D. Guin , 1985 , p.42 ) :

### - première phase :

**Enseignement de base du langage LOGO graphique et non graphique en insistant sur la programmation structurée .**

Les résultats d'une telle méthode enseignée dès le début de l'apprentissage en CM<sub>2</sub> ( D. Guin , J.G. Helm , 1984 ) , et les difficultés observées dans la préexpérimentation ( C. Dupuis , M.A. Egret , D. Guin , 1985 , p.8 ) nous permettent de penser qu'elle devrait s'avérer efficace en 4<sup>ème</sup> dans l'activité de programmation .

Ces sept séances se sont déroulées sur environ six mois , à raison de deux heures par quinzaine . Elles ont, bien sûr , été de durée inégale . ( Voir , en annexe 2 , page 81 , un planning des différentes séances ) .

### Recommandations données aux élèves pour toutes les séances :

- Nous leur demandons une **analyse avec papier-crayon** suivie d'un travail sur micro-ordinateur et d'éventuelles corrections à l'aide de l'éditeur LOGO , ce qui permet parfois des séances sans micro-ordinateur .

- L'analyse des problèmes peut être faite par binôme ( sauf lors des contrôles ) .

- Un cahier de cours et un cahier de recherche sont demandés. Toutes les fiches ( cours et travaux pratiques ) y seront collées .

## Séance 1 : FLEUR

### Objectifs de la séance :

- prise de contact avec les élèves .
- aperçu des connaissances en programmation LOGO des élèves .
- notion de procédure .
- introduction à la programmation structurée : présentation des arbres ( D. Guin , J.G. Helm , 1984 ) .

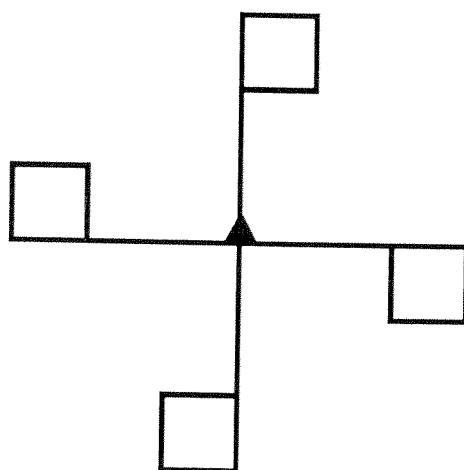
### Consigne :

\*\*\*\*\*

---

En faisant appel à vos " souvenirs " , écrivez une suite d'instructions permettant d'obtenir le dessin ci-dessous .

### Exercice 1 : fleur



▲ : positions initiale  
et finale

\*\*\*\*\*

---



Analyse de la situation :

A la suite d'observations réalisées durant la préexpérimentation , nous indiquons explicitement les **positions initiale et finale** de la tortue dans les projets graphiques . En effet, dans un premier temps nous ne précisons pas ces positions et nous nous sommes rendues compte que cette absence d'indications était une source fréquente d'erreurs .

Cette indication peut disparaître lorsque les élèves ont pris conscience de la nécessité de la **prise en compte de l'état de la tortue** au début et à la fin d'un tracé , en particulier si celui-ci correspond à une sous - procédure destinée à être **intégrée** à une procédure .

Nous avons remarqué que certains élèves ne s'éloignent que progressivement d'une conception "image" de la procédure : nous désignerons par **conception "image"** une **identification** de la **procédure** avec le tracé souhaité ou la **figure** obtenue . Les premiers programmes des élèves reflètent tous cette conception qui est un obstacle à la programmation en Logo .

Dans le choix des activités proposées aux élèves , nous avons veillé à mettre en échec assez rapidement une conception "image" de la procédure . Nous pensons comme J. Hillel et R. Samurçay ( 1985 ) que : " Ce n'est que lorsque l'interaction avec l'ordinateur évolue et met en évidence des **différences** entre ce qui est **attendu** et ce qui est **obtenu** que les enfants prennent en compte , graduellement , les **contraintes** introduites par la programmation " .

Certains élèves ont remarqué la répétition du dessin drapeau , mais ne se souviennent plus de l'instruction **REPETE** . Ils décrivent cependant le dessin du drapeau en expliquant :

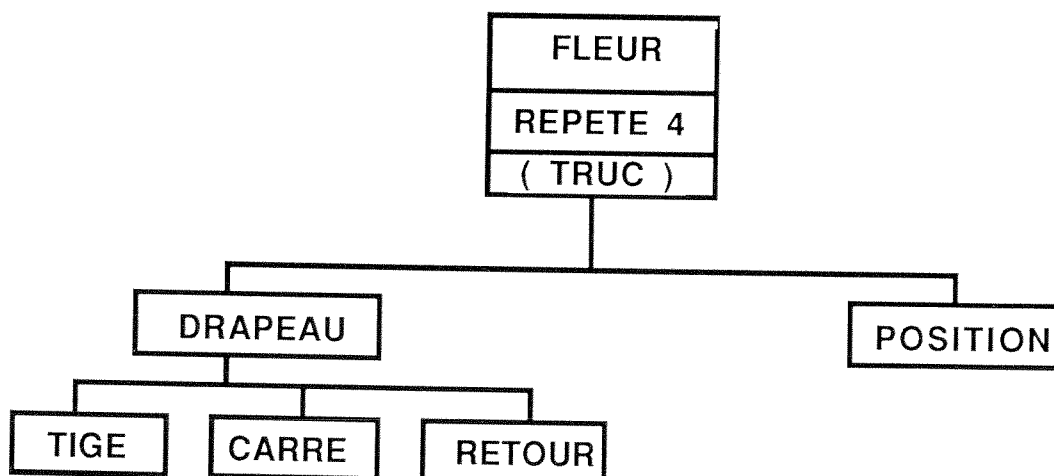
*" Pour faire la fleur , je ferai le drapeau , et je recommencerai ensuite trois fois "*

Ceci correspond au **modèle spontané de la répétition : faire une action et recommencer** . Nous supposons , comme R. Samurçay et A. Rouchier ( 1987 ) , que ce modèle est disponible chez des élèves de cet âge . 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* " .

Beaucoup d'élèves n'hésitent pas à aligner une page entière d'instructions : " TD 90 AV 45 etc... ", ce que nous appellerons écrire un programme "spaghetti". Ce type de programme disparaît lorsque les situations deviennent plus complexes , si la possibilité de **structurer** est présentée aux élèves et admise par eux comme une technique plus économique . Ceci nécessite un certain nombre d'heures de programmation et un **choix de situations** rendant difficile la programmation "spaghetti".

Un corrigé de l'exercice 1 est donné au tableau par le professeur . A cette occasion , elle explique aux élèves comment écrire une procédure LOGO . Elle souligne le choix de l' **invariant** ( dessin du drapeau ) et décompose le programme de la manière suivante :

### Arbre



#### POUR POSITION

TD 90

FIN

#### POUR FLEUR

REPETE 4 [ DRAPEAU POSITION ]

FIN

#### POUR CARRE

REPETE 4 [ AV 5 TD 90 ]

FIN

#### POUR DRAPEAU

TIGE CARRE RETOUR

FIN

POUR TIGE

AV 10

FIN

POUR RETOUR

RE 10

FIN

*Remarques :* - L' **invariant** correspond à la procédure DRAPEAU .

- Réaliser la **coordination** entre deux procédures consiste à écrire la procédure POSITION que nous appellerons **interface** . ( Cette interface peut n'être qu'une liste d'instructions ) .

- Dans DRAPEAU , la position initiale de la tortue est identique à sa position finale . La procédure ainsi écrite est , relativement , indépendante du contexte et peut être facilement réutilisée pour d'autres projets , par exemple en changeant l'interface POSITION et le nombre de répétitions . Nous dirons dans ce cas que la procédure comporte un **retour à une position " standard "** . Le retour à une position " standard " est l'un des types de prise en compte de l'état de la tortue . Un autre type de prise en compte consisterait à intégrer l'interface dans la procédure DRAPEAU , ce qui lui ferait perdre son caractère d'indépendance . ( Nous reprenons cette distinction pour nos observations dans le critère ETAT p .62 ) .

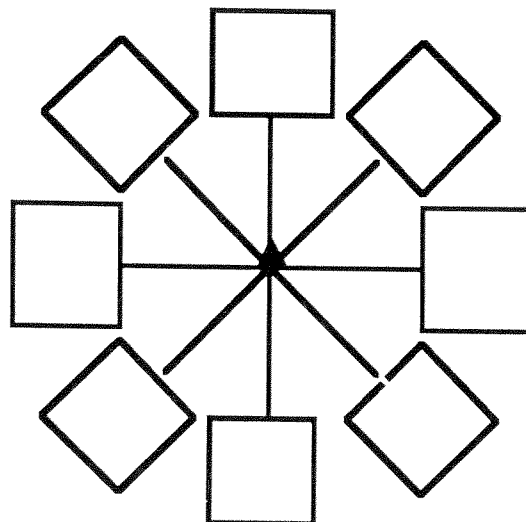
Séance 2 : MOULINObjectifs de la séance :

- notion de procédure .
- introduction à la programmation structurée .
- utilisation de l'instruction **REPETE** .
- utilisation de l'éditeur LOGO .

Consigne :

\*\*\*\*\*

La recherche de l'exercice 2 ayant été achevée avec papier-crayon , testez le programme sur l'ordinateur et corrigez-le éventuellement à l'aide de l'éditeur LOGO .

Exercice 2 : moulin

▲ : positions  
initiale et  
finale

\*\*\*\*\*

Analyse de la situation :

Pour évaluer la compréhension de la séance 1 , nous proposons aux élèves un travail analogue choisi de manière à ce que la décomposition de la figure et l'utilisation de l'instruction REPETE y soient réellement utiles .

Commentaires :

- Les élèves choisissent bien l'**invariant** , mais ils ont des difficultés à écrire la procédure correspondante ( mauvaise **coordination** entre le carré et la tige ) .

- Les élèves ont des problèmes liés à la manipulation de l'éditeur .

- L'instruction **REPETE** présente des difficultés d'utilisation : certains élèves utilisent cette instruction uniquement en se référant au **modèle spontané de la répétition** ( cf p.8 ) , ce qui les amène à écrire une liste d'instructions correspondant à l'invariant puis REPETE 7 fois la même liste d'instructions plutôt que REPETE 8 fois . Cette mauvaise utilisation ne disparaîtra pas rapidement ( cf test 1 p.49 et séance 7 p.30 ) .

Remarquons qu'à ce stade très peu d'élèves ont un programme " qui tourne " dès la première exécution .

### Séance 3 : TANGRAM

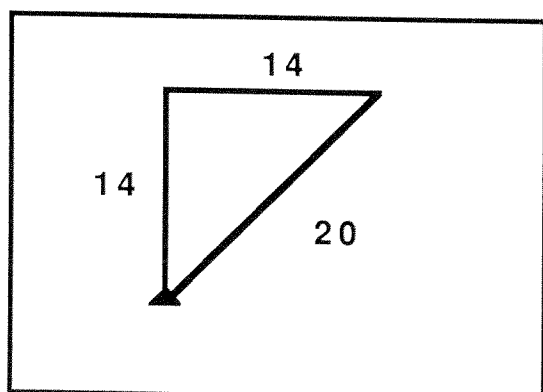
#### Objectifs de la séance :

- programmation structurée : mise en évidence d'invariants , construction d'arbres .

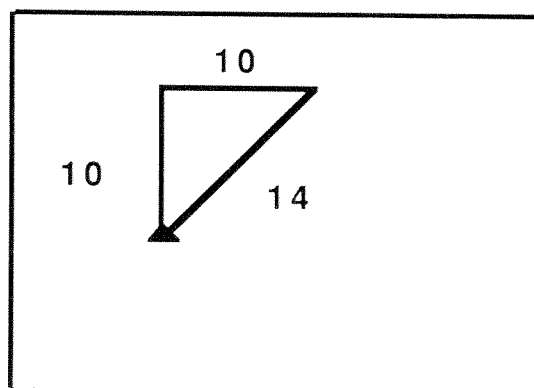
#### Consignes :

\*\*\*\*\*

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



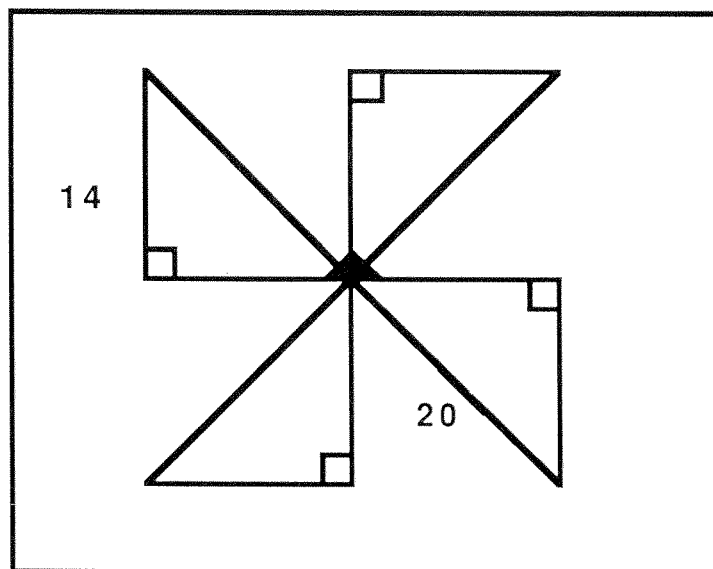
PIECE1



PIECE2

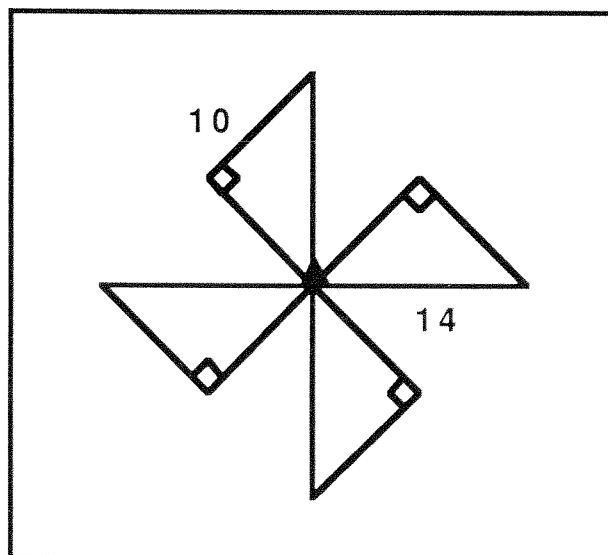
*Dans les deux cas , positions initiale et finale de la tortue : origine , cap 0 .*

