

«SCHEMES INFORMATIQUES PROGRAMMABLES» :

Utiliser l'environnement LOGO pour construire des situations
d'apprentissage de tracés graphiques

Patrick MENDELSON
Chargé de recherche C.N.R.S.
La boratoire de psychologie expérimentale
Universite II - Grenoble

I - PRESENTATION DU PROJET.

1. programmation et organisation des connaissances chez l'enfant.

Un langage de programmation est un ensemble d'instructions qui, une fois interprétées par un programme conçu à cet effet, sont exécutées par le processeur d'un ordinateur. Ces «primitives» sont des «formes élémentaires» imposées par le concepteur du langage et choisies en fonction de leur pertinence pour résoudre une classe de problèmes assez large. Il en résulte un certain «formalisme» qui rebute souvent le débutant qui ne sait pas pourquoi certains choix de fonctions ou d'opérations ont été faits car il n'a pas, au départ, d'idée précise sur les objets qu'il peut construire avec un tel ensemble d'instruments.

Le niveau d'entrée dans un langage de commande est relatif :

- Plus le système est évolué, plus les objets que l'on manipule ont une fonction et un usage spécifique (dessin, musique, traitement de texte...). On peut trouver une analogie de ces systèmes dans la langue naturelle avec le jargon des spécialistes : les concepts de «cognition», «d'intégration sensorielle» ou de «structure opératoire»... constituent, par exemple, un ensemble de termes tout à fait pertinent et utile pour un psychologue cognitiviste mais leur spécificité les rend pratiquement inutilisables dans un autre contexte.

- Plus le système est proche de la logique de la machine, plus les objets sont élémentaires et peu spécifiés. Pour reprendre notre analogie avec la langue naturelle, des mots simples comme «objet», «forme», «sujet»... permettent de communiquer dans des contextes très variés, mais l'expression de concepts évolués avec de tels termes suppose l'utilisation de périphrases qui peuvent être assez longues et rendre du même coup le discours incompréhensible.

L'entrée dans un système informatique est fonction des objectifs que l'on veut atteindre, elle est donc arbitraire. Généralement, quand on manipule un logiciel, le niveau 0 est constitué par les commandes disponibles qui sont recensées dans le manuel accompagnant le produit. Il n'est pas possible, en général, de descendre en dessous de ce niveau pour modifier les commandes. La seule activité autorisée consiste à combiner des séquences de commandes pour obtenir un effet particulier plus ou moins planifié suivant la maîtrise du sujet. Pour un langage de programmation, le niveau 0 c'est l'ensemble des instructions répertoriées dans le manuel de référence. Le système représenté par le langage et le processeur est un système ouvert car il permet de traiter une infinité de problèmes élémentaires. En contrepartie sa maîtrise est plus complexe car le sujet apprenant doit se faire une idée des sous-objets qui composent l'objet-but en termes assimilables par la machine. La fonction des manuels d'apprentissage consiste en général à parcourir l'ensemble des «objets-prototypes» compatibles avec le système. Avec l'expérience ces «objets-prototypes» (par exemple POLYGONE ou ARBRE ou encore TRI en LOGO) prennent le statut d'objets élémentaires pour le sujet et il dispose ainsi virtuellement d'un système plus spécialisé. L'«expert» peut cependant à tout moment redescendre d'un niveau et modifier l'écriture des objets-prototypes pour résoudre des problèmes particuliers. Par la suite, nous utiliserons le terme «expertise» pour désigner un ensemble d'acquisition cohérent qui permet de résoudre, avec une certaine habileté, une classe de problèmes bien définie.

L'acquisition d'une expertise par l'apprenti-programmeur est déterminée par deux facteurs : d'une part, par sa capacité de mise en relation et de coordination d'unités de connaissances déclaratives et/ou procédurales qu'il peut mettre en œuvre et, d'autre part, par l'expérience que le sujet acquiert d'un domaine de connaissance par la qualité et la variété des formes qu'il peut manipuler. Ces deux facteurs sont en interaction et ils sont indissociables. Notre projet est d'offrir des situations d'apprentissage qui permettent d'expérimenter ce modèle en contrôlant chaque phase du processus, depuis le choix des formes élémentaires jusqu'aux modalités de coordination que l'enfant met en œuvre.

2. A l'origine du projet, un constat.

Le projet S.I.P. est né du constat des difficultés rencontrées par les pédagogues et les chercheurs dans leurs expériences sur l'acquisition des concepts informatiques par des débutants. Ces problèmes comprennent entre autres les points suivants :

- Programmation éducative VS programmation professionnelle.

La programmation éducative est un domaine d'apprentissage qui se démarque de la programmation professionnelle par ses objectifs, par les caractéristiques du milieu dans lequel elle se développe, par les systèmes dont disposent les établissements et par la méthode utilisée pour son apprentissage. Toute proportion gardée, la distance entre les concepts informatiques utilisés par des débutants dans une formation à vocation non professionnelle et les concepts informatiques du professionnel est la même qu'entre les concepts mathématiques enseignés à l'école primaire (le nombre, la structure additive...) et les concepts mathématiques du chercheur dans la discipline.

- Les langages informatiques pour l'enseignement.

Les principaux systèmes disponibles sont conçus pour aborder des classes de problèmes très hétérogènes : graphisme, traitement de chaînes ou de listes, composition musicale... Le débutant a toujours beaucoup de difficultés pour isoler les opérations élémentaires pertinentes pour son problème. Il possède un alphabet sans savoir former les mots pour composer un message significatif. La composition des lettres pour former des mots puis des mots pour former des phrases est une intégration hiérarchisée qui suppose le franchissement de sauts qualitatifs dans le processus d'acquisition.

- Les contraintes pédagogiques.

Le temps consacré à l'apprentissage des concepts informatiques, à l'école et au collège, est d'une ampleur modeste (en moyenne une heure hebdomadaire, soit approximativement une trentaine d'heures sur l'année scolaire). Cette contrainte est importante car elle impose soit une limite du champ d'expertise, soit une limite du niveau d'expertise dans chacun des domaines abordés si ceux-ci sont trop nombreux.

3. Des idées directrices.

Cette analyse nous a conduit à opter pour une méthode d'apprentissage qui repose sur les principes suivants :

- Principe n° 1 : travailler avec les élèves sur la description d'objets ou de situations et sur le style de la représentation plus que sur un programme qui serait conçu comme un produit fini et utilisable tel quel.

- **Principe n° 2** : l'acquisition de concepts informatiques doit, comme tout apprentissage d'un champ de compétences étendu, se faire par modules. Cela signifie que l'on doit garder simultanément à l'esprit deux types d'objectifs :

a) un objectif local et restreint, l'acquisition d'expertises bien définies dans leur champ d'application, et

b) un objectif plus vaste et général, la coordination de ces différentes expertises locales pour permettre à l'apprenti-programmeur de faire face à de nouvelles situations.

- **Conséquence** : cette dernière distinction est importante car elle implique que la définition du champ d'application des expertises locales soit pensée en termes de classes de problèmes qu'il est possible, d'une part, de caractériser par un ensemble de propriétés communes et, d'autre part, de hiérarchiser par combinaison des différents formalismes locaux.

a) **En ce qui concerne la première classe d'objectifs**, l'acquisition d'une expertise locale, nous avons mis en œuvre les idées suivantes :

- Réduire une «situation de programmation» à un ensemble restreint de commandes (le vocabulaire) et de règles de composition (la grammaire) cohérent pour décrire une classe bien définie d'objets ou de situations problèmes. Chaque situation doit être pensée de manière à avoir une finalité bien précise en relation avec le type de problèmes qu'elle permet de résoudre. Suivant la même idée, le niveau de grammaire qu'elle implique pour aborder ces problèmes ne doit pas comporter, au départ, trop de difficultés.

- Apprendre à l'élève à relier les propriétés pertinentes d'une classe de problèmes à l'utilisation d'un formalisme adéquat et à certaines structures de contrôle. Ceci conduit l'élève à se poser le problème suivant : quelles sont les caractéristiques communes de tous les objets que je peux décrire avec les formes élémentaires dont je dispose ?

- Respecter les phases suivantes dans l'acquisition d'une expertise locale :

1) Identification des opérateurs de base et de la grammaire susceptibles de permettre d'arriver à une solution. Cette phase est rendue d'autant plus facile que l'ensemble des formes élémentaires est restreint.

2) Evaluation des effets produits en terme de distance entre le but poursuivi et le produit obtenu. Cette phase comprend autant la détection des erreurs que l'évaluation de la méthode utilisée et de son économie.

3) Etiquetage des principales étapes de la résolution afin de pouvoir résoudre, par la suite, des problèmes analogues sans avoir à reproduire le même cheminement dans la suite de l'exploration. Cette phase consiste à isoler et à identifier les principaux points critiques des problèmes ainsi que la manière dont ils peuvent être résolus.

4) Consolidation de l'expertise par l'exercice qui permet de gagner progressivement en rapidité et en charge de traitement, libérant par la même une disponibilité qui sera utilisée ultérieurement pour penser cette classe de problèmes bien précise en liaison avec d'autres expertises locales.

b) En ce qui concerne la seconde classe d'objectifs, ceux relatifs à la coordination des différentes expertises locales, leur prise en charge est plus délicate à l'école étant donné les limites temporelles et matérielles qui caractérisent l'utilisation de systèmes informatisés dans les classes. Il est néanmoins nécessaire de garder présent à l'esprit les principes suivants :

- Construire un ensemble hiérarchisé de situations cohérent du point de vue de l'enchaînement des structures de contrôle et/ou des structures de données qu'elles permettent de manipuler.

- Travailler sur des acquisitions qui ont un haut degré de portabilité. C'est-à-dire proposer des situations d'apprentissage qui, bien que leur champ d'application en extension dans chaque domaine soit limité, permettent d'aborder de nombreux problèmes différents dans des champs voisins.

Ces idées générales sont, pour le moment, des principes encore très abstraits pour le lecteur. Dans la partie suivante, nous allons montrer comment ils peuvent être mis en pratique de manière concrète à partir de thème du «dessin et de la représentation graphique».

4. Contenu du projet «Schèmes Informatiques Programmables».

Actuellement, seulement trois thèmes ont été développés. Chacun de ces thèmes aborde une classe de situations différentes qui permet de faire correspondre une technique d'écriture, une stratégie de construction des objets avec une série d'objectifs :

- Le GRAPHISME : représenter graphiquement des objets réels.
- L'EVOLUTION : faire grandir ou diminuer une forme.
- Les CYCLES : construire un compteur (horloge, calendrier...).

Chaque THEME comprend une introduction qui permet de situer son importance parmi l'ensemble des connaissances manipulables par les enfants et un ensemble de SITUATIONS qui conduiront l'apprenti-programmeur à la maîtrise d'objets de plus en plus complexes dans le domaine de connaissances considéré. Seul le premier thème, consacré au graphisme, sera développé ici. Une SITUATION est un ensemble de FORMES ELEMENTAIRES écrites en LOGO (des carrés, une aiguille, un déplacement...) plus un ensemble de FORMES COMPLEXES (une marelle, une horloge...), qui sont présentés comme des buts à atteindre. Ils disposeront, en plus de ces formes élémentaires, d'un micro-environnement pour tester la fonction de ces formes afin de leur donner une signification plus ou moins approximative dans un premier temps. Les formes sont regroupées en familles par classe d'effets.