2) Dessinez l'arbre , puis écrivez le programme qui permet d'obtenir le dessin suivant :



DESSIN1

3) Dessinez l'arbre , puis écrivez le programme permettant d'obtenir ce dessin :

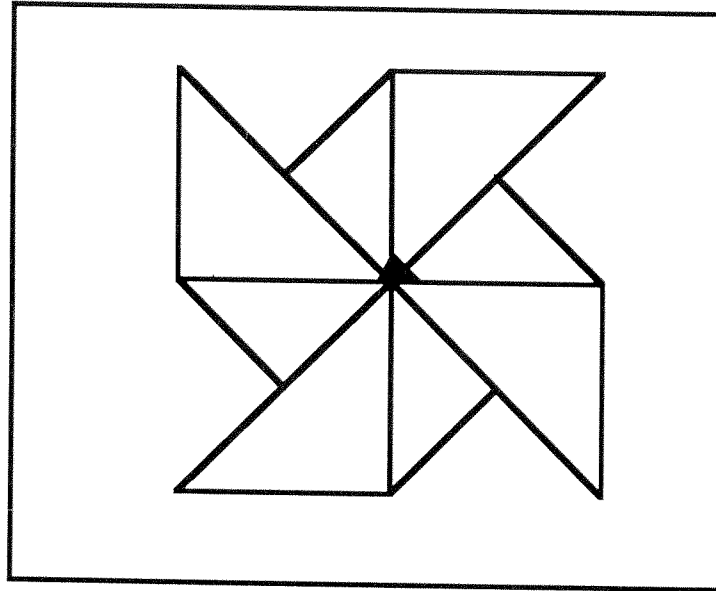


DESSIN2

dans 2) et 3) : positions initiale et finale de la tortue : origine , cap 0 .



4) Voici une figure obtenue à l'aide des pièces du **TANGRAM** ; construisez l'arbre et écrivez le programme :



**FIGURE**

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

5) Imaginez un dessin utilisant **uniquement** les pièces du **TANGRAM** . Construisez l'arbre et écrivez le programme .

\*\*\*\*\*

*Analyse de la situation :*

Nous recherchons des situations où la **structuration** du programme soit ressentie non comme une contrainte supplémentaire imposée par l'enseignant , mais comme **une aide effective** à la réalisation du projet . 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 . Le but de la consigne 1 est d'enlever les difficultés liées au choix de la décomposition . Il n'y a plus que des **problèmes de coordination** entre les différentes procédures .

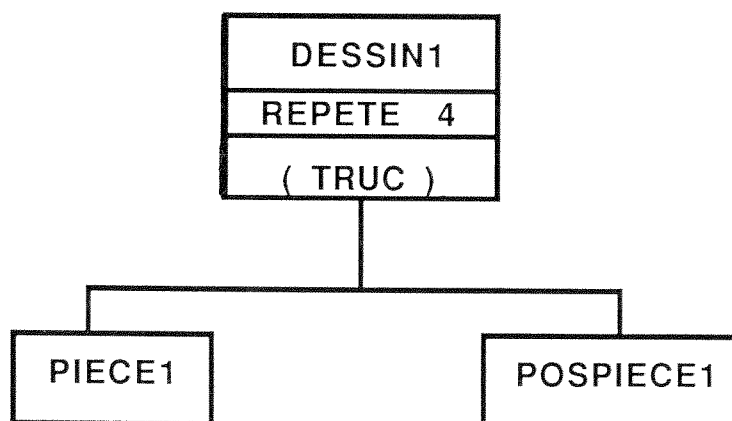
### Commentaires

Les élèves , par groupe de deux , cherchent les premiers exercices puis préparent un projet utilisant toutes les pièces du TANGRAM .

Les élèves testent leur programme : il y a quelques difficultés dues à l'utilisation de l'éditeur et au **manque de prise en compte de la position finale** de la tortue . On remarque que les élèves ont de la peine à comprendre la nécessité de préciser les positions initiale et finale de la tortue , puisque le **dessin** obtenu est identique au modèle . L'un d'entre eux utilise même dès le départ l'instruction **CT** ( cache-tortue ) . Par contre , les valeurs des angles sont en général correctes car une séance d'estimation d'angles a eu lieu quelques jours avant en cours de Mathématiques .

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 .

Une correction est proposée aux élèves :



#### 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 ]

FINPOUR POSPIECE1

TD 90

FINPOUR DESSIN2

POSDEPART

REPETE 4 [ PIECE2 POSPIECE2 ]

POSRETOUR

FINPOUR POSDEPART

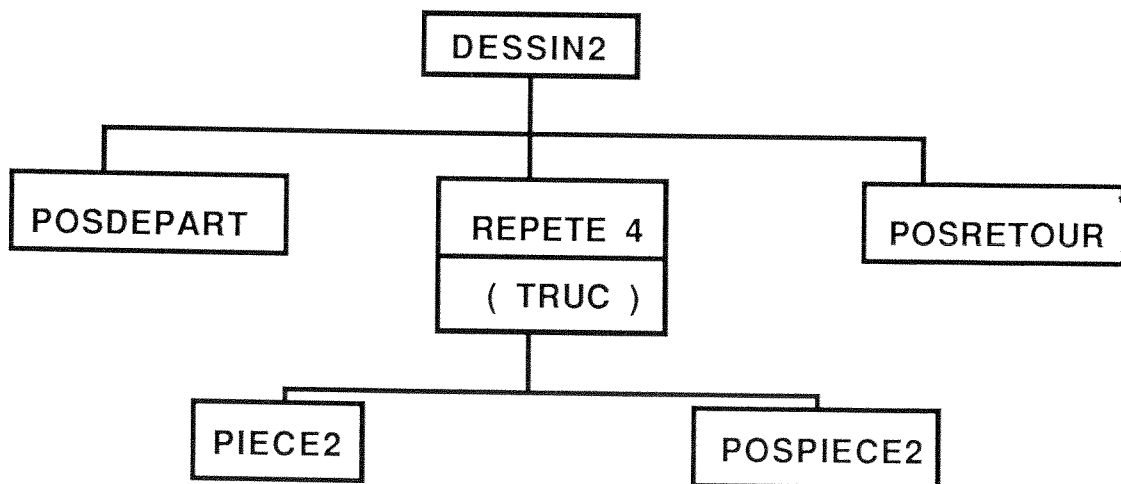
TG 45

FINPOUR POSPIECE2

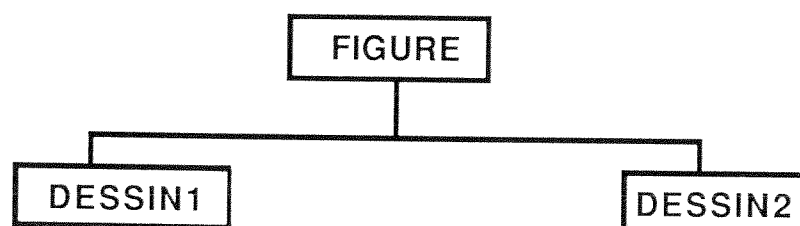
TG 90

FINPOUR POSRETOUR

TD 45

FINPOUR FIGURE

DESSIN1 DESSIN2

FIN

Pour les projets ( consigne 5 ) , seuls trois élèves écriront immédiatement un programme correct . Chez les autres élèves , on peut mettre en évidence trois types de difficultés :

**- des difficultés liées au choix du projet :**

Certains élèves , ayant bien compris la consigne , et ayant bien structuré leur programme n'aboutissent pas , à cause de la complexité de leur projet .

**- des difficultés liées au manque de compréhension des consignes :**

Certains élèves ne respectent pas la consigne : utiliser les programmes PIECE1 et PIECE2 . Après une mise au point , le travail est repris et bien compris , même s'il n'est pas terminé à la fin de la séance .

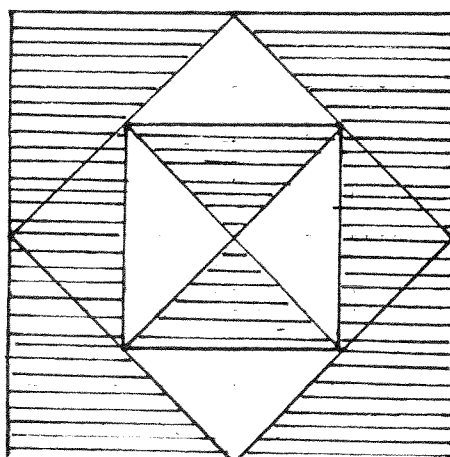
Signalons qu'une élève n'a utilisé que deux grands triangles et deux petits triangles pour son projet . Comme son programme est juste , elle propose de compléter par symétrie pour utiliser toutes les pièces du TANGRAM .

**- autres remarques :**

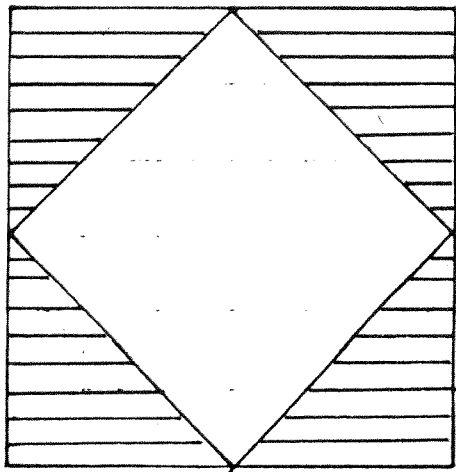
Notons que deux binômes acceptent mal le travail papier-crayon et la programmation structurée : ils se découragent car les programmes doivent sans cesse être corrigés .

A la fin de la séance, seuls quatre élèves n'ont pas vu leur projet aboutir . Un des binômes , plus lent ( du fait de difficultés dans le maniement de l'éditeur ) a un programme presque exact . L'autre a choisi une figure complexe où la mise en place de la tortue n'est pas toujours évidente .

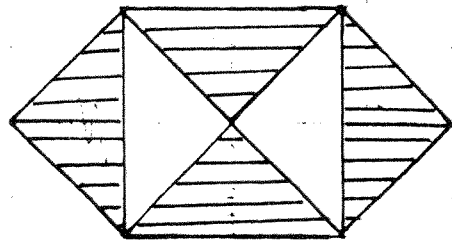
FIGURE :



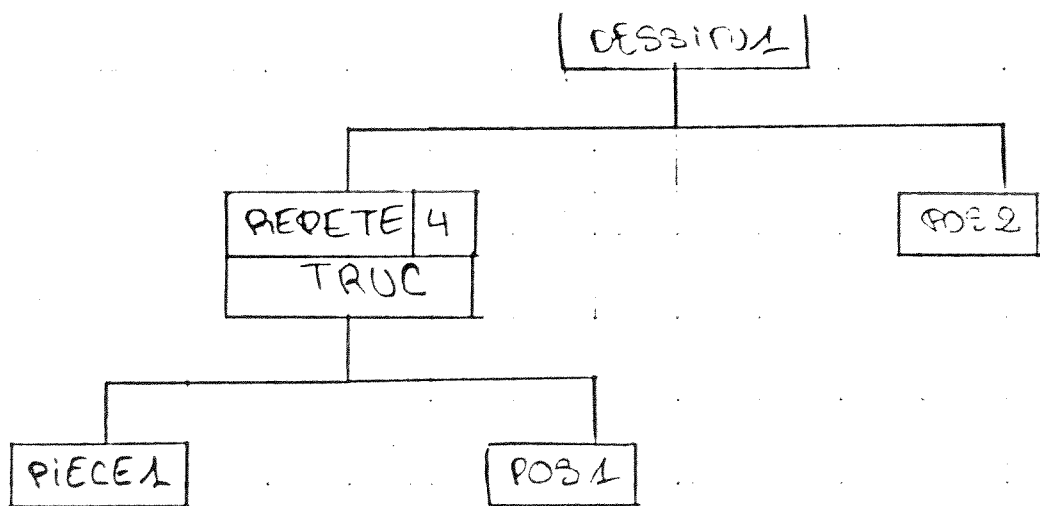
DESSIN 1 :



DESSIN 2 :



Arbre pour DESSIN 1 :



POUR POS 1

AV 14 TO 98 AV 14

FIN

POUR POS 2

TO 135 AV 10 IG 195

FIN

## Séance 4 : POLYS

### Objectifs de la séance :

- notion de variable ( cf. fiche de cours " Généralisons nos procédures " p. 74 ) .
- procédure paramétrée .

### Consignes :

\*\*\*\*\*

---

1 - Ecrivez un programme permettant d'obtenir le dessin d'un polygone régulier dont le nombre de côtés et la longueur du côté sont quelconques .

2 - Ecrivez un programme permettant d'obtenir le dessin d'un cercle à l'écran .

\*\*\*\*\*

---

### Analyse de la séance :

Il nous a paru intéressant que les élèves trouvent la relation entre l'angle de rotation de la tortue et le nombre de côtés du polygone . Nous leur avons demandé de présenter leurs résultats sous forme d'un tableau : cf. document p .21 .