Lorsque cette phase de familiarisation avec les formes élémentaires est terminée, nous présentons alors aux enfants des formes complexes que l'on peut obtenir par combinaison et arrangement des formes élémentaires. Les différentes applications suivent un ordre de complexité croissante. Un cheminement est proposé à titre d'exemple, il peut servir de prototype à des explorations sur des modes de coordination différents de celui qui est proposé.

Enfin, pour chaque application, nous proposons une série d'ACTIVITES dont le but est justement de palier aux effets négatifs du mimétisme sollicité dans les phases précédentes : l'élève peut modifier certaines des caractéristiques des formes complexes, trouver des variantes dans la construction de ces formes (d'abord sur des aspects ponctuels, ensuite sur des éléments plus importants), porter un jugement sur l'esthétique de certaines constructions, compléter l'ensemble des formes élémentaires et découvrir de nouvelles familles d'objets complexes, inventer des jeux à partir de la situation, faire varier des paramètres dans un espace qui lui permet de relier un ensemble de causes à une classe d'effets...

En résumé, l'élève se trouve confronté, pour chaque situation, à trois types de problèmes :

- Avec les formes dont je dispose, comment est-ce que je peux résoudre mon problème ?
- Avec les formes à ma disposition, qu'est-ce que je peux inventer ?
- Quelles nouvelles formes puis-je inventer pour résoudre mon problème d'une autre façon ?

II - GRAPHISME ET DESSIN : représenter des objets, des figures ou des schémas sur un écran.

Tout système informatique qui permet de représenter graphiquement des objets sur un écran repose plus ou moins implicitement sur un (ou des) formalisme(s) de base. Nous appellerons, par la suite, «formalisme de base» l'ensemble des primitives qui permettent d'obtenir une trace graphique à l'écran ainsi que celles qui leur sont associées (déplacement, gestion de l'écran). Cet ensemble de commandes peut être plus ou moins indépendant de la «grammaire» de ce langage, ainsi que de son système d'exploitation. La «grammaire» du langage, c'est l'ensemble des possibilités offertes par le système pour composer des commandes ou des sous-ensembles de commandes. Par exemple, la première règle de grammaire, valable sur presque tous les systèmes, c'est la «séquentialité» : chaque instruction est exécutée l'une après l'autre dans l'ordre d'écriture. Les règles de grammaire peuvent être «visibles» c'est-à-dire correspondre à une instruction (REPETE n [liste d'instruction] permet de composer plusieurs listes identiques entre elles) ou «invisibles» comme dans le cas de la séquentialité. Le «système d'exploitation» c'est l'ensemble des commandes qui permet de gérer l'édition, la sauvegarde et la récupération des lignes de codes produites. La maîtrise de ce système, que l'on confond souvent avec l'apprentissage du code proprement dit, est indispensable pour que l'élève puisse s'exprimer vraiment dans une activité de programmation.

Le formalisme de la tortue LOGO est, de loin, celui des formalismes graphiques qui a été le plus fréquemment utilisé avec les enfants. Des expériences, plus ou moins convaincantes, ont été conduites et il ressort de ces travaux que la principale difficulté pour les enfants réside dans la coordination des différents objets qui composent une figure. Cette décomposition peut porter sur des droites ou plus généralement sur des formes plus complexes, aisément identifiables par les enfants. Malgré tout, le formalisme de base implicite de la tortue est la droite. Cette unité, qui fait référence au déplacement d'un crayon, induit presque toujours une description spontanée des objets qui s'apparente au tracé de figures géométriques sur un support traditionnel, comme la feuille de papier, en excluant la possibilité de redondance et de superposition du tracé (Mendelsohn, 1984). Nous proposons ci-dessous, à titre d'illustration de notre démarche, un exemple de progression dans lequel le formalisme de base n'est plus uniquement la droite mais aussi des familles de formes, bien connues par ailleurs des élèves, les carrés et les rectangles. Notre objectif est que les enfants mettent en place des acquisitions locales dans la manipulation de ces formalismes en construisant, pour chacun d'eux, la classe des objets qui sont facilement réductibles à cette description. Chacune des situations présentée ici comprend d'une part la liste des instructions de

base que nous appellerons les «Formes Élémentaires» et un échantillon restreint des «Formes Complexes» que nous demandons aux sujets de décrire à l'aide du formalisme correspondant. A titre d'exemple, nous avons aussi rapporté les principaux commentaires qui accompagnent nos séances. Ils traduisent (ou trahissent !!) la nature du contrat que nous avons passé entre les élèves et nous.

Trois formalismes de base ont été expérimentés :

La SITUATION 0 est construite sur un formalisme «relationnel», c'est-à-dire qu'à chaque instruction correspond un objet graphique (ou un déplacement) et un seul. Il s'agit ici de la famille des «carrés».

La SITUATION 1, qui reprend un sous-ensemble de commande Logo classiques, est un formalisme «dimensionnel». Le tracé s'obtient par des primitives qui nécessitent une mesure de longueur (ici le pas «tortue»).

La SITUATION 2 correspond à un formalisme «vectoriel». Les objets graphiques, des rectangles, sont déterminés par deux dimensions coordonnées (longueur et largeur).

Pour mémoire, puisqu'il n'a pas été utilisé dans cette expérience mais qu'il fait partie de l'arsenal graphique de Logo et de bien d'autres systèmes, nous citerons un second formalisme vectoriel, le point. En effet, chaque point est déterminé par deux valeurs correspondant à sa projection sur un système orthonormé (abscisse et ordonnée). Ce formalisme s'avérera fort utile pour la description d'objets graphiques de formes irrégulières.

Les enfants doivent construire les objets que nous leur proposons en utilisant uniquement les formes disponibles dans chaque situation. Le travail consiste à trouver une description des objets complexes la plus concise possible. La comparaison des différentes solutions fait l'objet d'une discussion collective à chaque séance.

Les procédures utilisées dans les 3 situations sont détaillées sous la version Logo MO5 du Nanoréseau équipant les écoles et collèges français. Leur implantation sur d'autres systèmes est simple. Consulter le manuel de référence de votre matériel pour opérer les adaptations nécessaires.

A. DESSINER AVEC DES CARRES : UN FORMALISME «RELATIONNEL».

Présentation.

Objectifs.

- Tracer sur l'écran, l'aide de carrés de côté constant, des formes simples et bien connues des élèves.

- Apprendre à analyser une figure pour déterminer si elle se «laissera» décrire à l'aide de ce formalisme particulier.

- Maîtriser progressivement la grammaire de base du système : séquentialité, modularité, répétition.

- Exercer et automatiser la «conduite» du système d'édition et d'exploitation afin de libérer progressivement l'attention de cette tâche, souvent peu pertinente du point de vue des activités de programmation proprement dites.

Caractéristiques de la situation.

- Les formes proposées (voir page suivante) obligent les élèves à focaliser leur attention sur les rapports entre les éléments de la figure par le choix du ou des CARRES pertinent(s) pour résoudre le problème.

- Les figures à décrire permettent de nombreuses variations sur le thème de la répétition (les carrelages, les pavages). Ce type de travail est une préparation indispensable à l'introduction de l'itération.

Prérequis.

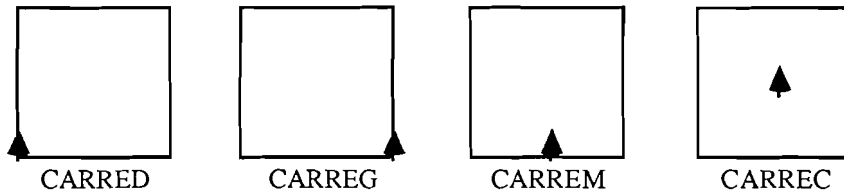
- Une connaissance de l'éditeur et du système d'exploitation de Logo sont indispensables pour bien profiter des applications proposées. Mais la simplicité des commandes permet d'utiliser cette situation pour une initiation à ces outils.

1. Les formes élémentaires.

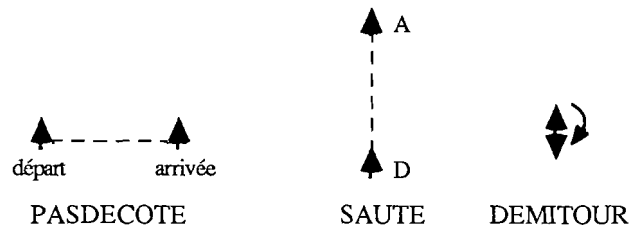
Il y a six familles de formes élémentaires. Les trois premières, formes tracées (1), formes déplacement (2), formes positions (3), sont un tout indissociable et cohérent.

Les formes «petits objets» (4) et le «moteur» (4) permettent des applications plus spécifiques sur les carrelages et la phase d'initiation.

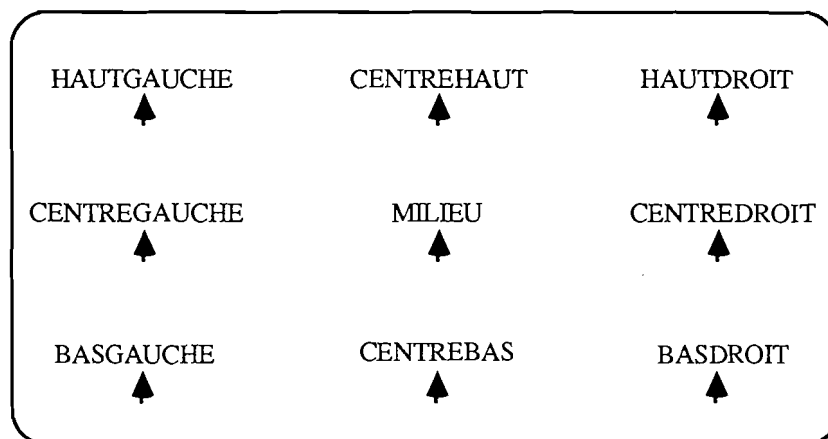
a. FORMES «TRACEES».



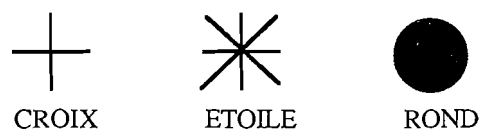
b. FORMES «DEPLACEMENTS» : il s'agit de déplacements relatifs qui dépendent de la position initiale de la tortue. Un déplacement ne laisse pas de trace sur l'écran.



c. FORMES «POSITIONS» : ces dernières sont des déplacements absolus et le point d'arrivée ne dépend pas du point de départ.



d. FORMES «PETITS OBJETS».



e. MOTEUR POUR L'INITIATION.

Effet : une fois lancé, c'est-à-dire la procédure MOTEUR exécutée, le moteur permet, en appuyant seulement sur une touche, de faire les actions correspondantes à certaines formes élémentaires : ? MOTEUR (Touche entrée).

Le système est en attente, on peut appuyer sur l'une des touches suivantes, la commande correspondante sera exécutée :

| | |
|---|--|
| S | SAUTE |
| P | PASDECOTE |
| D | DEMITOUR |
| C | CROIX |
| E | ETOILE |
| R | ROND |
| M | MILIEU |
| K | CARREC |
| F | FIN (cette dernière touche permet de sortir du moteur) |

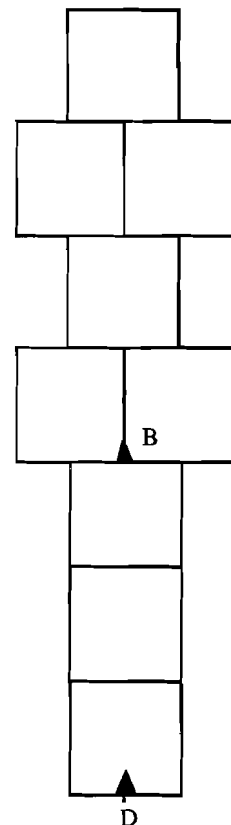
f. COMMANDES DE GESTION DE L'ECRAN.

| | |
|-------|--------------------|
| DEBUT | Initialise l'écran |
| VE | Vide écran |
| VT | vide texte |

2. Dessiner une marelle avec les carrés.

Analyse du tracé.

La marelle est composée de carrés qui ont tous la même taille que la série de formes élémentaires proposée. Les élèves n'ont pas de peine à reconnaître dans la figure un assemblage de CARRES (phase d'activation). Il leur faut maintenant trouver une règle d'assemblage. Quel CARRE choisir ? D'où faut-il démarrer ? Partir naturellement d'en bas paraît être une solution intéressante car cela permet d'utiliser une série répétitive de CARREM SAUTE et de se retrouver ainsi en B entre les deux carrés de la 4ème rangée. Les «carrés doubles» peuvent ainsi être facilement fabriqués par une succession, sans déplacement, de CARRED CARREG.



Cela nous donne, dans une version organisée par la structure répétitive :

POUR MARELLE

CENTREBAS

REPETE 3 [CARREM SAUTE]

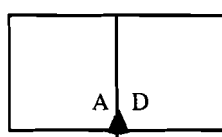
REPETE 2 [CARRED CARREG SAUTE CARREM SAUTE]

CENTREBAS

FIN

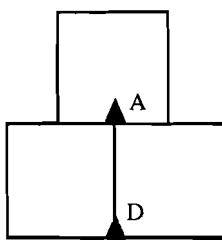
L'objet est ici décrit directement par des formes élémentaires. La modularité, autre règle de grammaire du système, permet de construire des objets intermédiaires. Dans le cas de la marelle, on peut choisir, entre autres solutions :

POUR BICARRE
CARRED CARREG
FIN



ou encore

POUR PODIUM
CARRED CARREG
SAUTE
CARREM
FIN



La marelle se décrit de manière encore plus simple maintenant.

POUR MARELLE

CENTREBAS

REPETE 3 [CARREM SAUTE]

REPETE 2 [PODIUM SAUTE]

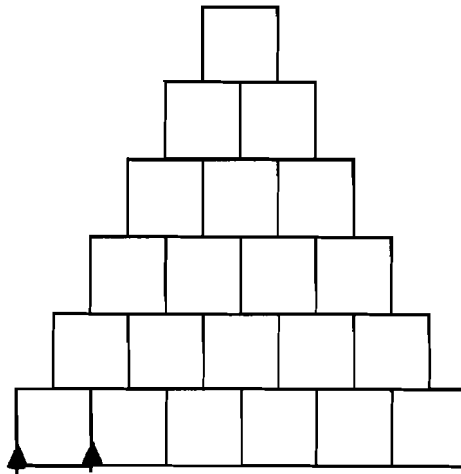
CENTREBAS

FIN

D'autres formes complexes peuvent être construites sur le même modèle. Certaines formes intermédiaires, comme PODIUM ou BICARRE, sont utilisables dans des projets différents. A ce stade, il est surtout important de travailler avec les enfants la grammaire du langage pour commencer à former des descriptions les plus concises et expressives possible. L'enfant apprend à reconnaître si la figure proposée est bien compatible avec le formalisme des CARRES. On peut introduire des contraintes du type : «ne pas se servir de la commande XXX» pour inciter les élèves à travailler certaines coordinations et à bien identifier le rôle et la fonction de la commande dans l'univers de la situation.

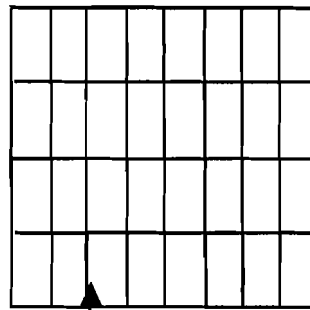
L'exemple suivant (PYRAMIDE) ne contient pas de DEMITOUR. Il utilise les relations possibles entre déplacements relatifs et déplacements absolus.

POUR PYRAMIDE
 BASGAUCHE
 REPETE 3 [PODIUM PASDECOTE SAUTE]
 BASGAUCHE
 REPETE 2 [PASDECOTE]
 REPETE 2 [PODIUM PASDECOTE SAUTE]
 BASGAUCHE
 REPETE 4 [PASDECOTE]
 REPETE 1 [PODIUM PASDECOTE SAUTE]
 BASGAUCGE
 FIN



La figure suivante laisse parfois les élèves (et même les adultes) perplexes. Ils ne pensent pas, a priori, que la GRILLE puisse être décrite à partir de CARRÉS. Il suffit de penser à la superposition de carrés, grâce au décalage autorisé par CARREM et CARRED ou CARREG, pour obtenir un résultat positif.

POUR GRILLE
 CENTREBAS
 REPETE 4 [CARREG CARREM CARRED SAUTE]
 CENTREBAS REPETE 2 [PASDECOTE]
 REPETE 4 [CARREG CARREM CARRED SAUTE]
 CENTREBAS
 FIN



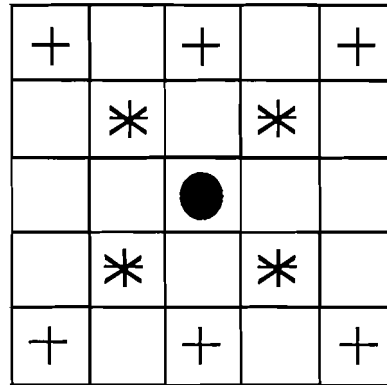
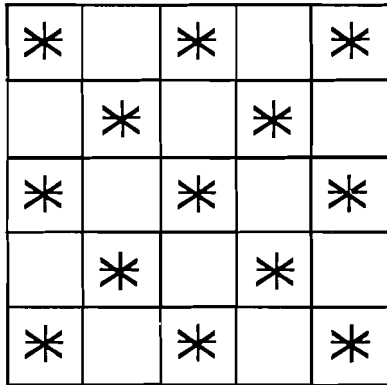
3. Les carrelages.

Cette famille de figures peut être utilisée dans plusieurs contextes :

1. Avant tout apprentissage du système d'édition, en association avec une procédure GRILLE qui trace le fond, il permet au débutant d'attribuer une signification précise aux commandes de déplacements grâce au MOTEUR.

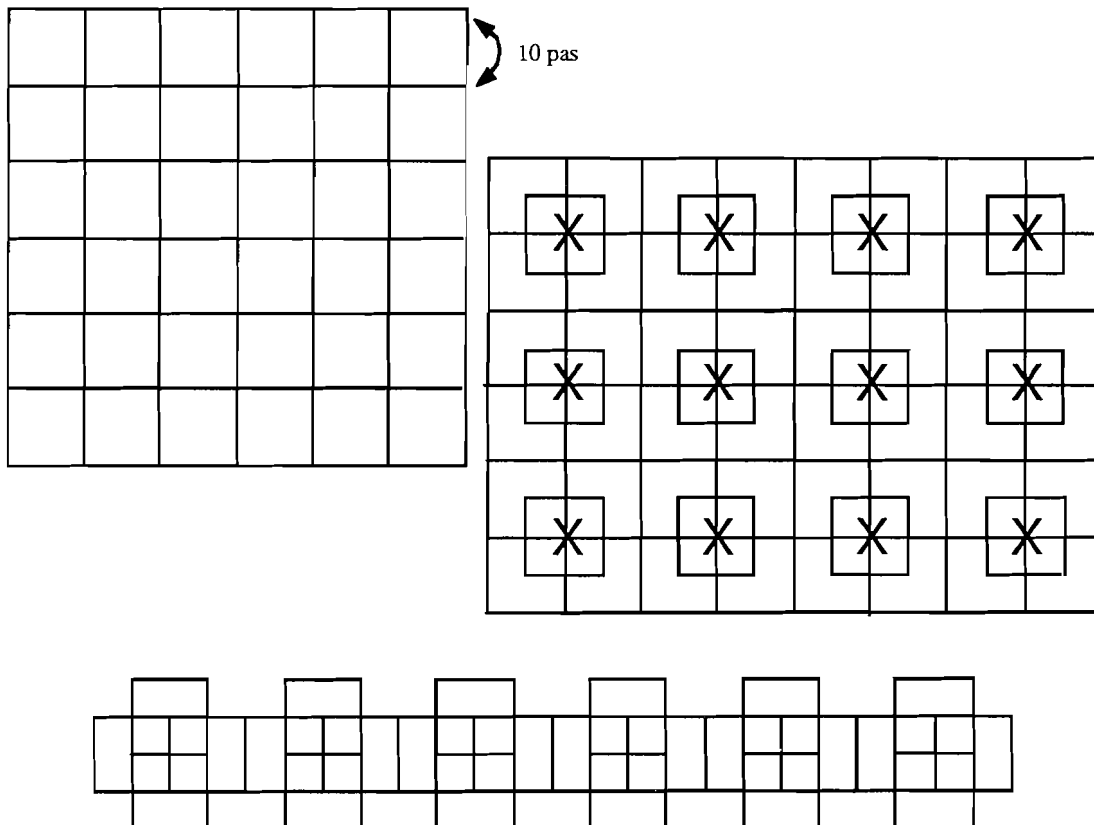
2. On peut ensuite inviter l'élève à construire une GRILLE de 5 par 5 en travaillant comme pour les exemples des pages précédentes.

3. Cette GRILLE associée au MOTEUR permet de jouer au Morpion, chaque enfant choisissant une forme «Petit Objet» à déposer dans les carrés. (Attention, pour cette dernière application, la GRILLE doit être composée de CARREC, afin que la «Tortue» se trouve au centre des carrés).



4. Les pavages.

Les figures ci-dessous, que l'on peut faire varier à l'infini, permettent de travailler très efficacement sur la structure répétitive et la modularité simple en même temps que la superposition du dessin. On peut inviter les élèves à inventer de nouveaux pavages, en leur demandant de bien veiller à leurs caractéristiques pour qu'ils puissent toujours être décrits à l'aide du formalisme CARREx.



5. Macroprocédures utilisées pour la situation \emptyset .

POUR CROIX
REPETE 4 [AV 5 RE 5 DR 90]
FIN

POUR ROND
REPETE 20 [AV 5 RE 5 BC DR 18]
FIN

POUR ETOILE
REPETE 8 [AV 5 RE 5 DR 45]
FIN

POUR MILIEU
LC FPOS [0 1] BC
FIN

POUR CENTREGAUCHE
LC FPOS [-120 1] BC
FIN

POUR BASGAUCHE
LC FPOS [-120 -59] BC
FIN

POUR CENTREBAS
LC FPOS [0 -59] BC
FIN

POUR BASDROIT
LC FPOS [120 -59] BC
FIN

POUR CENTREDROIT
LC FPOS [120 1] BC
FIN

POUR HAUTDROIT
LC FPOS [120 81] BC
FIN

POUR CENTREHAUT
LC FPOS [0 81] BC
FIN

POUR HAUTGAUCHE
LC FPOS [-120 81] BC
FIN

POUR DEMITOUR
DR 180
FIN

POUR PASDECOTE
LC DR 90 AV 20 GA 90 BC
FIN

POUR SAUTE
LC AV 20 BC
FIN

POUR CARREC
LC RE 10 GA 90 AV 10 DR 90 BC
CARRED
LC AV 10 DR 90 AV 10 GA 90 BC
FIN

POUR CARREM
GA 90 AV 10 DR 90
CARRED
DR 90 AV 10 GA 90
FIN

POUR CARREG
REPETE 4 [AV 20 GA 90]
FIN

POUR CARRED
REPETE 4 [AV 20 DR 90]
FIN

POUR MOTEUR
DONNE "REP RC
SI :REP = "C [CROIX]
SI :REP = "E [ETOILE]
SI :REP = "R [ROND]
SI :REP = "S [SAUTE]
SI :REP = "P [PASDECOTE]
SI :REP = "D [DEMITOUR]
SI :REP = "F [STOP]
SI :REP = "K [CARREC]
MOTEUR
FIN

POUR DEBUT
FENETRE
VE
VIDETEXTE
FIN

B. LA TORTUE LOGO POUR TRACER : UN FORMALISME DIMENSIONNEL.

Présentation.