### Commentaires :

L'utilisation de variables semble poser peu de problèmes dans cet exercice : il s'agit, en fait , de désigner des longueurs ou des angles par une lettre ou un nom , comme C pour la longueur du côté du polygone ou N pour le nombre de côtés ; l'écriture du ":" devant le nom de la variable est souvent omise par les élèves .

Pour trouver la formule entre angle de rotation de la tortue et nombre de côtés du polygone , les élèves procèdent par tâtonnement ou par raisonnement sur la proportionnalité

( exemple : pour un triangle équilatéral la tortue tourne de  $120^\circ$ , pour un hexagone, elle va tourner de  $120 / 2 = 60$  ).

Une mise au point est faite par le professeur . Le tableau ayant été construit dans les cahiers ,

	nombre de côtés	angle de rotation
triangle	3	$120^\circ$
carré	4	$90^\circ$
pentagone	5	$72^\circ$
hexagone	6	$60^\circ$
octogone	8	$45^\circ$
nonagone	9	$40^\circ$
décagone	10	$36^\circ$
dodécagone	12	$30^\circ$

Donc pour trouver la valeur d'un angle de rotation, il suffit de diviser  $360^\circ$  par le nombre de côtés de la figure.

les élèves écrivent les relations  $n = 360 / a$  ou  $a = 360 / n$  (  $a$  : angle de rotation de la tortue,  $n$  : nombre de côtés du polygone ).



Les deux programmes suivants sont alors proposés :

```
POUR POLY : COTE : N  
REPETE : N [ AV : COTE TD 360 / : N ]  
FIN
```

```
POUR POLYG : COTE : ANGLE  
REPETE 360 / : ANGLE [ AV : COTE TD : ANGLE ]  
FIN
```

Les élèves affirment à ce moment-là bien comprendre comment dessiner un cercle . Nous verrons , dans le test 2 , que les élèves n'ont pas pris conscience de la position du cercle par rapport aux positions initiale et finale de la tortue . Pour tracer **un** cercle ou **un** polygone **isolés** , il n'est pas nécessaire de percevoir cet aspect .

## Séance 5 : CARRÉS

### Objectifs de la séance :

- utilisation des variables .
- structuration des programmes : coordination de sous-procédures .

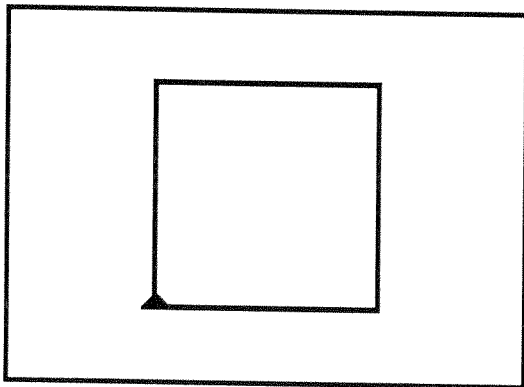
### Consignes :

\*\*\*\*\*

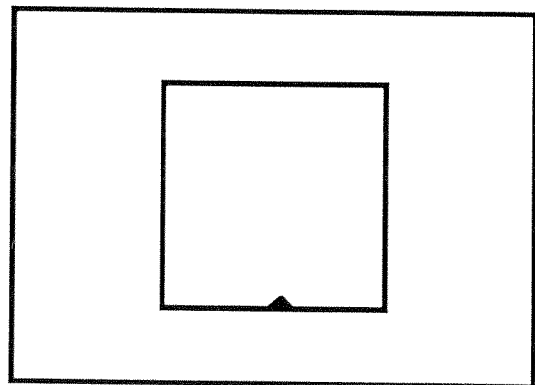
Le but de cette séance est d'écrire **de la façon la plus structurée possible** les programmes permettant de réaliser les dessins 1 , 2 et 3 . Pour cela , nous vous demandons de procéder par étapes :

### Etape 1 :

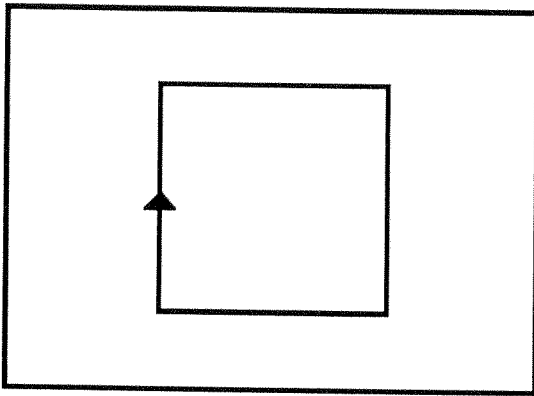
Ecrivez les trois procédures qui permettent la réalisation de ces carrés :



CARRE1



CARRE2



▲ : position s  
initiale et finale .

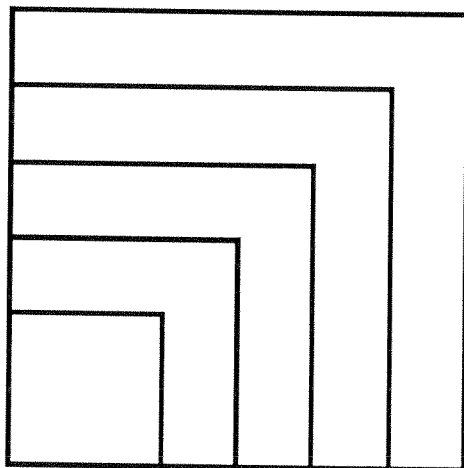
**CARRE3**

Etape 2 :

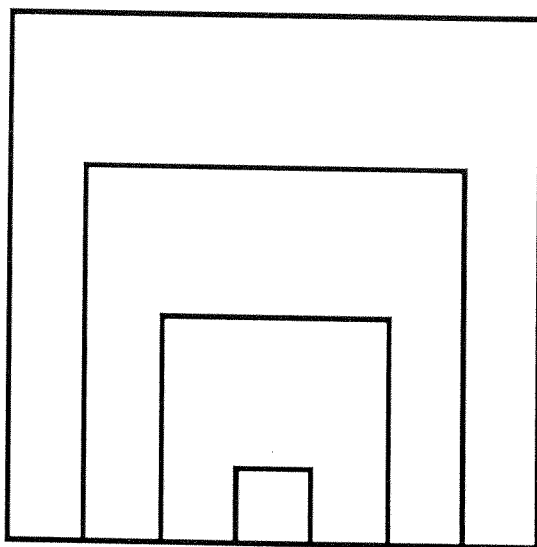
Ecrivez les procédures CARRE2 et CARRE3 en utilisant la procédure CARRE1 .

Etape 3 :

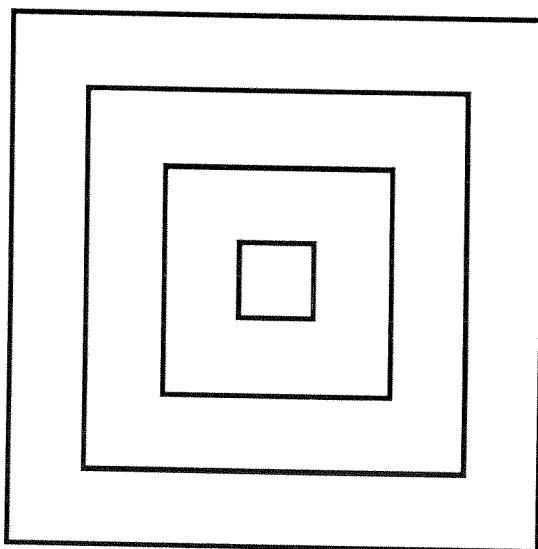
Ecrivez les procédures permettant d'obtenir à l'écran DESSIN1, DESSIN2 et DESSIN3 .



**DESSIN1**



DESSIN2



DESSIN3

\*\*\*\*\*

Commentaires :

L'étape 2 est très bien comprise et réalisée sans difficultés par les élèves ( qui ont même tendance à ne pas réaliser l'étape 1 pour CARRE2 et CARRE3 ). Le lien est bien fait

entre les étapes précédentes et l'étape 3 . Les élèves trouvent assez rapidement quel est le carré le plus adapté à leur tâche . Certains ( 4 ou 5 ) , cependant , utilisent dans un premier temps CARRE1 pour les trois dessins .

Il faut remarquer aussi que pour l'étape 3 , plus complexe que les deux autres , la plupart des élèves préfèrent au départ écrire DESSIN1 , DESSIN2 et DESSIN3 **sans variables** ( aucune précision de mesure ne leur était fournie sur la fiche distribuée ) . A notre demande , ils écrivent des programmes utilisant des variables . DESSIN3 leur apparaît alors plus difficile , car il faut d'une part modifier la position de la tortue à chaque carré et d'autre part exprimer correctement la longueur du côté du second carré **en fonction de** la longueur du côté du premier carré .

Exprimer **une relation entre des variables** est une tâche plus difficile qu'affecter des valeurs , respectant cette relation , à des variables différentes : nous aurons l'occasion d'approfondir ce point .

Quelques élèves , plus rapides , écrivent le programme permettant d'obtenir le tracé de la TOUR ( C. Dupuis , M.A. Egret , D. Guin , 1985 , p . 12 ) . Deux élèves font alors la remarque : " C'est presque comme s'il y avait un REPETE ; il n'y aurait pas autre chose qu'on pourrait utiliser ? " . Pendant ces deux séances , les problèmes d'éditeur ont disparu .

**ETAPE 3**

POUR DESSIN 1.

CARRE 1 30

" 40

" 50

" 60

" 70

FIN

POUR DESSINUS : COTE

CARRE 2 : COTE

CARRE 2 : COTE + 20

CARRE 2 : COTE + 40

CARRE 2 : COTE + 60

FIN

POUR DESSIN 3 : C

> CARRE 3 : C

> POS : C

> CARRE 3 : C \* 3

> POS : C

> CARRE 3 : C \* 5

> POS : C

> CARRE 3 : C \* 7

> FIN.

POUR POS : C

> TG 90

> LC

> AV : C

> BC

> TD 90

> FIN.

## Séance 6 : MOTS ET LISTES

### Objectif de la séance :

- Mots et listes ( cf. fiche " Mots et listes " p. 77 ) .

### Consigne :

\*\*\*\*\*

---

Cherchez les exercices proposés :

- concaténer plusieurs mots ( exemple : écrire BICYCLETTE à partir de BI , CY, CLET, TE ... ) ;
- rechercher une lettre dans un mot ;
- composer les instructions DER , PREM , SD , SP ( exemple : écrire le mot PAS à partir du mot DEPASSER .... ) ;
- rechercher un mot dans une liste ( exemple : écrire le mot CHINOIS à partir de la liste : LA PETITE CHINOISE COURT ) .

\*\*\*\*\*

---

### Commentaires :

Un peu déconcertés au départ par les exercices proposés ( absence de réalisation graphique ) , les élèves répondent sans difficultés et avec plaisir aux questions .



## Séance 7 : RELIE

### Objectif de la séance :

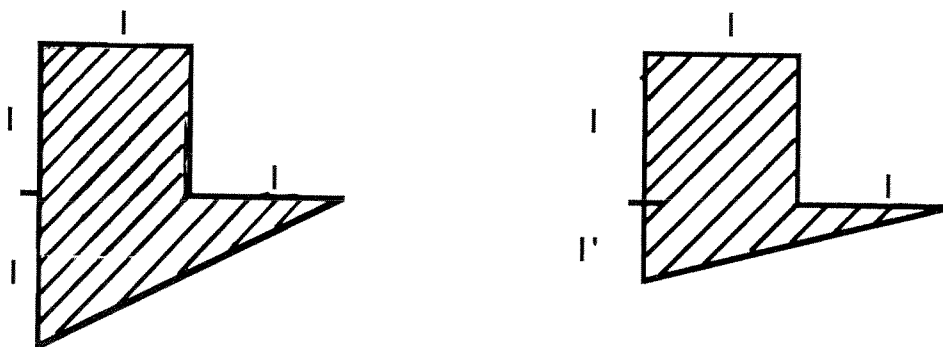
- Instruction RELIE .

### Consigne :

\*\*\*\*\*

Résolvez les exercices proposés après avoir lu attentivement la fiche "instruction RELIE " ( cf. p. 79 ) .

#### 1. Calculs d'aires .



#### 2. Ecrivez un programme qui permette d'obtenir :

- l'affichage d'un nombre , puis celui des neuf nombres pairs suivant ce nombre .
- l'affichage des dix nombres pairs suivant un nombre choisi ( ce nombre n'étant pas affiché ) .

3. Un marchand vend à un client des pêches et des cerises, par exemple :

3 kg de pêches à 6 F le kilogramme ;

1 kg de cerises à 9 F le kilogramme .

a) Ecrivez le lexique de ce problème ( \* ) .

b) Ecrivez une procédure permettant de créer une boîte pour chaque donnée numérique .

c) Ecrivez une procédure permettant , une fois la procédure b) exécutée , de donner le montant de la facture et de l'afficher à l'écran .

4. Ecrivez un programme permettant d'obtenir l'affichage de la moyenne de 6 notes données sous forme d'une liste .

\*\*\*\*\*

( \* voir D. Guin , J.G. Helm , 1984 , p. 31 )

### Commentaires :

Les élèves travaillent , mais progressent difficilement . Ils ont cependant tous cherché les exercices 1, 2 et 3 et les ont testés sur les micro-ordinateurs , très souvent en mode direct , le mot PROGRAMME n'étant pas explicitement indiqué dans l'exercice 1 .