Objectifs.

- Familiarisation avec les problèmes posés par la représentation graphique d'un espace réel : analyse du tracé, échelle, cadrage, procédure de construction. En géométrie, les notions suivantes seront abordées : notion de symétrie, repérage de formes géométriques, utilisation d'un repère orthonormé, formes semblables.

- Associer ces connaissances (les caractéristiques géométriques de la figure) à des structures de programmation : approche de la construction modulaire des procédures par la décomposition de la figure en éléments coordonnés les uns aux autres.

- Construction d'«objets extensibles» sur une ou plusieurs dimensions.

Caractéristiques de la situation.

- Cette situation permet une application des principales commandes graphiques du Logo : mouvements relatifs, déplacements sans tracé, déplacements absolus. Le concept de base que les élèves doivent maîtriser, est ici la dimension «longueur». C'est pourquoi nous avons choisi d'imposer les contraintes du tracé de l'extérieur par les dimensions réelles des objets à représenter et non par une recherche artificielle de régularités. Le modèle, en imposant ses caractéristiques, oblige, d'une part à une analyse précise des formes élémentaires qui la composent relativement à leur dimension et aux rapports entre ces dimensions et, d'autre part, à les traduire à l'aide des structures de programmation.

Prérequis.

- Connaissance et manipulation de l'éditeur LOGO.

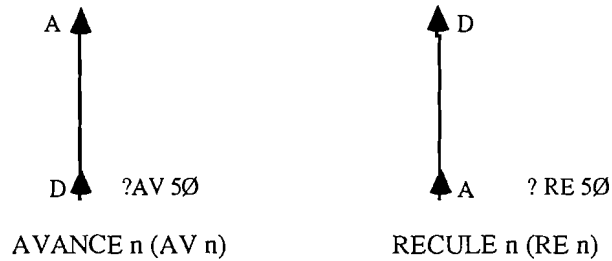
- Sensibilisation aux déplacements de la tortue-graphique, connaissance des abréviations les plus courantes.

1. Formes élémentaires.

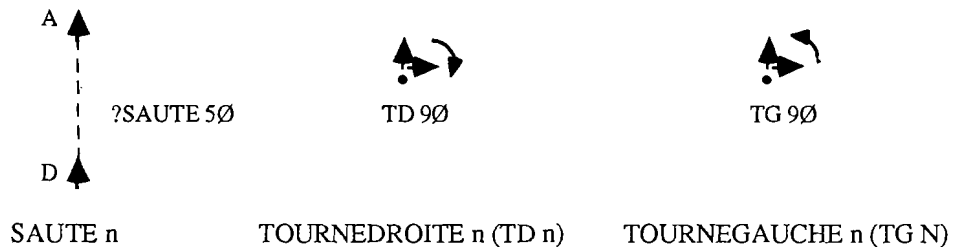
Nous utiliserons cinq familles de commandes regroupées de la façon suivante : (a) les formes «tracé» et (b) les formes «déplacement-rotation», (c) les formes «positions», (d) les formes «objets» et (e) les outils de gestion de l'écran.

Les commandes sont soit des commandes Logo classiques (avec ou sans restriction d'usage), soit des commandes conçues à partir de ces mêmes primitives.

a. FORME «TRACE».



b. FORMES «DEPLACEMENT-ROTATION».



On peut, en fonction des objectifs poursuivis, limiter les valeurs des rotations à 90° et 180° . La situation ne permet, alors, de traiter que des figures qui comprennent des angles droits. On limite ainsi dans un premier temps les problèmes de gestion de rotations angulaires. Les applications décrites ci-après intègrent cette limitation.

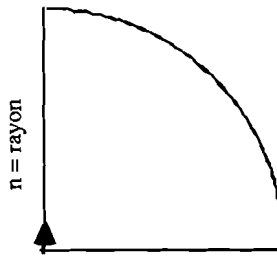
c. LES FORMES «POSITION».

| | | | |
|------------|-----|-----|-----|
| | . 8 | . 1 | . 2 |
| POSITION n | . 7 | . 0 | . 3 |
| | . 6 | . 5 | . 4 |

n correspond au secteur choisi (voir schéma). La tortue se positionne au centre du secteur sélectionné.

d. LES FORMES «OBJETS».

QUARTDECERCLE n
(n est le rayon du CERCLE)

**e. LES COMMANDES DE «GESTION-ECRAN».**

| | | |
|--------------|------|---|
| VIDEcran | (VE) | Efface l'écran graphique. |
| VIDETexte | (VT) | Efface la page texte. |
| PLEINEcran | (PE) | Pleine page graphique. |
| MIXEcran | (ME) | Page graphique plus quatre lignes de texte. |
| MONTRETORTUE | (MT) | Fait apparaître la tortue. |
| CACHETORTUE | (CT) | Fait disparaître la tortue. |

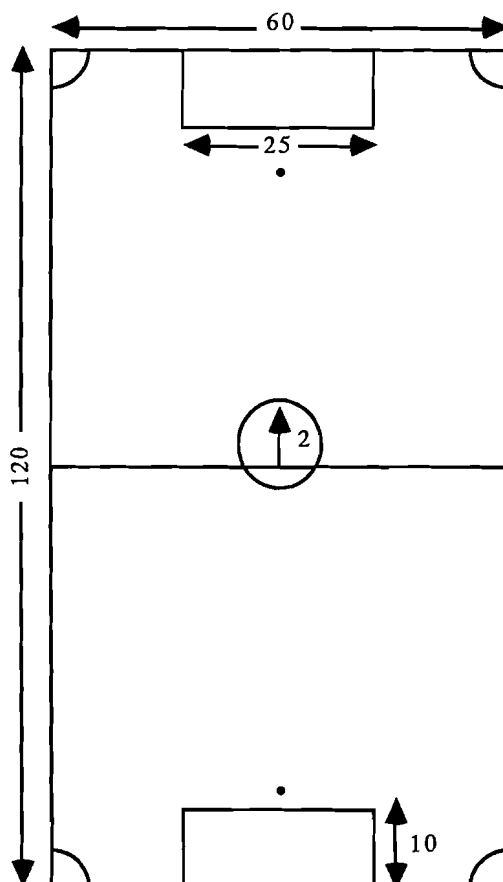
2. Dessiner un terrain de football avec la tortue.

Analyse du tracé : reconnaissance de formes, symétries, échelle.

Le terrain de football que nous nous proposons de tracer est une version un peu simplifiée d'un vrai terrain. Il est constitué par ses limites extérieures qui ont la forme d'un **rectangle**, la ligne **médiane** qui sépare les deux camps, deux **rectangles** en face de chaque but qui détermine une surface de jeu particulière (la surface de réparation), et pour finir un petit **rond** central.

En s'appuyant sur la connaissance (même approximative) du jeu, on peut remarquer certains **éléments de symétrie** dans la figure :

- la ligne médiane ;
- le point central ;
- les cages sont au centre des petits côtés.



Les dimensions données sur le schéma serviront à respecter les proportions du dessin sur l'écran. Pour commencer on peut adopter une convention simple du type : 1 mètre est représenté par un pas de tortue (1 m = 1 pas).

Activités.

- Faire décrire le terrain aux enfants en reliant la connaissance des formes géométriques repérées à des lieux réels de déplacement.
- Trouver les lignes et les points remarquables : axes et points de symétrie.
- Représenter le terrain sur une feuille quadrillée en adoptant une échelle simple du genre : 1 carreau représente 10 mètres. Faire remarquer aux enfants la nécessité de tenir compte de la position initiale du centre pour cadrer correctement le terrain sur la feuille.

Tracé guidé du terrain.

Le cheminement suivant est une des manières qui permet de tracer le terrain de foot. Suivre pas à pas cette construction permet de coordonner les connaissances acquises lors de l'analyse de la figure à des structures de programmation.

Le tracé se décompose en 5 étapes.

- La première étape consiste à tracer la ligne médiane et le pourtour du terrain.
- La seconde permet de dessiner les sections rectangulaires des deux surfaces devant chaque cage.

Pour ces deux premières étapes, nous décrirons deux variantes :

- La première, peu structurée consiste à suivre mentalement pas à pas le trajet que ferait un sujet qui ne sait qu'avancer et tourner à angle droit.

- La seconde méthode, plus structurée, consiste à répéter des séquences d'instructions.

- Dans la troisième étape, les deux premières séquences (étape I et II) seront assemblées pour construire la procédure principale à partir des deux sous-procédures initiales (modularité).

- Enfin, la quatrième étape permet d'ajouter quelques détails au tracé : rond central et, éventuellement, marque des «corners» en insérant une sous-procédure dans le programme principal.

ETAPE I.

Au départ, la tortue est au centre de l'écran comme le ballon au début de la partie. Si ce n'est pas le cas taper VE (Vide Ecran).

1ère variante.

POUR LIMITE1

AV 30 TG 90 AV 40 TG 90

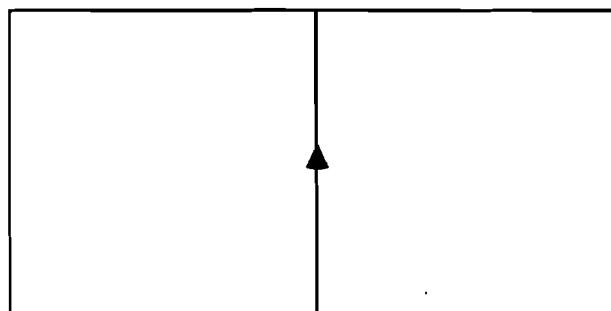
AV 60 TG 90 AV 40 TG 90

AV 60 TD 90 AV 40 TD 90

AV 60 TD 90 AV 40 TD 90

AV 30

FIN



Noter que la procédure ramène la tortue au centre du terrain.

2ème variante.

POUR LIMITE2

REPETE 2 [AV 30 TG 90 AV 40 TG 90 AV 30]

REPETE 2 [AV 30 TD 90 AV 40 TD 90 AV 30]

FIN

Chaque séquence répétée consiste à tracer la moitié du rectangle formé par chaque moitié du terrain.

Au besoin faire constater aux élèves l'effet de chacune des séquences de l'itération en l'exécutant au niveau supérieur avec la tortue en position centrale.

ETAPE II.

La tortue est de nouveau au centre, comme pour la première étape. Le projet consiste à tracer les surfaces de réparation en se servant de l'axe de symétrie horizontal du terrain pour se déplacer au bon endroit.

1ère variante.

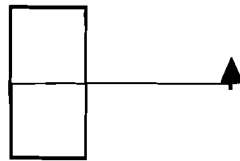
POUR SURFACE 1

TG 90 SAUTE 40 TD 90

AV 20 TD 90 AV 20 TD 90

AV 40 TD 90 AV 20 TD 90

PLACE 0



2ème variante.