Deux types d'erreurs apparaissent :

#### - 1. Mauvaise utilisation de l'instruction REPETE :

Au lieu d'utiliser REPETE n fois une liste d'instructions , les élèves écrivent une première fois la liste puis REPETE n -1 fois cette liste ( c'est le **modèle spontané** que nous avons déjà mis en évidence ( cf. p. 8 ) ) .

#### - 2. Mauvaise compréhension de l'instruction RELIE :

Les élèves n'utilisent pas l'instruction RELIE sans l'instruction ECRIS . Cette erreur est peut-être due aux exemples présentés dans la fiche de cours .

Une erreur supplémentaire est apparue dans l'exercice 3 :

**- 3. Mauvaise représentation de l'exécution d'un programme :**

Les élèves , après avoir écrit un programme DONNEE , puis un programme DEPENSE utilisant des variables dont les valeurs sont entrées dans DONNEE , ne font pas exécuter DONNEE avant DEPENSE . Le message d'erreur :

*" dans DEPENSE PRIXPECHE n'a pas de valeur "*

les met alors sur la voie .

Seuls , trois élèves ont cherché l'exercice 4 avec l'aide du professeur . Cet exercice était d'une difficulté nettement supérieure , et peu accessible à ce niveau .

I. a) POUR MACHIN

RELIE "L 20

RELIE "A 2 x (:L x :L)

ECRIS : A

FIN

b) POUR NOMBRES

RELIE "M 25

RELIE "M' 20

RELIE "N (:M x :M) + (:M' x :M)

ECRIS : N

FIN

II a) POUR TRUC

RELIE "B 4 ECRIS : B

REPETE 9 [RELIE "B : B+2 ECRIS : B]

FINb) POUR BIDULE

RELIE "C 2

REPETE 10 [RELIE "C : C+2 ECRIS : C]

FINIII  
1)

	Nom du tableau	contenu
Données	Q . P	3
	P . P	6
	Q C	1
	P C	9
Résultats	D (dépense)	? Dépense (nombre)

2) pour données

relie "Q P 3

relie "P P 6

relie "Q C 1

relie "P C 9

FINpour dépenses

relie "A ( : Q P x : P P )

(écrit : A)

relie "B ( : Q C x : P C )

(écrit : B)

relie "C ( : A + : B )

écrit : C

FIN

## Présentation des différents tests

( On trouvera en annexe 2 ( p. 81 ) le planning des séances et des tests . )

### Test 1 :

#### 1 - Consigne :

\*\*\*\*\*

#### 1 - COMPREHENSION DE PROGRAMMES :

*A lire attentivement :* Les rectangles qui figurent sur les pages suivantes représentent l'écran de l'ordinateur . Une suite d'instructions **LOGO** est proposée sous chaque rectangle . Il s'agit de remplacer la tortue par ton crayon , et d'exécuter les instructions **LOGO** .

- La position de départ de la tortue est indiquée .
- 10 pas de tortue la font avancer de 1 cm .
- Il ne faut pas oublier d'indiquer la **position finale** de la tortue , après l'exécution des instructions .

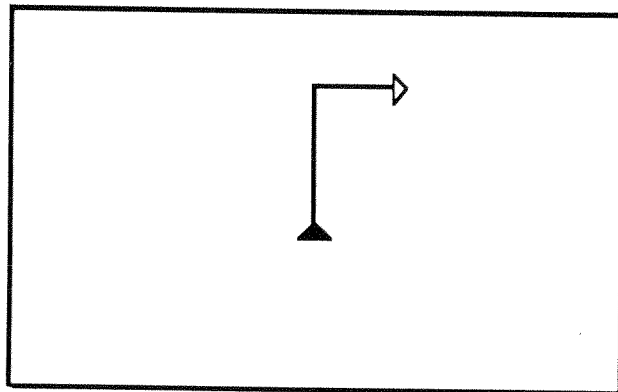
#### Exemple :

Position initiale

de la tortue : ▲

Position finale

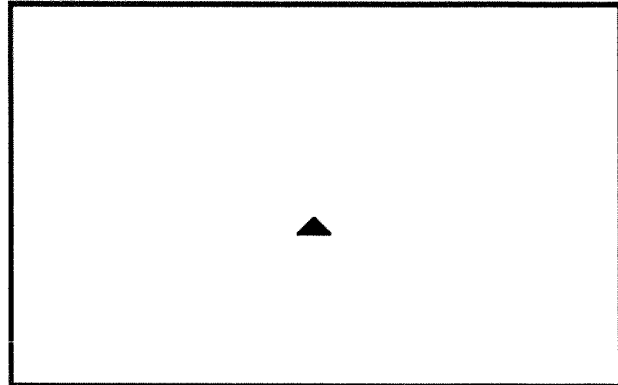
de la tortue : △



Dessine le trajet de la tortue dans les cas suivants :

- Dessin 1 :

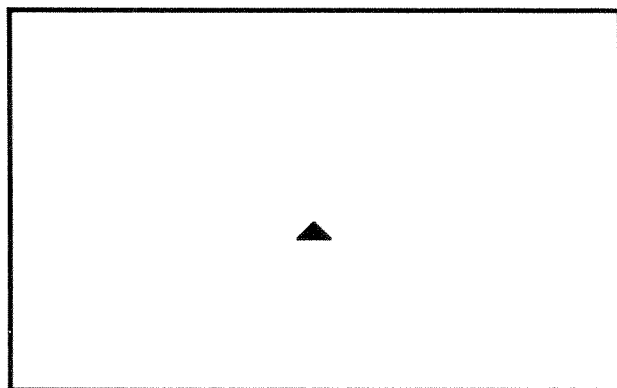
Position initiale  
de la tortue : ▲



AV 10 RE 20

- Dessin 2 :

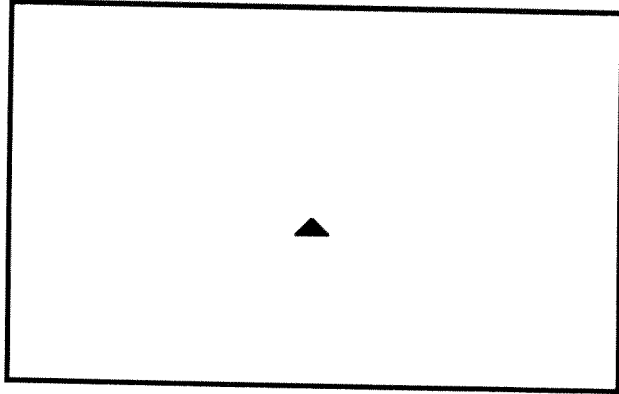
Position initiale  
de la tortue : ▲



AV 20 TD 90 AV 20 TD 90 AV 10

- Dessin 3 :

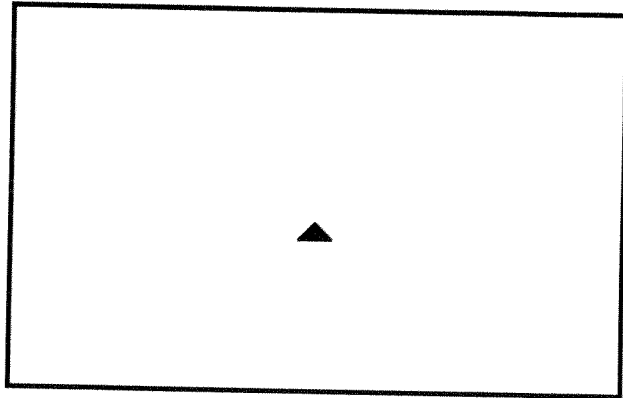
Position initiale  
de la tortue : ▲



TG 45 AV 20 RE 20 TD 90 AV 20 RE 20

- Dessin 4 :

Position initiale  
de la tortue : ▲

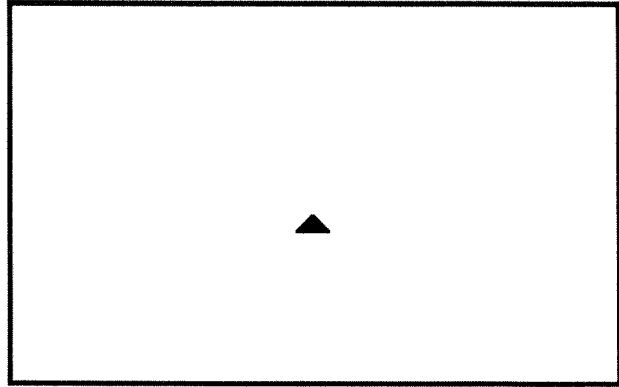


AV 20 LC TD 90 AV 20 BC



- Dessin 5 :

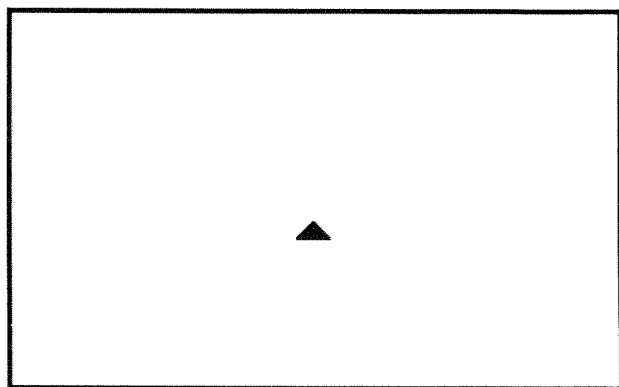
Position initiale  
de la tortue : ▲



TD 90 AV 20 TG 135 AV 10 LC RE 10  
TD 45 BC

- Dessin 6 :

Position initiale  
de la tortue : ▲



AV 20 TG 135 AV 28 TD 90 RE 28 TD 45  
RE 10

## 2. COORDINATION DE PROCEDURES :

*A lire attentivement* : L'écran de l'ordinateur est représenté par un rectangle . Dans chaque exercice , nous te proposons un dessin réalisé par la tortue sur l'écran , et le texte d'une ou plusieurs procédures . En te **servant au moins une fois** des procédures proposées , tu dois écrire un programme qui permet à la tortue de réaliser le dessin proposé .

- Les positions initiale et finale de la tortue sont indiquées .
- 10 pas de tortue la font avancer de 1 cm .
- **N'oublie pas de dessiner l'arbre du programme .**

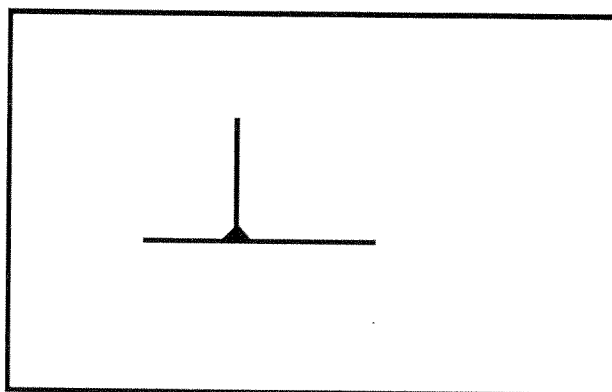
### - Exercice 1 : TIGES

Voici 3 procédures :

<u>POUR TIGE20</u>	<u>POUR TIGE10</u>	<u>POUR TIGE15</u>
AV 20 RE 20	AV 10 RE 10	AV 15 RE 15
<u>FIN</u>	<u>FIN</u>	<u>FIN</u>

Voici le dessin que l'on veut obtenir à l'écran :

Position initiale et  
finale de la tortue: ▲



Dessine l'arbre et écris le programme réalisant ce dessin en tenant compte des consignes .

-Exercice 2 : SAPIN

Voici 2 procédures :

POUR TIGE17

AV 17

FIN

POUR TRIANGLE

TG 90

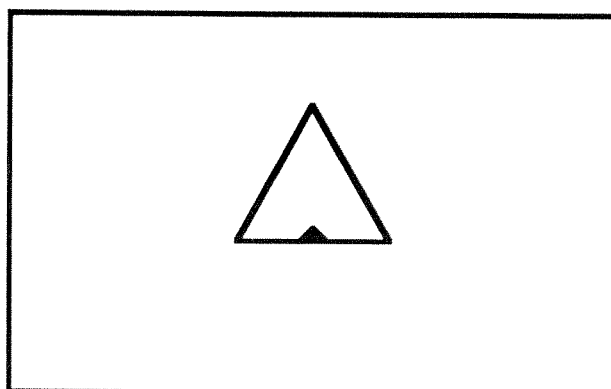
REPETE 3 [ AV 10 TD 120 AV 10 ]

TD 90

FIN

Voici la réalisation à l'écran de la procédure TRIANGLE :

Position initiale et  
finale de la tortue : ▲



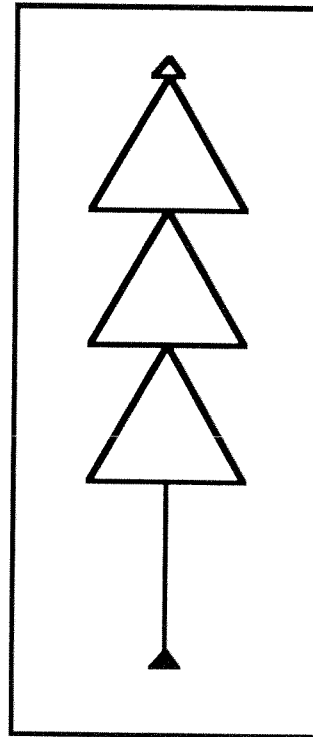
Voici le dessin que l'on veut obtenir à l'écran :

Position initiale

de la tortue : ▲

Position finale

de la tortue : ▲



Dessine l'arbre et écris le programme réalisant ce dessin en tenant compte des consignes .

### 3. COMPLETION DE PROGRAMMES :

*A lire attentivement* : L'écran de l'ordinateur est représenté par un rectangle . Nous te proposons un dessin réalisé par la tortue sur l'écran et un programme **LOGO** . Il faut que tu trouves si le programme donné permet de réaliser le dessin proposé . Si ce n'est pas le cas , tu indiques les **instructions qu'il faut ajouter** au programme , et **leur place** pour obtenir le dessin souhaité .

- Les positions initiale et finale de la tortue sont indiquées .
- 10 pas de tortue la font avancer de 1 cm .

#### POUR DRAPEAU

AV 20 TD 60

REPETE 2 [ AV 10 TG 120 ]

AV 10

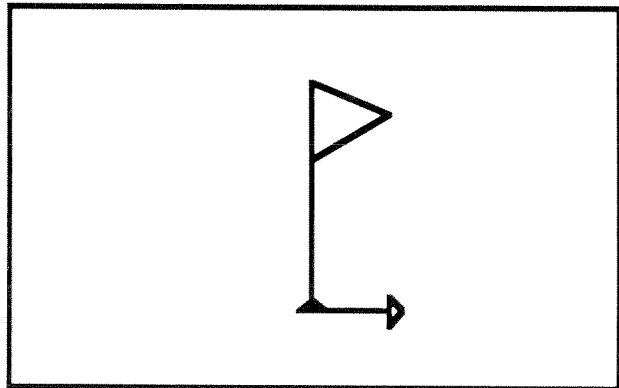
FIN

Position initiale

de la tortue : ▲

Position finale

de la tortue : ▲



Ce programme permet-il d'obtenir le dessin ci-dessus ? Corrige- le éventuellement .

\*\*\*\*\*

## 2 . Analyse des questions :

Le test comporte trois types de questions , correspondant aux trois parties :

### 1 - Compréhension de programmes

### 2 - Coordination de procédures

### 3 - Complétion de programmes

L' activité d'un élève , lorsqu'il écrit un programme et qu'il dispose de l'ordinateur pour exécuter son programme , n'est pas du même ordre que dans ce type d' épreuve . Les différences sont nombreuses .

Dans ce test " papier-crayon " , l'élève doit au début lire et comprendre des textes de programmes qui lui sont fournis . Il sait qu'il ne pourra pas tester son programme . De plus , on lui demande :

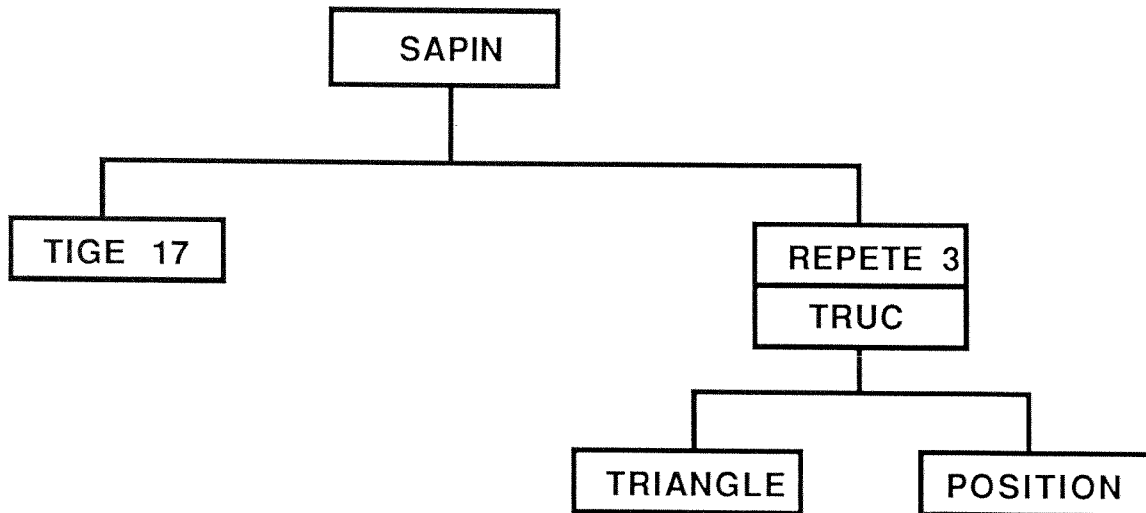
- en ( 1 ) de jouer le rôle de la tortue ,
- en ( 2 ) d'utiliser des procédures écrites par l'enseignant alors qu'il pourrait écrire un programme sans les utiliser ,
- en ( 3 ) de modifier un programme uniquement en y ajoutant des instructions en vue d' obtenir le résultat donné .

Si nous avons , après d'autres ( P. Mendelsohn , 1985 ; R.Samurçay , 1985 ) , choisi ce type de questions , c'est pour **mettre en évidence** les difficultés des élèves . Elles sont plus difficiles à distinguer lorsqu'elles sont mêlées dans une tâche d'écriture de programme et souvent inobservables lorsque l'élève renonce ou échoue . Ayant identifié des difficultés dans ce type d'épreuve , nous pourrions les rechercher dans des activités de programmation plus ouvertes .

Nous avons prévu un questionnaire initial " trop long " . Les élèves progressant lentement , nous avons choisi de le couper en deux ( après l'exercice 1 de la partie 2 ) . Un corrigé nous a permis d'attirer leur attention sur les erreurs commises .

3. Corrigé intermédiaire :

L'arbre et le programme suivants sont proposés aux élèves :



POUR SAPIN

TIGE 17  
 REPETE 3 [ TRUC ]  
 FIN

POUR TRUC

TRIANGLE  
 POSITION  
 FIN

POUR POSITION

TG 90 AV 10 TD 120  
 AV 20 TG 30  
 FIN

Remarque :

Il faut rappeler que sur le micro-ordinateur MICRAL , LOGO ne connaît que les entiers ! Nous avons choisi , par conséquent , de ne pas introduire le théorème de Pythagore contrairement à ce qui avait été fait dans l'expérimentation précédente ( D. Guin , J.G. Helm , 1984 ) . Nous n'avons donc pas utilisé un déplacement de la tortue suivant la hauteur verticale du triangle .

#### 4. Analyse des réponses :

L'analyse des réponses des élèves suit l'ordre de présentation des questions .

##### 1 - Compréhension de programmes

Les dessins 1 à 4 ont été réussis par la majorité des élèves et aucun élève n'a échoué aux quatre tracés .

Pour le dessin 5 , la difficulté essentielle résidait dans la **position finale** de la tortue , qui suit une instruction **RE**cule : il y a eu huit réussites complètes , et cinq tracés corrects avec erreur sur la position finale .

Le dessin 6 n'est réussi que par cinq élèves sur dix-neuf . **Les erreurs sont toutes dues aux changements de cap de la tortue** et sont très variées ; sept élèves se trompent dès le premier changement de cap .

Cette suite d'instructions est , a priori , plus difficile à simuler que les autres parce que le premier changement de cap n'est pas un multiple de  $90^\circ$  et fait " revenir " la tortue vers l'élève . Les problèmes de latéralisation et d'estimation des angles sont alors plus grands que lors de déplacements verticaux ou horizontaux de la tortue . De plus , le dessin ne présente aucune symétrie .

Nous avons prévenu les élèves que le tracé pouvait sortir du cadre , celui-ci ayant été dessiné plus petit que l'écran pour des raisons d'économie... de papier .

##### 2 - Coordination de procédures

Cette partie comprend l'exercice 1 : TIGES et l'exercice 2 : SAPIN .



### Critères d'analyse des réponses :

Nous en définirons trois pour ce test . Dans les tableaux de résultats ci - après , nous désignerons les différentes modalités d'un critère par quelques lettres choisies pour rappeler la signification de cette modalité .

Ainsi , la première modalité : "Utilisation de **toutes** les procédures fournies " , du critère **OUTIL** ci - dessous sera désignée par les lettres : "**Tout**".

Nous espérons vous rendre ainsi plus agréable la lecture des tableaux !

### \* OUTIL ou modalité d'écriture du programme :

Ce critère est étroitement lié à la consigne qui précisait d'écrire le programme :

*" en te servant au moins une fois des procédures proposées "*

Les élèves ont-ils respecté cette consigne et si oui , est-ce avec succès ?

" **Tout** " : Utilisation de **toutes** les procédures fournies .

" **Part** " : Utilisation de certaines des procédures fournies .

" **Autr** " : Ecriture d'**autres** procédures .

" **Sans** " : Ecriture du programme **sans** aucune sous-procédure .

### \* COORDINATION de procédures :

Ce critère est d'une portée générale et sera aussi utilisé pour analyser des programmes entièrement écrits par les élèves .

La coordination des procédures pour obtenir le programme cherché est un problème qui se pose dès qu'il y a décomposition en sous-procédures . Ce critère est donc commun à beaucoup de situations .

Le problème de la coordination ne se pose pas si l'élève écrit un programme sans aucune sous-procédure ( modalité " **Sans** " du critère OUTIL ) ; 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 .

"0" : la coordination des procédures N'EST PAS REALISEE .

"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 , c'est à dire une identification de la procédure avec le tracé souhaité ( cf. p 8 ) .

Dans l'exercice 1 , les seules **erreurs simples** observées sont des erreurs de **latéralisation** ou des erreurs sur la **valeur d'un angle** .

Dans l'exercice 2 , on a observé des erreurs simples sur la **valeur d'un angle** ou sur les **relations** entre des longueurs ( passage par la hauteur du triangle ) .

Les **erreurs de conception** sont attestées par l'absence de **rotation** de la tortue entre les tiges à l'exercice 1 ou par l'absence de **déplacement** entre les triangles à l'exercice 2 .

Outil ou modalités d'écriture du programme dans les deux exercices :

exercice 2 ; SAPIN

	Tout	Part	Sans
	Tout	12	1
<u>exercice 1</u>	Part	2	
	Autr	3	
	Sans		1

Croisons les deux critères pour chaque exercice :

exercice 1 ; TIGES

		Coordination			
		C	ES	EC	0
	Tout	8	2	3	
Outil	Part		2		
	Autr				3
	Sans	1			

exercice 2 ; SAPIN

		Coordination		
		C	ES	EC
	Tout	3	14	
Outil	Part			1
	Sans			1

## Outil

Le premier tableau montre une **augmentation de l'utilisation des procédures** proposées . Remarquons qu'il reste encore deux élèves qui ne les utilisent pas **toutes** dans l'exercice 2 . Rappelons qu'entre-temps , il y a eu un corrigé indiquant la manière d'utiliser des procédures fournies .

La seule élève ( MAG ) qui avait écrit un programme sans aucune sous - procédure dans l'exercice 1 ( modalité " Sans " du critère Outil ) , avait réussi à obtenir le tracé ; malgré le corrigé et le rappel de la consigne , elle a conservé cette stratégie mais n'a pas pu écrire un programme correct dans l'exercice 2 .

**L'utilisation de procédures représente une réelle difficulté pour les programmeurs débutants ; cette technique doit être enseignée mais aussi justifiée par des situations suffisamment complexes pour que l'élève en ressente la nécessité . Pour certains élèves , 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 .**

## Coordination

" 0 " : On constate que tous les élèves ont écrit un programme dans l'exercice 2 , même les trois élèves qui , dans l'exercice 1 , avaient écrit d'autres procédures sans obtenir le programme complet .

" C " : Si l'on considère la réussite stricte " C " , l'exercice 1 est réussi par neuf élèves et l'exercice 2 par seulement trois ; mais si l'on élargit la réussite à ceux qui ne commettent que des erreurs simples ( " C " ou " ES " ) , l'exercice 2 , quoique plus difficile , est mieux réussi que l'exercice 1 .

## un critère pour la répétition

Par ailleurs , dans l'exercice 2 , une situation nouvelle apparaît : un élément se répète trois fois . 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 .

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

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 fausse .

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

Croisons -le avec le critère coordination dans l' exercice 2 :

		Coordination		
		C	ES	EC
	<b>00</b>		1	
	<b>CC</b>	2		
<b>Répète</b>	<b>CF</b>		11	1
	<b>FF</b>		1	1
	<b>XC</b>	1		
	<b>XF</b>		1	

Ce tableau montre que les erreurs simples de coordination ES ont lieu majoritairement dans la liste d'instructions à répéter ( codes CF, FF, XF ) .

Deux élèves ont écrit un programme en utilisant la structure " X " du REPETE . L 'un d'entre eux ( DAV ) a écrit un programme correct ( coordination C ) et économique , en répétant deux fois une séquence déplacement - triangle . Mais l'autre ( GIL ) répète trois fois de sorte qu'il obtient quatre triangles .

Il n'est pas surprenant de retrouver ici cette forme de répétition que nous avons déjà rencontrée et qui correspond au **modèle spontané** de la répétition , à l'expression " *et ainsi de suite* " ( cf. p. 8 ) . Si nous mettons ici en évidence ce phénomène qui peut paraître marginal , c'est parce que ce type d'utilisation du REPETE persiste assez longtemps et nous le retrouverons dans des observations ultérieures .

### 3 - Complétion de programmes

Tous les élèves ont répondu à cette question . Deux possibilités s'offraient pour compléter le programme :

- soit immédiatement après le REPETE [ ] par AV 30 TG 90 : deux élèves ont choisi cette possibilité .

- soit à la fin du programme en ajoutant AV 20 TG 90 AV 10 : six élèves ont choisi cette solution . Elle est plus simple à concevoir que la précédente puisqu'on ajoute ce qui manque pour terminer le tracé .

Deux élèves ont mesuré les angles du triangle au rapporteur . Ils ne l'ont pas jugé équilatéral et ont donc réécrit tout le programme . Leurs programmes sont parfaitement corrects . Ceci illustre bien une des difficultés que l'on rencontre dans la **mise au point des consignes** ; pour éviter une présentation verbale trop longue , on passe à une présentation figurale où beaucoup de choses restent implicites .