POUR SURFACE2

TG 90 SAUTE 40 TD 90

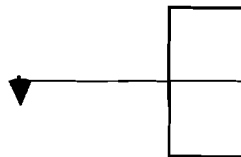
REPETE 2 [AV 20 TD 90 AV 20 TD 90 AV 20]

PLACE 0

FIN

Pour dessiner la deuxième surface à droite, on remarquera qu'elle peut s'obtenir par le même tracé si on dirige préalablement la tortue vers le bas. Regarder la figure à l'envers pour s'en assurer.

?TD 180 SURFACE2



C'est presque terminé pour le gros œuvre !

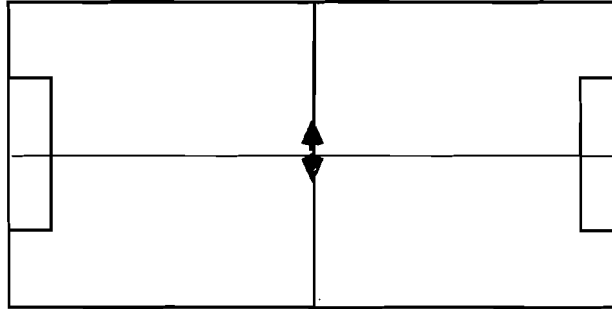
ETAPE III.**Pour dessiner le terrain d'un seul coup.**

Il suffit maintenant de rassembler les différentes étapes pour construire une procédure qui dessine l'ensemble du terrain :

```

POUR TERRAIN
LIMITE1
SURFACE1
TD 180
SURFACE1
FIN

```



L'assemblage des sous-procédures dans la procédure principale est ici facilité par le fait que la tortue se retrouve toujours au centre. La coordination de ces sous-procédures est réduite à une rotation de 180° entre les deux appels de SURFACE1. Le choix du point de départ joue donc un rôle déterminant dans l'organisation du travail.

Construire un programme structuré c'est comme apprendre à bien parler. Il faut employer le mot juste qui prend tout son sens dans le contexte de la phrase et facilite la compréhension chez l'interlocuteur. Il ne faut pas seulement programmer pour obtenir un résultat, il faut penser avant tout à se faire comprendre. Travaillez comme si un être imaginaire lisait par dessus votre épaule.

ETAPE IV.**Le tracé du rond central.**

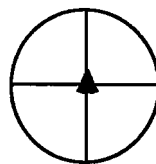
Il y a plusieurs manières de concevoir un cercle. Nous nous référerons ici à une conception classique : un cercle est un ensemble de points équidistants d'un point appelé «centre». Il faut pour cela utiliser la forme QUARTDECERCLE.

Pour obtenir un cercle de 6 pas de rayon :

```

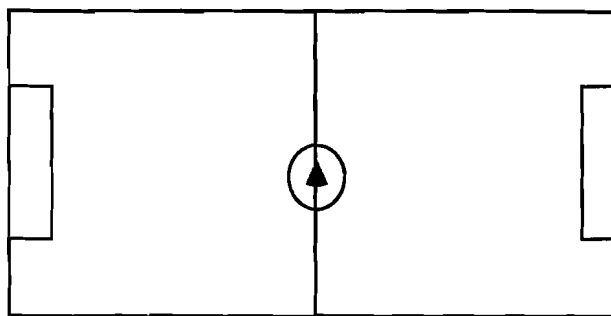
POUR ROND
REPETE 4 [QUARTDECERCLE 6 TD 90]
FIN

```



Il nous reste à insérer cette nouvelle procédure dans la procédure principale :

POUR TERRAIN
 LIMITE1
 SURFACE1
 DR 180
 SURFACE1
 ROND
 FIN



L'insertion d'une sous-procédure dans le programme principal est une activité essentielle en programmation comme d'ailleurs en situation de résolution de problème. Cela suppose que chaque élément de la figure soit défini indépendamment de son contexte pour être ensuite coordonné aux autres dans la procédure principale. Nous retrouverons cette démarche dans de nombreuses occasions par la suite.

PROLONGEMENTS.

Compléments au programme.

- Compléter le tracé par des quarts de cercle à chaque angle du terrain pour représenter les «corners» (ATTENTION à l'orientation initiale de la tortue pour obtenir le quart de cercle voulu).

Variantes.

- Essayer de trouver d'autres variantes pour dessiner le terrain et les comparer aux deux premières. En quoi sont-elles différentes ?

Echelle.

- Dessiner un terrain dont chaque dimension est deux fois plus petite que le terrain initial.

Jeu.

- En utilisant SAUTE n (pour ne pas laisser de trace sur l'écran), placer la tortue à n'importe quel endroit du terrain et dans une orientation quelconque. Demander à l'enfant de marquer un but, avec la tortue comme ballon, en utilisant une seule instruction de rotation et une seule instruction de déplacement.

Ce jeu permet à l'élève d'acquérir une bonne intuition des ordres de grandeur des rotations ainsi que des déplacements.

Pour réaliser cette dernière application, il convient de lever les restrictions concernant l'usage des degrés avec les primitives TD n et TG n.

3. Construction de formes semblables.

Comme nous venons de le voir à propos du tracé du terrain de football, Logo est un langage de programmation modulaire et procédural. Il est toujours possible de définir de nouvelles procédures qui contrôlent, à leur tour, d'autres procédures ou des primitives du langage. Un programme correctement écrit est un empilement d'objets intégrés hiérarchiquement de manière à, d'une part effectuer une tâche particulière (dessiner, calculer, trier...), et, d'autre part, à pouvoir être une représentation intelligible de ce même problème.

Ce second aspect de la programmation structurée n'est pas seulement une distraction pour programmer en mal d'esthétisme, il devient une obligation pour aborder des problèmes d'une certaine complexité. Rappelons-nous que l'intelligence est une fonction organisatrice qui permet à notre «faible» capacité de stockage en mémoire de travail de traiter des objets dont la complexité croissante est liée à une prise en compte, non moins croissante, de nouvelles propriétés dans les situations problèmes que nous cherchons à résoudre. Faire plus dans le même espace oblige à relier, coordonner, généraliser. Apprendre la programmation structurée permet, sur un système artificiel, de simuler cette activité ainsi que de mettre en évidence les différentes étapes de ce mécanisme d'intégration hiérarchique.

Une phase clef de ce mécanisme consiste à généraliser une solution découverte pour la solution d'un problème particulier. Le graphisme et le système d'exploitation de Logo offrent de nombreuses possibilités d'applications de ce mécanisme. L'apprentissage d'un système de commande relié au formalisme dimensionnel (AV n...) conduit tout naturellement à vouloir construire des procédures qui, tout en gardant une forme semblable, permettent sa représentation sur des tailles apparentes différentes. Conceptuellement, cette étape est une étape décisive. L'objet (représenté par sa procédure paramétrisée) devient un objet plus puissant puisqu'il contient virtuellement une infinité de formes apparentes. Ce changement quantitatif pourra même déboucher sur des changements qualitatifs lorsque l'on commencera à faire covarier les valeurs des paramètres ou lorsque l'on s'amusera à effectuer des passages à la limite (une droite peut ainsi devenir un point, un rectangle se transformer en droite...).

Le quadrillage extensible.

L'objectif est d'écrire une procédure paramétrisée qui trace n'importe quel quadrillage de $N \times N$ carrés le longueur variable.

ETAPE I : résoudre le problème sur un cas particulier.

Représenter, à l'aide du formalisme de la tortue, un quadrillage (5 x 5) composé de carreaux de 20 pas de côté. La solution proposée ci-dessous suppose la construction d'une représentation structurée de la figure en trois points :

1. Le quadrillage est composé de 2 ensembles de droites parallèles disposés perpendiculairement l'un par rapport à l'autre.

2. Chaque ensemble est formé de 6 «droites» de 100 pas, équidistantes l'une de l'autre de 20 pas.

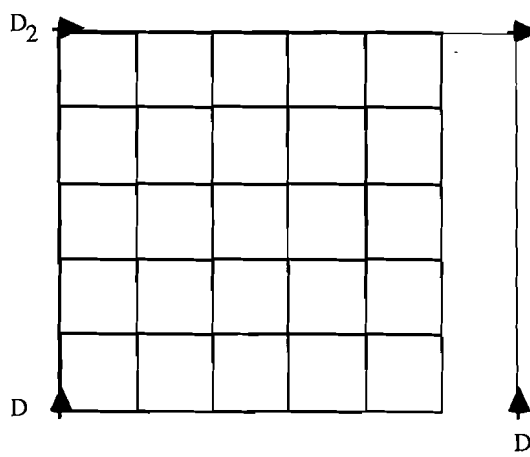
3. Chaque droite est un aller et retour de la tortue, chaque déplacement un SAUTE.

```

POUR QUADRILLAGE
GRILLE
SAUTE 100 TD 90 SAUTE -120
GRILLE
FIN

POUR GRILLE
REPETE 6 LIGNE DEPLACEMENT
FIN

```



| | |
|-----------------------|----------------------|
| POUR LIGNE | POUR DEPLACEMENT |
| AVANCE 100 RECULE 100 | TD 90 SAUTE 20 TG 90 |
| FIN | FIN |

ETAPE II : généraliser la solution obtenue.

Cette seconde étape suppose que l'on coordonne le premier objectif, tracer un quadrillage de taille quelconque, avec l'analyse en trois points effectuée dans l'étape I. Cela revient à focaliser maintenant toute son attention sur les rapports qui existent entre ce qui varie dans un quadrillage extensible et ce qui demeure invariant. Les éléments de variation seront formalisés par des paramètres :N pour le nombre de carreaux et :COTE pour la longueur du côté d'un carreau.

1. Les variations portent sur la longueur des lignes, sur la valeur des déplacements et sur le nombre de répétition de la séquence LIGNE DEPLACEMENT.

2. Les invariants sont constitués par les rapports entre ces éléments, c'est-à-dire la suite des séquences d'instructions et les angles de rotation.

POUR QUADRILLAGE :N :COTE

GRILLE

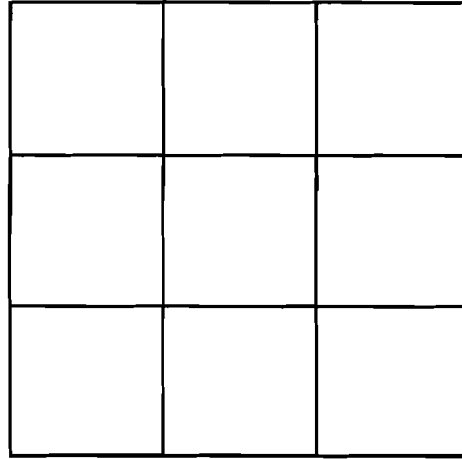
SAUTE :N * :COTE

TD 90

SAUTE -(N * :COTE + :COTE)

GRILLE

FIN



?QUADRILLAGE 3 40

POUR GRILLE

REPETE :N LIGNE DEPLACEMENT

FIN

POUR LIGNE

AV :N * :COTE

RE :N * :COTE

FIN

POUR DEPLACEMENT

TD 90 SAUTE :COTE

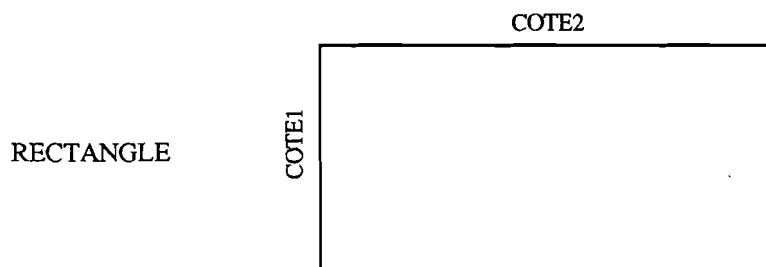
TG 90

TG 90

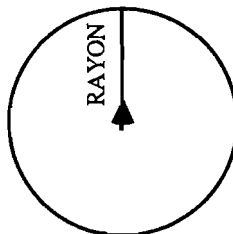
FIN

PROLONGEMENTS.