Trois élèves ont complété avec des longueurs fausses , un autre avec REcule 30 au lieu d'AVance . Cinq élèves ont ajouté des Tourne Droite ou Gauche inutiles et variés . Nous n'irons pas plus loin dans l'analyse de leurs erreurs dans cette épreuve écrite . Pour les interpréter , il nous faudrait savoir où l'élève situe sa tortue , lorsqu'il écrit une instruction . Il est très vraisemblable que le programme écrit n'a pas été simulé correctement .

La grande variété d'erreurs dans cette partie est à rapprocher de la grande variété d'erreurs dans le dessin 6 de la première partie et de la grande quantité d'erreurs simples "ES" de coordination dans l'exercice 2 SAPIN .

#### 4 - Les arbres

Dans les exercices 1 et 2 , nous avons demandé aux élèves d'écrire les arbres de programmation correspondants . Les arbres avaient été systématiquement utilisés par l'enseignante pour expliquer , décomposer ou corriger les problèmes .

Il est apparu que les élèves n'avaient pas pris conscience de toutes les conventions d'écriture dans un arbre , à savoir :

- de gauche à droite , dans l'ordre d'exécution ,
- de haut en bas , lorsque l'on décompose une procédure .

Dans l'exercice 1 , l'erreur la plus fréquente consiste à écrire A au-dessus de B pour exprimer que la procédure A doit être exécutée avant B . Il y a néanmoins neuf arbres respectant les conventions d'écriture , mais trois seulement correspondent au programme qui sera écrit ensuite . Dans le corrigé qui a eu lieu entre l'exercice 1 et l'exercice 2 ( cf. p.42 ) ,



nous avons rappelé les conventions d'écriture . Ce seul rappel a suffi à faire disparaître complètement le type d'erreur mentionné ci-dessus .

Dans l'exercice 2 , l'erreur la plus fréquente correspond à une difficulté qui n'existait pas dans l'exercice 1 : la **séquence** à **répéter** doit être **désignée** par un **nom de procédure** , cette procédure se décomposant au niveau suivant . Cette procédure intermédiaire n'a pas vraiment d'utilité dans le programme et cette erreur est vraiment mineure . Nous n'avons pas tenu compte de cette erreur faite par sept élèves , que nous avons classés dans le tableau ci-dessous avec ceux qui n'avaient commis aucune erreur . Parmi les arbres représentés sans aucune erreur , seuls trois correspondent au programme écrit ( dans les deux exercices , les trois arbres corrects ne correspondent pas aux mêmes élèves ).

**Critère ARBRE :**

" C " : l'arbre est correctement écrit .

" E " : l'arbre comporte des erreurs .

		exercice 2		
		C	E	
	C	5	4	9
exercice 1	E	8	2	10
		13	6	

### 5 . Conclusion par rapport à notre objectif , la programmation structurée :

#### **l' utilisation des procédures fournies**

L ' aptitude à utiliser des procédures fournies augmente d'un exercice à l'autre .

#### **la répétition**

La possibilité de répéter une liste d'instructions est connue des élèves , même si le maniement de l'instruction REPETE pose encore des problèmes .

#### **la coordination de procédures**

Si nous pouvons dire que la coordination des procédures est assez bien réussie , c'est en tenant pour mineures les **erreurs simples** . En effet seules les **erreurs de conception** sont significatives . Mais si l'on s'intéresse à ces erreurs simples de coordination , on constate qu'elles sont encore nombreuses et variées .

**la simulation de procédures** , qui est une activité différente de la programmation , reste difficile pour les élèves .

#### **les erreurs de pilotage**

Les élèves ont acquis une certaine maîtrise du pilotage de la tortue comme en témoignent les dessins 1 à 5 . La perte de contrôle survient plutôt lorsqu'il y a des changements de cap et évidemment surtout si la gauche de la tortue n'est plus la gauche de l'élève , comme en témoignent les nombreuses erreurs dans le dessin 6 . Dans ce domaine , des observations plus riches ont été réalisées avec des élèves plus jeunes , par exemple par P. Mendelsohn ( 1985 ) , J. Hillel ( 1985 ) , J. Hillel et R. Samurçay ( 1985 ) .

## Test 2 : POLYGONES

Un test individuel et court est donné sur les polygones réguliers quelconques , après la séance POLYS .

### 1. Consigne :

\*\*\*\*\*

---

Ecrivez individuellement , **sans utiliser votre cahier** , un programme permettant d'obtenir un polygone régulier quelconque .

\*\*\*\*\*

---

### 2. Remarque :

En temps normal , les élèves disposent toujours de leur cahier . Mais nous voulions savoir ce que les élèves pourraient réaliser dans ces conditions .

### 3. Analyse des réponses :

#### **1 - les programmes corrects**

Huit élèves ont écrit un programme correct , à la syntaxe du LOGO près . Ainsi , on trouve des écritures du type " *REPETE le nombre de côtés du polygone* " ; ceci n'est pas surprenant pour un programme papier-crayon qu'ils savent ne pas devoir exécuter .

**Une stratégie : utiliser beaucoup de variables pour ne pas expliciter la relation entre les variables .**

Une élève ( ALI ) a écrit un programme correct , mais qui n'exprime pas les relations entre les variables ANGLE et NC, le nombre de cotés :

POUR POLY : COTE : ANGLE : NC  
 REPETE : NC [ AV : COTE TD : ANGLE ]  
FIN

Nous signalons ici cette stratégie , qui consiste à **choisir autant de variables d'entrée** dans la procédure qu'il y a d'éléments variant dans le problème sans exprimer de **relation** entre ces éléments : pour que l'exécution de la procédure ci - dessus réalise un polygone régulier , il faudra que les valeurs données aux variables respectent la relation :

$$NC * ANGLE = 360$$

Ce comportement n'est pas un fait isolé mais bien **une stratégie de programmation** qui réapparaîtra plus tard , chez cette élève et d'autres . Nous développerons ce point dans la brochure suivante ( C. Dupuis , M.A. Egret , D. Guin , à paraître ) .

## 2 - les erreurs

Deux autres élèves évitent d'exprimer la relation en termes de variables . KAR écrit trois programmes corrects pour des polygones à quatre , six et douze côtés ; il utilise donc la relation sans l'expliciter . STÉ , par contre , laisse le nombre de côtés variable mais donne à l'angle la valeur magique  $90^\circ$  ! Ces deux élèves utilisent une variable pour la longueur du coté .

### les problèmes de désignation :

Sept élèves ont désigné par le même nom des quantités différentes : c'est soit le nombre de côtés et la mesure de l'angle , soit le nombre de côtés et la longueur du côté .

#### a ) le nombre de côtés et la mesure de l' angle

Trois élèves ont écrit un programme où **le même nom de variable** désigne le nombre de côtés et la mesure de l'angle , ce qui donne par exemple :

POUR POLYGONE : COTE : N  
 REPETE : N [ AV : COTE TD : N ]  
FIN

Une grande partie du travail en classe avait porté précisément sur l'expression de la **relation** entre le nombre de côtés et la mesure de l'angle  $NC * ANGLE = 360$  et son utilisation dans l'écriture des programmes . Les deux programmes proposés à la correction expriment une de ces deux quantités en fonction de l'autre . Une seule variable est donc utilisée pour l'expression de ces deux quantités . On peut penser que ces élèves ont voulu exprimer l'**existence d'une relation** en désignant par le même nom les deux quantités en relation .

#### b) le nombre de côtés et la longueur du côté

Quatre élèves désignent , par **le même nom de variable** , le nombre de côtés et la longueur du côté , en fait tout ce qui se rapporte au côté ! Ce qui donne , par exemple :

POUR POLYGONE : COTE

REPETE : COTE [ AV : COTE TD 360 / : COTE ]

FIN

Le nom de la variable est parfois N , parfois COTE , comme dans l'exemple ci-dessus . Les problèmes de désignation d'objets par des noms de variables ne sont pas propres au LOGO et s'observent en mathématique au même niveau scolaire dans les problèmes de mise en équations .

Nous reviendrons dans la brochure suivante sur les problèmes de **désignation** et sur les **stratégies** développées par les élèves pour les résoudre .

### Test 3 : BALLONS

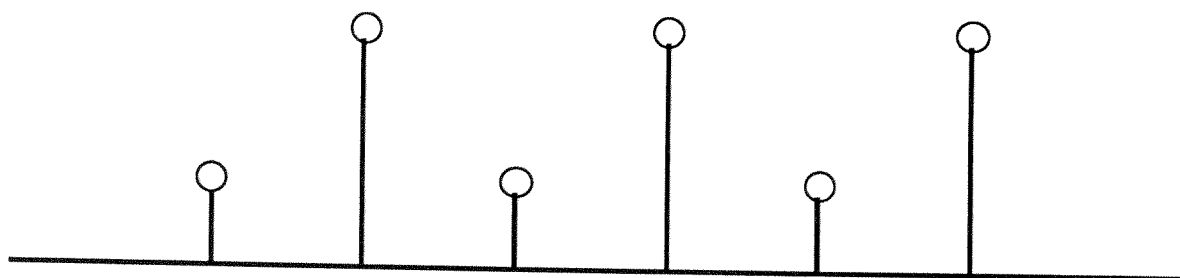
#### 1. Consignes :

\*\*\*\*\*

---

Répondez individuellement sur le cahier au test proposé .

Dessinez l'ARBRE et écrivez 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 .

Tapez ensuite le programme " BALLON " .

\*\*\*\*\*

---

## 2. Présentation du test :

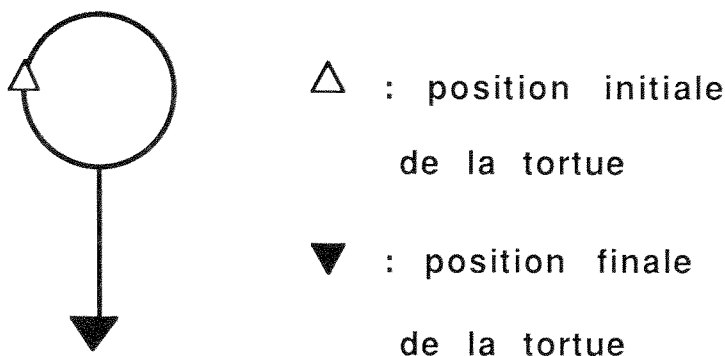
Ce test porte essentiellement sur un point : la structuration des programmes . Les élèves peuvent, au fur et à mesure des places vacantes sur micro-ordinateur , tester leur programme et le corriger .

Le dessin s'inspire d'une activité proposée et observée par J. Hillel et R. Samurçay (1985 ) . Mais si les dessins sont semblables ( nous avons rajouté le sixième ballon pour faciliter l'utilisation d'un REPETE 3 fois ), les situations sont différentes .

### Situation de J. Hillel et R. Samurçay :

Les deux élèves ont neuf - dix ans . Ils ont auparavant , travaillé douze heures dans un environnement LOGO particulier . Ils sont observés une heure par semaine pendant dix semaines . L'activité se place au cours de la quatrième séance . Ils disposent d'une procédure LOLLY paramétrée . Le paramètre désigne la longueur de la tige .

Cette procédure LOLLY n'a pas été conçue pour être utilisée dans ce contexte . Elle répond à une tâche particulière : le dessin d'un ballon isolé . Sa construction commence par le cercle et finit par la tige .



Ce qui était une solution raisonnable dans un autre contexte devient une **source de difficultés** dans cette nouvelle situation . La réutilisation de LOLLY pose aux élèves d'énormes problèmes d' **interface** . Les auteurs font remarquer qu'en général des élèves ont tendance à ne pas modifier une procédure " qui marche " : si cette situation est intéressante pour des chercheurs , elle n'est pas optimale pour l'enseignement .

Notre situation :**A priori :**

La figure est présentée sans indication de décomposition . Les élèves disposent de deux procédures pour tracer des cercles ( considérés comme des polygones réguliers ) :

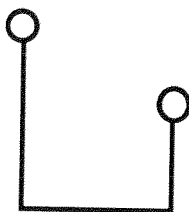
POLYGONE : longueur du coté : nombre de cotés et

POLYGONE : longueur du coté : angle .

Ces procédures sont écrites dans leurs cahiers . En effet , nous avons travaillé , lors d'une précédente séance , à l'écriture de procédures permettant de tracer un polygone régulier quelconque , puis un cercle ( polys , p .20 ) . Les élèves ont exécuté ces procédures , par contre ils n'ont pas écrit de procédure BALLON dans un autre contexte que celui de ce test .

On peut raisonnablement penser que **le contexte va influencer le choix des positions initiale et finale** dans l'écriture des procédures ( ou des listes d'instructions ) BALLON .

Le fait que la " plage " , comme certains ont appelé le grand trait horizontal , soit tracée au-delà des pieds des tiges renforce l'impression visuelle que les cercles sont " au bout des tiges " et ne sont pas le point de départ d'un tracé , comme on pourrait le concevoir dans ce dessin :





### A posteriori :

Tous les élèves ont utilisé une des procédures POLYGONE pour tracer les cercles . POLYGONE a , dans notre situation , un rôle analogue à LOLLY chez J.Hillel et R. Samurçay ; c'est une procédure qui a été conçue , écrite , et exécutée pour répondre à une tâche particulière : le dessin d'un polygone , puis d'un cercle isolé .

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 .

- 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 celui qui se pose entre deux procédures qui ont été conçues et écrites pour être coordonnées . C'est pourquoi nous analyserons , dans un critère particulier , la position du cercle .

L'**interface** à écrire pour placer les cercles correctement est assez courte et la réutilisation de la procédure POLYGONE n'a donc pas posé autant de problèmes que celle de LOLLY .

Nous n'avons observé aucune procédure ( ou liste d'instructions ) commençant par le cercle comme LOLLY , ce qui s'explique facilement par le fait que , dans ce contexte , LOLLY ne constitue pas une solution économique .

### 3. Analyse des réponses :

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

#### Critères d'analyse des réponses :

##### CERCLE :

- " 0 " - Aucun cercle n'est tracé .
- " C " - Les cercles sont correctement placés aux bouts des tiges .
- " T " - Les cercles sont tangents aux tiges .
- " A " - Les cercles ne sont ni tangents , ni corrects .

La modalité " Autre " a été observée chez une élève ( SAB ) qui a mis un angle de  $45^\circ$  entre la tige et la tangente au cercle !

Le problème des cercles étant traité à part , tous les autres critères sont observés en dehors de la procédure éventuellement utilisée pour tracer les cercles .

Parmi les critères définis précédemment , nous réutiliserons ici les deux critères :

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

COORDINATION de procédures .

( cf. analyse du Test 1 p. 45 et 49 )

SP : Utilisation d'AU MOINS UNE SOUS - PROCEDURE .

" 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 celle de la " tige " , il est possible de **réutiliser** la même liste d'instructions pour les deux sortes de ballons .

**ETAT : PRISE EN COMPTE DE L' ETAT DE LA TORTUE A LA FIN** d'une procédure graphique (ou d'une liste d'instructions ) destinée à être intégrée à un projet .

" **PS** " - Il y a un **RETOUR A UNE POSITION " STANDARD "** à la fin de la procédure . 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 .

**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 .**

" **INT** " - L' **INTERFACE EST INTEGREE** à la procédure . 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 coordination ne nécessite aucune interface , ou quasiment aucune .

**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 le problème du moment . 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 .

" **SANS** " - AUCUN RETOUR A 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 .

" **IMPL** " - POSITION " STANDARD " IMPLICITE . Ceci peut être observé dans des situations où un tracé amène **automatiquement** , à la fin , la tortue dans une position " standard " .

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 .

Les observations des critères sont présentées dans le tableau :

CRITERES :	CERcle	S	P	VAR	ETAT	COor -dination	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 .

Sur les **quatre premières lignes** sont regroupés les huit élèves pour lesquels la **coordination** des procédures est **correcte**, l'instruction REPETE est bien utilisée et les cercles sont bien placés. Il est à noter que l'on trouve parmi eux **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 ont utilisé au moins une sous - procédure.

**Les huit premières lignes** : si l'on ne tient pas compte de la position du cercle par rapport à la tige, 12 élèves ont écrit un programme " Correct ", c'est à dire qui fournit le dessin attendu.

### **REPETE :**

" **00** " : CYR n'utilise aucune instruction REPETE, il utilise trois fois les mêmes procédures pour écrire son programme principal. L'exécution de son programme produit le dessin attendu.

" **XF** " : La structure du programme de GIL est :

#### POUR GIL

*liste d'instructions avec une erreur simple*

REPETE 2 [ GIL ]

FIN

Nous avons interrogé GIL : il n'a pas découvert la récursivité terminale ; son programme s'arrête " hors limite " et il n'a rien observé de particulier. Pour lui, la liste d'instructions est désignée par GIL et après l'avoir exécutée une fois, il faut la répéter encore deux fois pour obtenir le dessin complet ( Dans le TEST n° 1, nous avons déjà rencontré une écriture analogue de la répétition chez deux élèves ; GIL était l'un d'eux ).

" **FF** " : REG voit la figure comme six répétitions d'un élément de taille variable. Il écrit donc une procédure paramétrée. Mais il fait une erreur simple de coordination et utilise **le même nom** COTE pour désigner **deux longueurs** : celle de la tige et celle du côté du polygone régulier faisant office de cercle. Nous avons déjà observé ce problème de **désignation** dans l'épreuve POLYGONE ( cf. p. 56 ). Cette accumulation d'erreurs produit un dessin très difficile à analyser, et REG ne parviendra pas à corriger son programme.

### Réussite et Sous - Procédures :

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 sous - procédures est **liée à la réussite** d'ensemble puisque sur douze élèves qui ont réussi la coordination , onze d'entre eux ont utilisé des sous - procédures .

### Les arbres de programmation :

Dix élèves n'ont pas écrit d'arbre , ce qui correspond donc au comportement le plus fréquent . La situation diffère du test 1 en ce que :

- les élèves sont moins débutants .
- ils savent qu'ils auront à exécuter leur programme , ce qui les conduit à privilégier le travail sur le programme par rapport à un travail sur l'arbre du programme .

Seuls deux des arbres écrits sont corrects ; ils correspondent au programme qui est écrit .

#### 4. Conclusion :

##### **la situation**

La situation n'était pas suffisamment complexe pour qu'il soit **absolument nécessaire** pour réussir d'utiliser des **variables** , des **sous - procédures** ou même l'instruction REPETE.

##### **l'utilisation des régularités**

Malgré les échecs , on peut affirmer que tous les élèves ont perçu et tenté d'exploiter des **régularités** de la figure . C'est pourquoi nous n'avons pas fait figurer dans le tableau un critère d'utilisation des régularités , il n'aurait pris qu'une seule valeur pour tous les élèves : OUI .

##### **la décomposition**

La décomposition de la figure est aussi évidente pour tous , car ceux qui n'ont pas éprouvé le besoin d'écrire formellement des sous - procédures ont tous utilisé l'instruction REPETE [ une liste d'instructions ] . De plus , parmi les douze élèves qui ont réussi la coordination , onze d'entre eux ont utilisé des sous - procédures .

##### **la répétition**

La répétition d'éléments dans la figure est évidente pour tous , même lorsque l'instruction correspondante n'est pas utilisée . Sur les dix-huit élèves qui ont écrit un programme , dix-sept ont utilisé l'instruction REPETE . Le seul qui se singularise , et qui a réussi , a décomposé en procédures et répété les noms de ses procédures trois fois .

##### **la prise en compte de l'état de la tortue**

Tous les élèves ont pris en compte l'état de la tortue à la fin de la procédure ou de la liste d'instructions à répéter . Il n'y a dans cette épreuve aucune erreur de coordination que l'on puisse qualifier d'erreur de conception " EC " .



### **et les spaghettis ?**

La grande variété des erreurs ne doit pas masquer le fait que nous n'avons plus observé, à ce stade, de tentative d'écriture d'un programme " spaghetti " où les instructions résultent d'un simple pilotage à vue de la tortue du début à la fin du tracé .

### **les arbres de programmation**

Les arbres de programmation ont tendance à disparaître lorsque la compétence des élèves augmente par rapport à la tâche demandée . On peut penser que les élèves n'en ressentent plus le besoin . Nous pensons néanmoins que les arbres de programmation aident les élèves à concevoir la décomposition des programmes .

**On peut donc affirmer que les éléments de base d'une programmation graphique structurée sont en place chez ces élèves .** La structuration du programme est devenue une méthode efficace de résolution de problème .

Bien entendu , tout n'est pas définitivement acquis . Il faut tempérer notre enthousiasme par le fait que cette épreuve est relativement simple pour eux . Ainsi , nous verrons réapparaître plus tard , dans un contexte plus difficile , certaines erreurs dues à une **conception " image "** de la procédure . Tout se passe comme si cette conception première réapparaissait quand la situation est plus complexe , au moins chez certains . Mais ceci est une autre histoire , que nous vous conterons dans un autre document ...

## Conclusion

Rappelons nos objectifs :

A) **La construction de situations** destinées à produire chez l'élève l'**acquisition des connaissances significatives** dans le domaine de **la programmation structurée graphique** .

B) **L'analyse des procédures de résolution** , des difficultés des élèves et des conceptions mises en oeuvre dans ces situations .

Ces deux points sont indissociables : il est , en effet , inconcevable de construire sans analyser ou d'analyser sans construire .

A) **Construction de situations** :

A la suite d'observations réalisées durant la préexpérimentation , dans les premiers projets graphiques , nous indiquons explicitement les **positions initiale et finale** de la tortue. Préciser les projets , afin que les **positions initiale et finale** de la tortue soient claires , permet l'**indépendance** des procédures et une **coordination plus simple** . Cette indication peut disparaître lorsque les élèves ont pris conscience de la nécessité de la **prise en compte de l'état de la tortue** au début et à la fin d'un tracé , en particulier si celui-ci correspond à une procédure destinée à être **intégrée** à une autre procédure . Dans le choix des activités proposées aux élèves , nous avons veillé à mettre en échec assez rapidement une **conception " image "** de la procédure ( cf. p. 8 ) .

Nous recherchions des situations où la **structuration** du programme soit ressentie non comme une contrainte supplémentaire imposée par l'enseignant , mais comme **une aide effective** à la réalisation du projet . Beaucoup d'élèves débutants n'hésitent pas à aligner une page entière d'instructions : " TD 90 AV 45 etc... " , ce que nous appellerons écrire un programme " spaghetti " . Ce type de programme disparaît lorsque les situations deviennent plus complexes , si la possibilité de **structurer** est présentée aux élèves et admise par eux comme une technique plus économique . Ceci nécessite un certain nombre d'heures de programmation et un **choix de situations** rendant difficile la programmation " spaghetti " :

- dans un premier temps , nous avons construit des situations en **explicitant des invariants** ( cf. p. 13 ) . Le jeu du TANGRAM est par nature " structuré " . On enlève ainsi les difficultés liées au **choix de la décomposition** . Il n'y a plus que des **problèmes de coordination** entre les différentes procédures . La coordination nécessite l'utilisation d'**interfaces** ( cf. p.10 ) en dehors des procédures .

- dans un deuxième temps , nous avons choisi des projets où l'explicitation d'invariants constitue une **démarche efficace de résolution du problème** . Les projets graphiques doivent permettre d'exploiter **les régularités** d'une figure pour inciter **naturellement** à l'utilisation de l'instruction **REPETE** . La décomposition d'une figure devrait devenir évidente pour tous et les projets assez complexes pour que la **décomposition** du problème en sous - problèmes soit **liée à la réussite** ( cf. p. 66 ) : l'élève doit sentir la **nécessité** d'utiliser l'instruction **REPETE** et des **procédures** .

- dans un troisième temps , les situations choisies doivent être suffisamment complexes pour qu'il soit **plus économique** d'utiliser des **variables** ( cf . carrés p.23 , polygones p.20 ) .

Divers types d'épreuve ont été utilisés en tenant compte des difficultés spécifiques repérées et des conceptions spontanées identifiées :

- **Ecriture de programmes avec ou sans ordinateur pour les tester** ( cf . Test 2 p.54 , Test 3 p. 57 ) .
- **Compréhension de programmes** ( cf . Test 1 p. 33 ) .
- **Coordination de procédures** ( cf . Test 1 p. 37 ) .
- **Complétion de programmes** ( cf . Test 1 p. 40 ) .

Si nous avons choisi ces types de questions , c'est pour **mettre en évidence** les **difficultés** des élèves . Elles sont plus difficiles à distinguer lorsqu'elles se mêlent dans une tâche d'écriture de programme et souvent inobservables lorsque l'élève renonce ou échoue .

## B) Analyse des procédures de résolution :

### 1. Difficultés spécifiques repérées :

#### a) le pilotage de la tortue

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

#### b) la structuration et la décomposition en procédures

La programmation structurée avec sous-procédures représente une réelle difficulté pour les programmeurs débutants . Pour certains , 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 structurer . **Les procédures** , si elles sont fournies par l'enseignant , facilitent la décomposition mais leur utilisation présente encore des difficultés : les problèmes de la prise en compte de l'état de la tortue et de la coordination ne sont pas résolus .

#### c) la prise en compte de l'état de la tortue ( cf. p. 62 )

Dans les projets graphiques , l'absolue nécessité de connaître les **positions initiale et finale** de la tortue pour pouvoir coordonner les procédures est longue à acquérir .

**d) la coordination de procédures graphiques** est une activité dans laquelle se révèlent toutes les difficultés signalées ci-dessus . Nous avons distingué deux types d'erreurs de coordination : les **erreurs simples** d'interface et les **erreurs de conception** ( cf. p.45 ) .

#### e) l'utilisation de variables

On constate que , pour certains , la difficulté est telle qu'ils évitent d'en utiliser tant que la situation ne les y **contraint** pas . Mais lorsque la situation amène les élèves à définir des variables , on voit apparaître ( cf. p .54 ) :

- **des problèmes de désignation** : le même nom de variable désigne deux éléments différents , par exemple deux longueurs ou deux caractéristiques d'un même objet . Les problèmes de désignation d'objets par des noms de variables ne sont pas propres au LOGO et s'observent en mathématique au même niveau scolaire dans les problèmes de mise en équations .

- **des problèmes d'expression d'une relation entre des variables**

Les programmes de certains élèves font apparaître **une stratégie** de contournement de la difficulté : définir et utiliser **autant de variables d'entrée** dans la procédure qu'il y a d'éléments variant dans le problème , de façon à **éviter d'explicitier les relations entre ces éléments** . L'explicitation de relations entre des variables s'apparente à l'écriture d'équations , qui est une tâche difficile à ce niveau scolaire au moins . Pour ces élèves , la désignation n'est plus un problème . Les relations qui existent entre les variables sont respectées lors de l'affectation de valeurs pour l'exécution .

Ce comportement n'est pas un fait isolé mais bien **une stratégie de programmation** qui réapparaîtra plus tard . Nous développerons ce point dans la brochure suivante ( C. Dupuis , M.A. Egret , D. Guin , à paraître ).

## 2. Conceptions spontanées identifiées :

a) Certains élèves ne s'éloignent que progressivement d'une conception " image " de la procédure : nous désignons par **conception " image "** une **identification** de la **procédure** avec le tracé souhaité ou la **figure** obtenue . Les premiers programmes des élèves reflètent tous cette conception qui est un obstacle à la programmation en Logo .

b) Au lieu d'utiliser **REPETE** n fois une liste d'instructions , les élèves écrivent une première fois la liste puis **REPETE** n-1 fois ; ceci correspond au **modèle spontané de la répétition** déjà mis en évidence : **faire une action et recommencer** ( cf. p.8 ).