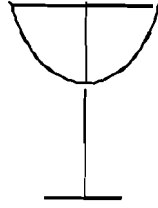
Faire un travail analogue sur les objets suivants.



ROND



VERRE



4. Macroprocédures utilisées pour la situation 1.

AVANCE n (AV n) primitive LOGO

RECULE n (RE n) "

POUR SAUTE :N

LC AV :N BC

FIN

TOURNEDROITE n (TD n) primitive LOGO

TOURNEGAUCHE n (TG n) "

POUR PLACE :N

SI :N = 8 [LC FPOS [-100 80] BC]

SI :N = 1 [LC FPOS [0 80] BC]

SI :N = 2 [LC FPOS [100 80] BC]

SI :N = 7 [LC FPOS [-100 0] BC]

SI :N = 0 [LC FPOS [0 0] BC]

SI :N = 3 [LC FPOS [100 0]]

SI :N = 6 [LC FPOS [-100 -80] BC]

SI :N = 5 [LC FPOS [0 -80] BC]

SI :N = 4 [LC FPOS [100 -80] BC]

FIN

POUR QUARTDECERCLE :N

TD 90 ATTENDS 10 CT

REPETE 180 [LC AV :N BC AV 0.1 RE 0.1 LC RE :N TG 0.5]

MT

FIN

C. DESSINER AVEC DES RECTANGLES : UN FORMALISME VECTORIEL.

Présentation

Objectifs.

- Dessiner sur l'écran, à l'aide d'une forme déterminée par deux dimensions (vecteur), des figures représentant des schémas ou des projections d'objets réels.
- Apprendre à analyser une forme géométrique plane pour déterminer si elle peut se décrire à partir de ce formalisme.
- Améliorer la maîtrise des règles de composition des commandes Logo (grammaire) et la conduite du système d'exploitation.

Caractéristiques de la situation.

- Les formes proposées (RECTANGLE_x..) autorisent, par inclusion de classe, de traiter au moins tous les problèmes de la SITUATION 0. En effet, RECTANGLED 20 20 c'est CARRED, SAUTE 20 c'est SAUTE...
- Il en est de même pour la SITUATION 1. RECTANGLED x 0 est une «droite» dessinée à partir d'une extrémité avec retour de la tortue au point de départ (RECTANGLED 50 0 SAUTE 50 est équivalent à AVANCE 50).
- Le formalisme des RECTANGLE_x.. permet une représentation des figures par superposition de rectangles avec, pour certaines figures, un déplacement minimum de la tortue. Ce qui n'est pas le cas des formalismes de SITUATION 0 et de SITUATION 1 qui supposent obligatoirement un déplacement plus important du crayon.

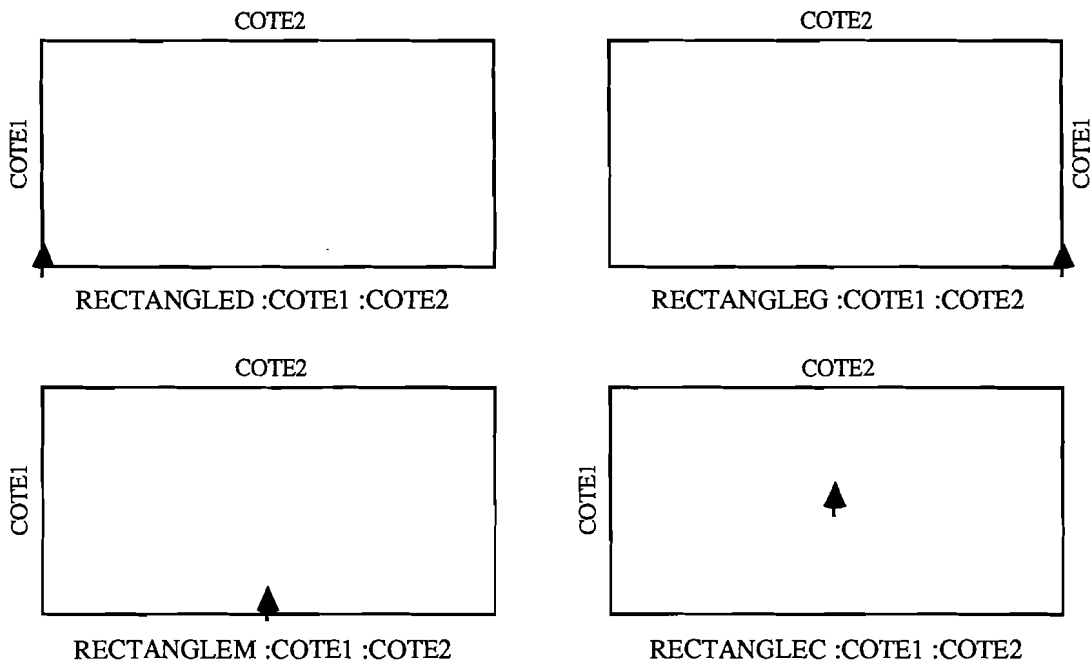
Prérequis.

- A ce niveau, une bonne connaissance de l'éditeur de procédures Logo et de son système d'exploitation et de gestion écran est nécessaire. Si ce n'est pas le cas, il est préférable de revenir un moment à la SITUATION 0 pour exercer cette compétence. Aucune autre compétence n'est indispensable pour profiter des applications proposées.

1. Formes élémentaires.

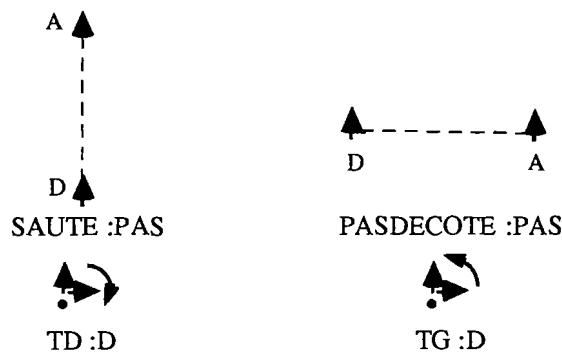
Les commandes sont regroupées en quatre familles de la façon suivante : (a) les formes «tracé» et (b) les formes «déplacement-rotation», (c) les formes «positions» et (d) les outils de gestion de l'écran. Les commandes sont, soit des commandes Logo classiques (avec ou sans restriction d'usage), soit des commandes conçues à partir de ces mêmes primitives.

a. FORME «TRACE».

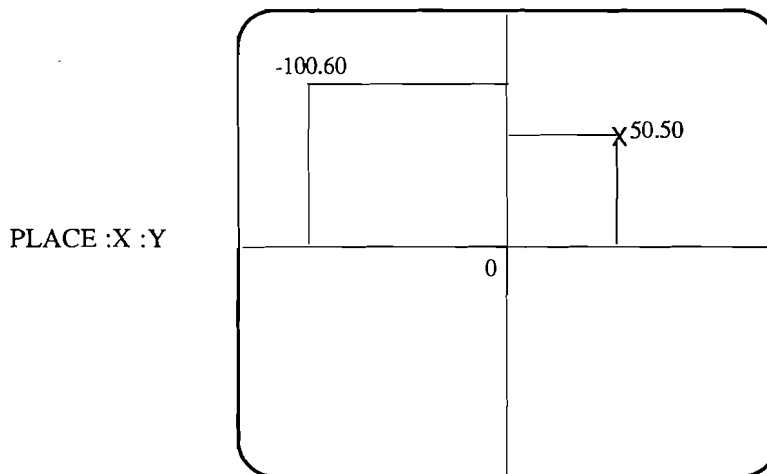


Attention : les formes «tracé» sont déterminées par deux paramètres. L'ordre de ces valeurs est important car la première valeur est, quelque soit l'orientation de la tortue, toujours la dimension du côté du rectangle qui est dans l'axe de la tortue. Le second paramètre correspond au côté perpendiculaire à l'axe de la tortue

b. FORMES «DEPLACEMENT-ROTATION».



c. LES FORMES «POSITION».



d. LES COMMANDES DE «GESTION-ECRAN».

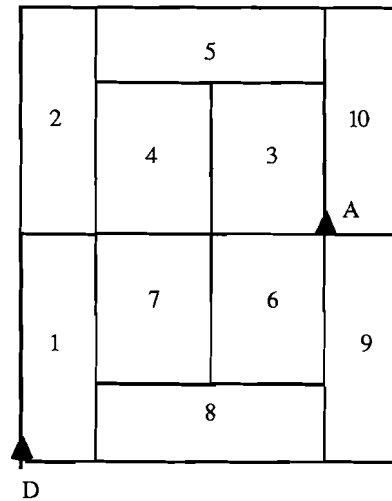
| | | |
|--------------|------|---|
| DEBUT | | Initialise l'écran |
| VIDEcran | (VE) | Efface l'écran graphique. |
| VIDETEXTE | (VT) | Efface la page texte. |
| PLEINEcran | (PE) | Pleine page graphique |
| MIXEcran | (ME) | Page graphique plus quatre lignes de texte. |
| MONTRETORTUE | (MT) | Fait apparaître la tortue. |

2. Dessiner un terrain de tennis avec les rectangles.

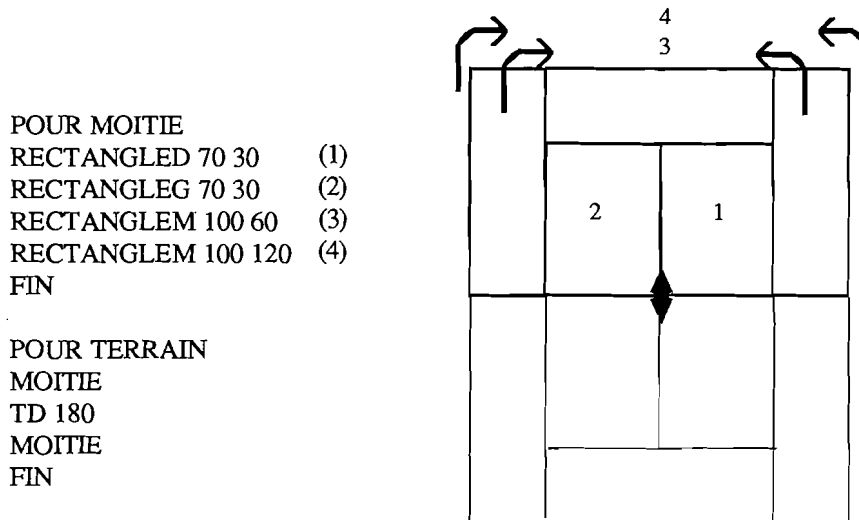
Ce terrain de sport prête bien à une décomposition de ses sous-parties en RECTANGLEx.. Plusieurs familles de stratégies sont possibles.

a. Le terrain est analysé comme un puzzle formé de rectangles de dimensions variables. Le programme est alors une suite de RECTANGLEx.. juxtaposés (sans superposition de formes) et la tortue se déplace entre chaque tracé. Le programme peut être assez «touffu» et peu lisible.

POUR TENNIS
 POSITION -60 -100
 RECTANGLED 100 30 SAUTE 100 (1)
 RECTANGLED 100 30 (2)
 PASDECOTE 60 RECTANGLED 70 30 (3)
 RECTANGLEG 70 30 SAUTE 70 (4)
 RECTANGLEM 30 60 SAUTE -140 (5)
 RECTANGLED 70 30 (6)
 RECTANGLEG 70 30 (7)
 TD 180 RECTANGLEM 30 60 (8)
 SAUTE 30 TD 180
 PASDECOTE 30
 RECTANGLED 100 30 (9)
 SAUTE 100 RECTANGLED 100 30 (10)
 FIN



b. Le terrain est conçu comme une figure géométrique symétrique, l'axe de symétrie pouvant être le filet qui partage le tracé en deux parties identiques. Un premier module est alors construit, puis répliqué par rotation de 180° . Il paraît judicieux de bien choisir le point de départ pour éviter des déplacements inutiles de la tortue. Pour la même raison, on peut aussi inciter les élèves à superposer les rectangles. Dans l'exemple ci-dessous, la tortue est au centre du terrain.



POUR MOITIE
 RECTANGLED 70 30 (1)
 RECTANGLEG 70 30 (2)
 RECTANGLEM 100 60 (3)
 RECTANGLEM 100 120 (4)
 FIN

POUR TERRAIN
 MOITIE
 TD 180
 MOITIE
 FIN

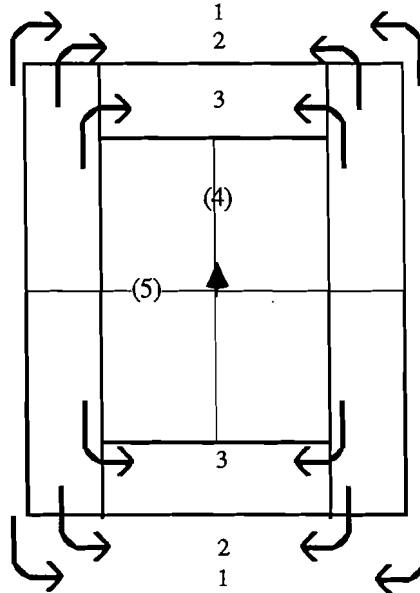
c. Le terrain est un ensemble de RECTANGLEC tracés à partir du centre de l'écran. La tortue n'a alors pas besoin de se déplacer, ni de subir de rotations. Cette solution, très économique et élégante, ouvre de nombreuses possibilités d'applications. La clef du problème réside dans la découverte du tracé de la droite à l'aide du formalisme RECTANGLEx.. En effet, RECTANGLEC 0 100 est une «droite» perpendiculaire à l'axe de la tortue et RECTANGLEC 100 0 est une «droite» dans l'axe de la tortue.

La procédure qui découle de cette analyse, a plusieurs propriétés remarquables qui seront très utiles par la suite :

- elle est composée uniquement de RECTANGLEC ;
- les seuls éléments qui varient, sont les valeurs des paramètres de chaque RECTANGLEC ;
- les différentes composantes du programme peuvent être disposées dans n'importe quel ordre. On obtient toujours le même terrain de tennis.

POUR TERRAIN.DE.TENNIS

| | |
|--------------------|-----|
| RECTANGLEC 200 120 | (1) |
| RECTANGLEC 200 60 | (2) |
| RECTANGLEC 140 60 | (3) |
| RECTANGLEC 140 0 | (4) |
| RECTANGLEC 0 120 | (5) |
| FIN | |

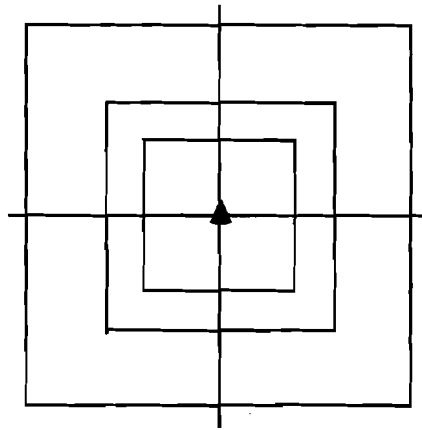


En termes plus techniques, on peut dire que cette procédure est une **description** de l'objet (le terrain de tennis est composé de l'ensemble non-ordonné de RECTANGLEC affectés de valeurs particulières) plus qu'une **prescription** qui correspondrait à une démarche séquentielle comme en programmation classique. Les conséquences théoriques de cette analyse ne sont pas négligeables. Outre qu'elle illustre bien, à notre avis, la distinction entre deux formes de connaissances : connaissances déclaratives versus connaissances procédurales (savoirs opposés à savoir-faire), cette opposition permet de franchir une étape importante dans l'acquisition de structures de contrôle puissantes en programmation. En effet, une procédure ayant ces propriétés, ensemble non-ordonné de formes élémentaires identiques mais dont les valeurs des paramètres sont variables, permet l'utilisation de **boucles itératives** dont le but est de répéter une série de commandes identiques assortie d'une fonction faisant varier les paramètres de cette forme. Le problème de la description du terrain de tennis se réduit alors à la recherche d'une fonction sur des valeurs numériques. Cette question sera abordée dans le thème sur l'évolution.

3. Prolongements.

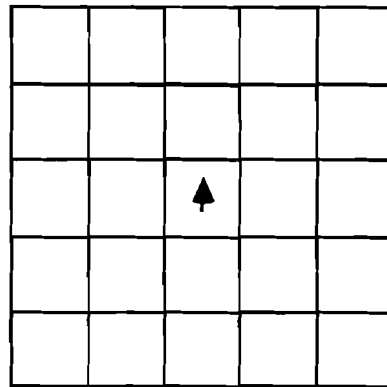
Dans une perspective analogue, on peut travailler sur d'autres figures. Les deux exemples suivants peuvent être traités de la même manière que le terrain de tennis. Une solution au problème est proposée en regard du dessin.

POUR MIRE
 RECTANGLEC 40 40
 RECTANGLEC 80 80
 RECTANGLEC 120 120
 RECTANGLEC 0 140
 RECTANGLEC 140 0
 FIN



LA MIRE

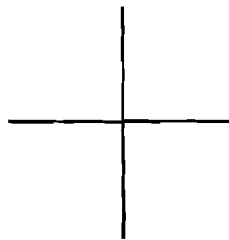
POUR QUADRILLAGE
 RECTANGLEC 100 100
 RECTANGLEC 100 60
 RECTANGLEC 100 20
 RECTANGLEC 60 100
 RECTANGLEC 20 100
 FIN



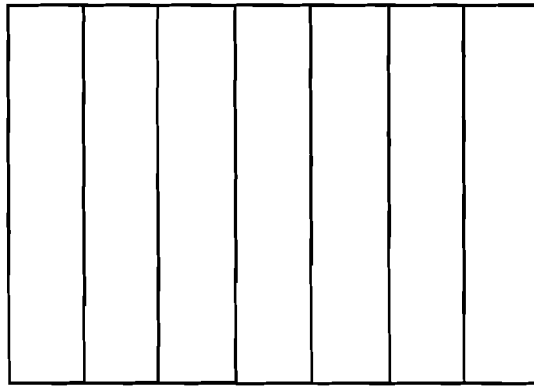
LE QUADRILLAGE

Les projets que nous venons de décrire (TENNIS, MIRE, QUADRILLAGE) ont une caractéristique commune, difficile à formaliser, qui permet de ramener leur description à un ensemble de formes identiques. Du point de vue de cette description, ils constituent une classe d'équivalence : «celle des figures qui se réduisent à un ensemble de RECTANGLEC». On peut ainsi explorer cette nouvelle façon de décrire le réel et faire des rapprochements insoupçonnés. Appartient aussi à cette classe, des objets bien connus.

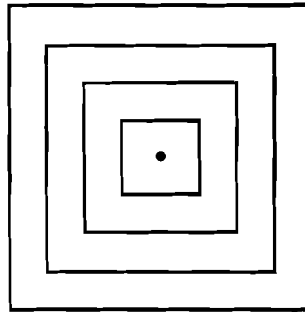
POUR CROIX
 RECTANGLEC 100 0
 RECTANGLEC 0 100
 FIN



POUR GRILLE
 RECTANGLEC 100 0
 RECTANGLEC 100 20
 RECTANGLEC 100 40
 RECTANGLEC 100 60
 RECTANGLEC 100 80
 RECTANGLEC 100 100
 FIN



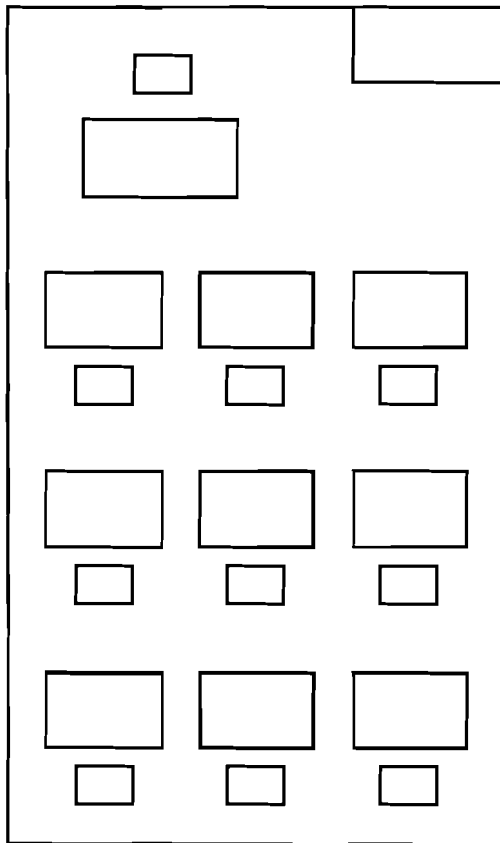
POUR TUNNEL
 RECTANGLEC 0 0
 RECTANGLEC 20 20
 RECTANGLEC 40 40
 RECTANGLEC 60 60
 RECTANGLEC 80 80
 FIN



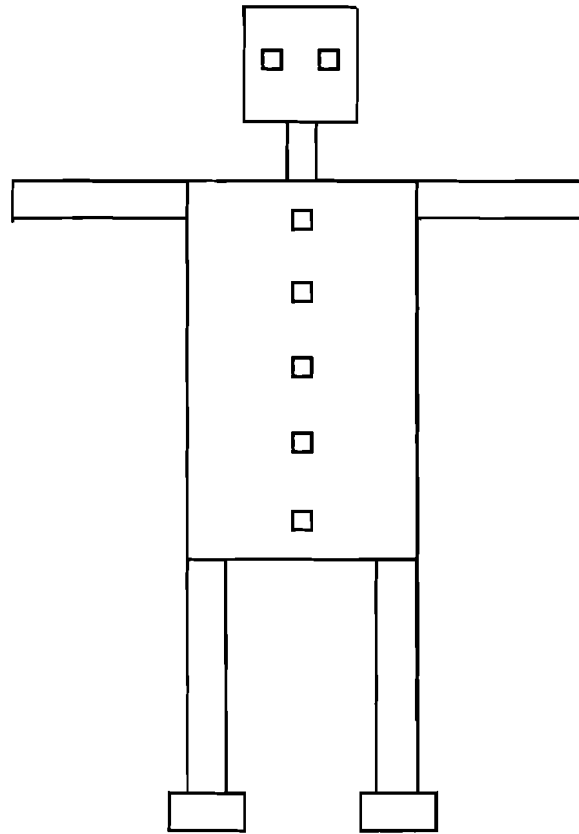
4. Figures non-symétriques.