### 3. Critères d'analyse

Les critères d'analyse des réponses sont résumés dans l'annexe 3 ( cf. p .83 ).

A l'issue de cette première phase d'expérimentation , on peut 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 chez ces élèves . La structuration du programme est devenue une méthode efficace de **résolution de problème** .

## ANNEXE 1 : Trois fiches de cours

### Généralisons nos procédures !

Pour généraliser une procédure , on peut y introduire des **variables** :

*Exemple :*

<p><u>POUR TRIANGLE</u> : COTE          REPETE 3 [ AV : COTE TD 120 ]  <u>FIN</u></p>
---

COTE : **Nom** de la variable ( On peut considérer que c'est le nom d'une boîte ) . Pour faire exécuter cette procédure , il suffit de taper :

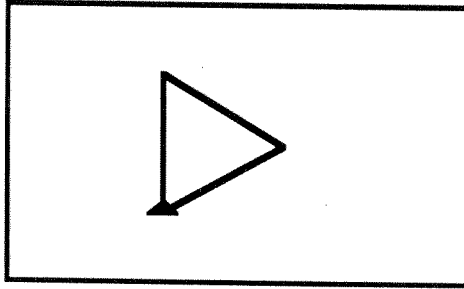
**TRIANGLE 20**

Cet ordre a pour effet :

1 - De placer la valeur 20 dans la boîte de nom COTE . Le **contenu** de la boîte de nom COTE est alors 20 . Le contenu de la boîte ( ou **valeur** de la variable ) de nom COTE est noté **:COTE** .

	Boîte
Nom	COTE
Contenu	: COTE ou 20

2 - D' exécuter la procédure en remplaçant : COTE par 20 . Ce qui donne :



▲ : positions initiale  
et finale de la tortue

**Attention :** Si la boîte contient déjà une valeur au moment de l'appel de la procédure , cette valeur est **remplacée** par la valeur donnée dans l'appel ( ici 20 ) durant l'exécution de la procédure . A la fin de la procédure , la boîte récupère l'ancien contenu .

Exemple :

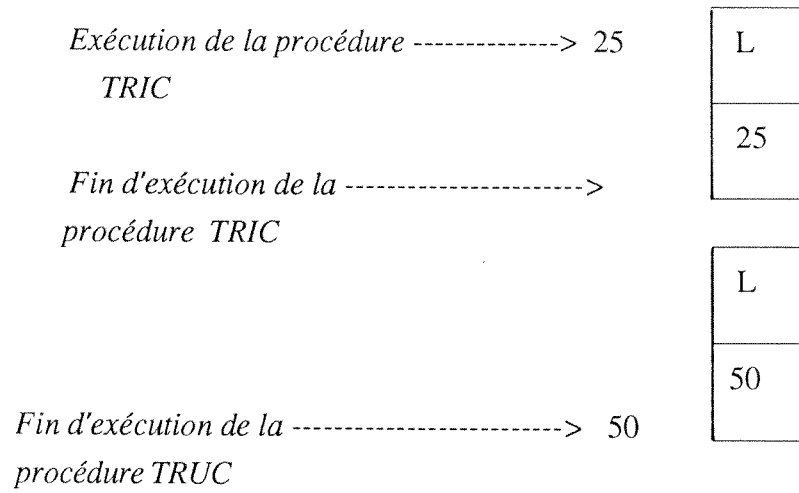
<u>POUR TRUC : L</u>	
ECRIS : L	<u>POUR TRIC : L</u>
TRIC 25	ECRIS : L
ECRIS : L	<u>FIN</u>
<u>FIN</u>	

Appel de procédure -----> TRUC 50

Exécution de la procédure -----> 50  
TRUC

L
50





**Remarque:** Si la boîte ne contient aucune valeur avant l'exécution de la procédure , elle n'en contient aucune après l'exécution .

## Mots et listes

### 1. Objets LOGO :

En LOGO , il existe deux types d'objets :

- **le mot** ( On le fait **précéder de guillemets** pour que l'ordinateur reconnaisse que c'est un mot )

*Exemples :* "JEAN , "PIROUETTE , "CHOU.FLEUR

**Remarque :** Un **nombre** est un mot LOGO , mais il n'est pas nécessaire de mettre les guillemets .

- **la liste** ( Que l'on met entre crochets )

*Exemples :* [ LE PARAPLUIE ] , [ IL PLEUT BEAUCOUP ]

### 2. Instruction de sortie : ECRIS

*Exemple :* ECRIS "JEAN ECRIS [ LE PARAPLUIE ]

*Exécution de l'instruction :* -----> JEAN  
LE PARAPLUIE

### 3. Former d'autres mots et listes : MOT , PHRASE

*Exemples :* ECRIS MOT "BON "JOUR  
ECRIS PHRASE [ IL ] [ PLEUT ]

*Exécution des instructions* : - - - - -> BONJOUR  
IL PLEUT

#### 4. Manipuler listes et mots : SP , SD , PREM , DER , ITEM

*Exemples* : ECRIS SP "TROIS ECRIS SP [ IL PLEUT BEAUCOUP ]

*Exécution des instructions* : - - - -> ROIS  
PLEUT BEAUCOUP

*Exemples* : ECRIS SD "TROIS ECRIS SD [ IL PLEUT BEAUCOUP ]

*Exécution des instructions* : - - - -> TROI  
IL PLEUT

*Exemples* : ECRIS PREM "TROIS ECRIS PREM [ IL PLEUT BEAUCOUP ]

*Exécution des instructions* : - - - -> T  
IL

*Exemples* : ECRIS DER "TROIS ECRIS DER [ IL PLEUT BEAUCOUP ]

*Exécution des instructions* : - - - -> S  
BEAUCOUP

*Exemple* : ECRIS ITEM 3 [ TROIS IL PLEUT BEAUCOUP ]

*Exécution de l'instruction* : - - - -> PLEUT

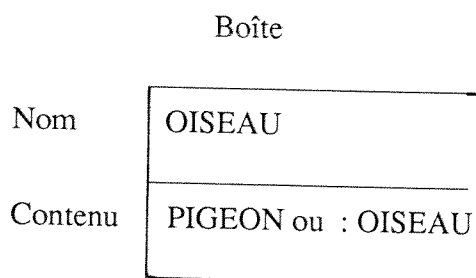
## Instruction RELIE

1. Un **mot LOGO** peut s'utiliser comme **variable** :

*Exemple :* RELIE "OISEAU "PIGEON

Cet ordre a pour effet :

- de **créer une boîte** de nom OISEAU .
- de **placer la valeur** PIGEON dans la boîte de nom OISEAU .



**Remarque :** La valeur de la variable peut être un mot ou une liste . Ce n'est pas forcément un nombre .

2. Pour **connaître le contenu** de la boîte de nom OISEAU , on tape :

ECRIS : OISEAU

*Exécution de l'instruction:* -----> PIGEON

**Remarque :** L'instruction ECRIS "OISEAU a pour effet d'afficher à l'écran OISEAU .

*Exemple :* RELIE "TROIS 3  
ECRIS : TROIS

*Exécution des instructions :* -----> 3

3. Pour **modifier le contenu de la boîte**, il suffit d'utiliser l'instruction RELIE :

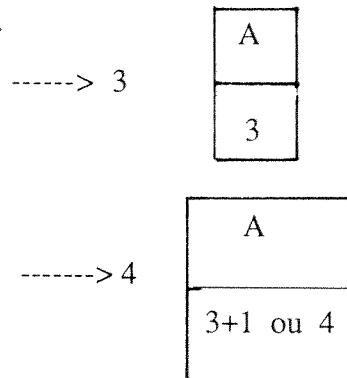
*Exemple :* RELIE "OISEAU "MOINEAU  
 ECRIS : OISEAU  
 RELIE "OISEAU [ PERRUCHE PERROQUET ]  
 ECRIS : OISEAU

*Exécution des instructions :* -----> MOINEAU  
 -----> PERRUCHE PERROQUET

*Exemple :*

RELIE "A 3  
 ECRIS : A  
 RELIE "A : A +1  
 ECRIS : A

*Exécution des instructions :*



## ANNEXE 2 : Planning

La durée des séances décrites dans le tableau ci-dessous est de **deux heures** . Seule la séance 7 a duré **une heure** .

Séance	Travail proposé
1	fleur ( p 7 )
2	moulin ( p 11 ) ( une heure ) tangram ( p 13 ) : recherche des questions 1 , 2 , 3 , et 4
3	tangram : suite test 1 : Compréhension de programmes ( p 33 ) test 1 : TIGES ( p 37 ) ( une heure )
4	Corrigé du test 1 ( p 42 ) . ( une heure ) tangram : fin
5	test 1: SAPINS et Complétion de programmes ( p 38 ) ( une heure ) polys ( p 20 ) : recherche des exercices
6	polys : mise au point ( p 21 ) ( une heure ) carrés ( p 23 ) : étapes 1 et 2

- 7 carrés ( p 24 ) : début de l'étape 3
- 8 test 2 : polygones ( p 54 ) ( vingt minutes )  
carrés : étape 3 ( p 24 )
- 9 Mots et listes ( p 28 )  
test 3 ( p 57 ) : ballons , début de la recherche  
( une heure )
- 10 test 3 : fin de la recherche ( une heure )  
  
Relie ( p 29 ) : recherche des exercices sans  
micro-ordinateur
- 11 Correction du test 3 ( trente minutes )  
Relie : exercices 1 , 2 et 3

## ANNEXE 3 : Rappel des critères d'analyse

Pour analyser les productions des élèves , nous avons défini des critères d'analyse que nous résumons ici .

**ETAT : PRISE EN COMPTE DE L'ETAT DE LA TORTUE A LA FIN** d'une procédure graphique (ou d'une liste d'instructions ) destinée à être intégrée à un projet ( cf. p.8 , 62 ) .

" **PS** " : Il y a un **RETOUR A UNE POSITION " STANDARD "** à la fin de la procédure . Une position "**standard**" est une position à partir de laquelle la **coordination** de cette procédure avec les autres est **économique** .

" **INT** " : **L'INTERFACE EST INTEGREE** à la procédure .

" **SANS** " : **AUCUN RETOUR A UNE POSITION " STANDARD "** ne figure dans la procédure .

" **IMPL** " : **POSITION " STANDARD " IMPLICITE** .

**COORDINATION des procédures :**  
( cf. p. 45 )

" **0** " : la coordination des procédures **N'EST PAS REALISEE** .

" **C** " : le programme est **CORRECT** .

Nous distinguons 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 ( cf. page 8 ) . Cette conception ne peut se repérer uniquement dans l'écriture des procédures . Elle ne peut être observée qu'au moment où l'élève veut coordonner ses procédures : il n'écrit aucune interface pour les coordonner ou écrit une interface supposant un retour à une position " standard" alors que :

- d'une part , il n'a pas fait de retour à une position " standard " .
- d'autre part , il n'a pas intégré l'interface aux procédures .

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

( cf. p. 49 )

**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 .

**\$P : Utilisation d' AU MOINS UNE SOUS - PROCEDURE :**

" **OUI** " / " **NON** " ( cf. p. 61 ) .

**VAR** : Utilisation d' **AU MOINS UNE VARIABLE** pour désigner une longueur :  
" **OUI** " / " **NON** " ( cf. p. 62 ).

**OUTIL** ou modalité d'écriture du programme dans le cas où des procédures sont fournies ( cf. p. 44 ) :

- " **Tout** " : Utilisation de **toutes** les procédures fournies .
- " **Part** " : Utilisation de certaines des procédures fournies .
- " **Autr** " : Ecriture d' **autres** procédures .
- " **Sans** " : Ecriture du programme **sans** aucune sous-procédure .

**CERCLE** ( cf. p. 61 ) .

## Bibliographie

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

DUPUIS C. , EGRET M. - A. , GUIN D. ( à paraître ) : Récurtivité et Logo - 2 : Présentation et analyse de situations , Brochure I.R.E.M de Strasbourg .

GUIN D. , HELM J.- G.( 1984 ) : LOGO - 2 : Rapport d'expérimentation en CM<sub>2</sub> , 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 .

HILLEL J. ( 1985 ) : Mathematical and programming concepts acquired by children , aged 8 - 9 , in a restricted logo environment , Recherches en Didactique des Mathématiques , Vol . 6 , n°2 . 3 , pp. 215 - 268 .

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 .

ROUCHIER A., SAMURCAY R. (1987 ) : Didactique de l'informatique , Actes du colloque du GRECO " Didactique et Acquisition des Connaissances Scientifiques" du CNRS.

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

SAMURCAY R. , ROUCHIER A. ( 1987 ) : Acquisition de la notion de récursivité comme modèle de programmation itérative en Logo , Colloque de la Société Française de Psychologie - Les apprentissages : les perspectives actuelles .

## Table des Matières :

	page
Avant - propos	1
Introduction	1
Conditions de l'expérimentation	5
Présentation des différentes séances	6
Séance 1 : Fleur	7
Séance 2 : Moulin	11
Séance 3 : Tangram	13
Séance 4 : Polys	20
Séance 5 : Carrés	23
Séance 6 : Mots et listes	28
Séance 7 : Relie	29
Présentation des différents tests	
Test 1 : Compréhension de programmes	33
Test 1 : Coordination de procédures	37
Test 1 : Complétion de programmes	40
Test 2 : Polygones	54
Test 3 : Ballons	57
Conclusion	69
Annexe 1 , trois fiches de cours :	
Généralisons nos procédures	74
Mots et listes	77
Instruction Relie	79
Annexe 2 : Planning	81
Annexe 3 : Rappel des critères d'analyse	83
Bibliographie	86