Il existe aussi, à côté de ces figures régulières, un grand nombre d'objets dont la description est facilitée par le formalisme des RECTANGLEx.. associé aux déplacements de la tortue (SAUTE :PAS et PASDECOTE :PAS). Nous présentons, dans les pages suivantes, quelques-unes des figures sur lesquelles les élèves ont travaillé. L'intérêt didactique de ces descriptions recoupe celui que nous avons décrit dans les SITUATION 0 et SITUATION 1, à savoir :

- analyse préalable des caractéristiques métrique et topologique de l'objet ;
- Mise en relation de ces propriétés avec des structures de contrôle pertinentes du point de vue de la programmation (modularité et répétition).



LE PLAN DE LA CLASSE



LE ROBOT

5. Applications à des contenus «scolaires».

Cet intérêt didactique, lié à la programmation structurée, peut être associé aussi à des activités plus «scolaires». Les deux exemples suivants nous ont été inspirés par un souci de mieux impliquer les connaissances, manipulées par les enfants à l'école, à leur analyse passée par le filtre des structures de contrôle acquises en programmation. L'un fait référence au problème des surfaces et des recouvrements par des «unités» arbitraires (division), l'autre est associé au problème de la métrique et des intervalles.

Problème n° 1 : Recouvrir un mur de papier peint. Le mur est un RECTANGLED de taille variable, assorti éventuellement de fenêtres et/ou de porte. Le papier peint se présente sous la forme de rouleaux d'une largeur donnée. La taille d'un lé, représenté par un RECTANGLED, est déterminée par la forme de la pièce (hauteur) et par la dimension du rouleau. De nombreuses variantes sont imaginables, elles peuvent faire appel progressivement à la notion de division euclidienne (quotient exact et reste), à la notion de surface (un RECTANGLE est une aire, alors que la primitive AVANCE est une longueur)...

POUR RECOUVRIR

REPETE 3 [RECTANGLED 120 40 PASDECOTE 40]

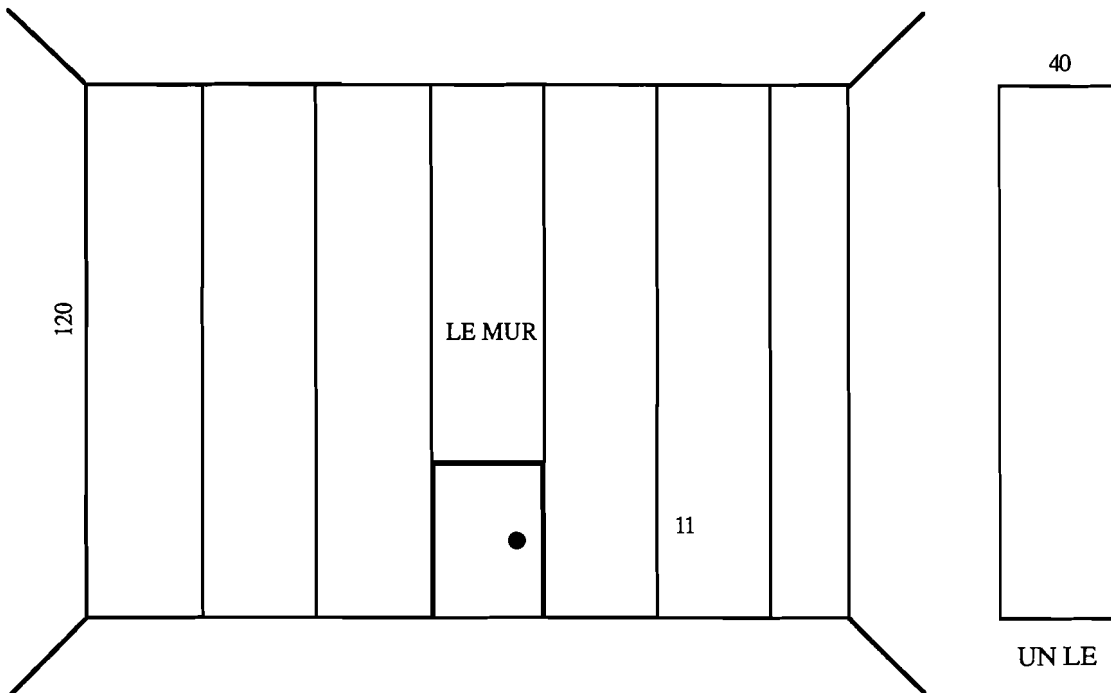
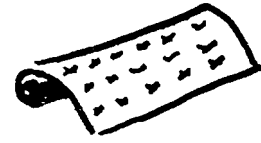
SAUTE 50

RECTANGLED 70 40 PASDECOTE 40

SAUTE -50

REPETE 2 [RECTANGLED 120 40 PASDECOTE 40]

FIN



Problème n° 2 : Construire une règle graduée en respectant certaines contraintes d'intervalles.



UNE REGLE GRADUEE

Plusieurs solutions sont possibles. Les unes (1) privilégient une analyse modulaire, fabrication d'un morceau de règle et répétition de cette unité, les autres (2) reposent sur une analyse des différentes fréquences de barres : grandes barres tous les 10 intervalles, barres moyennes tous les 5... Chaque barre peut être considérée comme un RECTANGLEM x 0 et chaque intervalle comme un PASDECOTE.

1. POUR MORCEAU

RECTANGLEM 15 0
 REPETE 5 [PASDECOTE 5 RECTANGLEM 5 0]
 RECTANGLEM 10 0
 REPETE 5 [PASDECOTE 5 RECTANGLEM 5 0]
 RECTANGLEM 15 0
 FIN

POUR REGLE

PASDECOTE -100
 REPETE 4 [MORCEAU]
 RECTANGLEG 0 200
 FIN

2. POUR REGLES

RECTANGLEC 0 200
 PASDECOTE -100
 REPETE 40 [RECTANGLEG 5 0 PASDECOTE 5]
 PASDECOTE -200
 REPETE 8 [RECTANGLEG 10 0 PASDECOTE 25]
 PASDECOTE -200
 REPETE 5 [RECTANGLEG 15 0 PASDECOTE 50]
 FIN

6. Macroprocédures utilisées dans la situation 2.

TG n primitive LOGO
 TD n primitive LOGO

POUR RECTANGLEC :COTE1 :COTE2
 LC GA 90 AV :COTE2 / 2 DR 90 RE :COTE1 / 2 BC
 REPETE 2 [AV :COTE1 DR 90 AV :COTE2 DR 90]
 LC DR 90 AV :COTE2 / 2 GA 90 AV :COTE1 / 2 BC
 FIN

POUR RECTANGLEG :COTE1 :COTE2
 REPETE 2 [AV :COTE1 GA 90 AV :COTE2 GA 90]
 FIN

POUR RECTANGLED :COTE1 :COTE2
 REPETE 2 [AV :COTE1 DR 90 AV :COTE2 DR 90]
 FIN

POUR RECTANGLEM :COTE1 :COTE2
 GA 90 AV :COTE2 / 2 DR 90
 REPETE 2 [AV :COTE1 DR 90 AV :COTE2 DR 90]
 DR 90 AV :COTE2 / 2 GA 90
 FIN

POUR SAUTE :N
 LC AV :N BC
 FIN

POUR PASDECOTE :N
 LC TD 90 AV :N TG 90 BC
 FIN

POUR PLACE :X :Y
 LC FPOS LISTE :X :Y BC
 FIN

POUR DEBUT
 FIXECHELLE 1.02
 FENETRE
 VE
 VT
 FIN

CONCLUSION

Dans la conception de Papert, LOGO est construit pour permettre à l'enfant un auto-apprentissage de la programmation avec un minimum d'intervention de la part de l'adulte. Dans cette situation, de nombreux auteurs (Papert et al. 1979 ; Hillel 1984 ; Mendelsohn 1984, 1988...) constatent que spontanément les enfants utilisent largement la technique «essais et erreurs» et, travaillent en mode pilotage. Chaque instruction est exécutée et l'activité se résume à un contrôle de la tortue graphique pas à pas. La rencontre de l'enfant avec d'autres instructions que les primitives de déplacement et d'orientation, comme l'instruction REPETE, entraîne chez lui un changement de stratégie. Il devient alors moins sensible à un projet défini à l'avance et il utilise de manière moins contrôlée la puissance de la machine pour tenter des expériences où le contrôle de l'action passe d'un guidage pas à pas, aux systèmes de contrôle inclus dans le système.

Toujours en situation de libre apprentissage, on peut constater que l'enfant tient peu compte de paramètres fondamentaux en programmation LOGO graphique comme,

1) la prise en compte de l'état de la tortue entre sous-séquences ;

2) l'indépendance entre la définition d'une sous-procédure (assimilable à un sous-programme) et son utilisation dans un contexte différent comme par exemple un programme principal ou une autre procédure.

Cette observation traduit le fait que l'enfant ne reconnaît pas dans une suite d'instructions qui dessine un carré, un invariant qui permet de le tracer dans n'importe quelle position. On retrouve ici un résultat classique de la psychologie génétique : un carré posé sur une pointe n'est plus un carré, et pourtant en situation de programmation la liste d'instructions matérialise l'invariant de la figure à travers les actions qui déterminent son tracé.

Autre difficultés, celles inhérentes à la manipulation du contenu des objets programmés. Par exemple, les figures dessinées avec la tortue graphique révèlent de nombreux problèmes liés à l'enseignement de la géométrie en même temps qu'elles font l'objet d'un apprentissage des concepts fondamentaux de la programmation. On peut citer en vrac : la notion de rotation, la paramétrisation des angles en degrés, les problèmes de symétrie, les «théorèmes de géométrie tortue» (Rouchier 1986) que découvre l'enfant dans la construction de figures régulières comme les polygones etc.

De nombreux auteurs ont souligné récemment les limites d'une telle approche de la programmation (Pea et Kurlan 1984, Hillel 1984, Mendelshon 1988...) qui reposent essentiellement sur l'hypothèse, maintenant très largement répandue sans être pour

autant validée empiriquement, qu'en apprenant à programmer, l'enfant apprend plus que de la manipulation de concepts informatiques mais aussi des aptitudes à planifier, des heuristiques, une manière de raisonner...

A l'autre extrémité, on trouve les défenseurs de la tradition béhavioriste classique pour qui tout apprentissage n'est, en caricaturant, qu'une simple accumulation de compétences sans liens entre elles. Pour ceux-là, l'apprentissage d'un code de programmation se résume à apprendre un vocabulaire accompagné de ses règles d'écriture. Cette activité est considérée, dans cette perspective, comme une technique sans intérêt pour des non-professionnels.

Le projet que nous avons proposé se situe dans une perspective intermédiaire. Notre conception de l'apprentissage de la programmation nous amène à concevoir cet apprentissage comme l'acquisition d'expertises locales, dont le but est de permettre une représentation des connaissances qui induit l'extraction de nouvelles propriétés. L'enfant apprend à décrire le réel à partir d'un nouveau système de référence, la pertinence de cette description est un élément essentiel de la réussite de cette entreprise. C'est sur ce dernier point que la responsabilité des enseignants est entière.

REFERENCES.

BOURBION M., (1987) : *Le choix LOGO*, Armand Colin, Paris.

HOC J.M., MENDELSON P., (1987) Les langages informatiques dans l'enseignement, *n° spécial de Psychologie Française*, déc. 87, 32-4.

MENDELSON P., (1988), Les activités de programmation chez l'enfant : le point de vue de la psychologie cognitive, *Technique et Science Informatiques*, Vol. 7 n°1, 47-58.