irem
de GRENOBLE

LOGO POUR LES PETITS ET LES GRANDS

Rédigé par :
R. DE GRAEVE
E. MARTINELLI

Université Joseph Fourier Grenoble 1

La colaboration d'Elise Martinelli
à ce travail a été réalisée dans le cadre
des moyens donnés à l'I.R.E.M. par la
M.A.F.P.E.N. de Grenoble

Le groupe ELEM de l'Irem de grenoble présente dans cette brochure le travail fait avec des élèves de différentes classes sur l'utilisation du langage Logo à l'école.

Nous avons pensé que certains enseignants aimeraient approfondir leur propre connaissance du langage Logo. Nous leur proposons donc dans une deuxième partie d'examiner de plus près les cercles de la tortue et de découvrir avec nous les mystères de la récursivité.

Nous remercions pour leur participation à notre réflexion les institutrices et les instituteurs dans les classes desquels nous avons travaillé.

Mesdames Chevrot, Croquette, Doussan, Lavagne
Ecole Clémenceau. Grenoble

Madame M. Brenner
SES du Collège Charles Munch. Grenoble

Madame Bouquerelle
Ecole Elysée Chatin. Grenoble

Monsieur Glénat, Madame Vallier
Ecole Barnave. St Egrève

Messieurs Duranton, Garcia, Montagnié,
Mesdames Gourmelin, Perrin, Leyssieux,Chuzel
Ecole H.Barbusse. St Martin D'Hères

Mesdames Jusot, PellouxPrayer, Monsieur Salon
Ecole Clos-Marchand.

Animateurs de l'IREM de Grenoble ayant participé aux travaux du groupe Elem.

Henri-Claude ARGOUD
Christian BARTH
Renée DE GREAVE
Raymond GUINET
Elise MARTINELLI
Liliane MONACI
Isidore MORYUSSEF

La frappe de ce document a été réalisée par Annie BICAIS

SOMMAIRE

LOGO POUR LES PETITS

LOGO POUR LES GRANDS

LE LANGAGE LOGO

HISTORIQUE DU LANGAGE LOGO

A l'origine du langage Logo, on trouve Seymour Papert, chercheur au Laboratoire d'Intelligence Artificielle du M.I.T (Massachusetts Institut of Technology).

Travaillant à une théorie générale de l'intelligence, avec pour objectif initial de fabriquer des machines douées de pensée, les chercheurs en Intelligence Artificielle ont eu, de tout temps, une stratégie menant de front l'étude de la manière dont pensent les humains et celle dont les ordinateurs pourraient penser.

Papert, quant à lui, s'intéresse plus particulièrement à la génèse de l'intelligence chez l'enfant. Ayant travaillé plusieurs années à Genève, avec Jean Piaget, et fortement influencé par les théories constructivistes de celui-ci, il forme le projet de fournir aux enfants un environnement informatique comportant des "objets-pour-penser-avec". Il part en effet du principe que l'enfant, à partir de l'interaction qu'il peut entretenir avec son milieu, est le propre bâtisseur de ses connaissances. Papert propose donc d'enrichir ce milieu et de favoriser cette interaction grâce à l'ordinateur.

Dès 1967, il entame avec ses collaborateurs la mise au point d'un langage informatique qui conviendrait aux enfants. LOGO est le nom qu'ils choisissent pour désigner à la fois une conception de la pédagogie et la famille de langages de programmation allant de pair avec elle.

Papert a longuement développé dans son livre "Jaillissement de l'esprit" (Edition Flammarion) ses théories sur l'apprentissage et le rôle que peut y jouer l'ordinateur :

"A l'heure actuelle, dans l'enseignement, quand les enfants sont mis en présence de l'ordinateur, c'est presque toujours pour les mettre à l'épreuve, pour leur faire effectuer des exercices d'un niveau de difficultés approprié, pour leur fournir de l'information. C'est l'ordinateur qui programme l'enfant, ni plus ni moins. Dans l'environnement Logo, la situation est renversée: c'est l'enfant qui maîtrise la machine; il programme l'ordinateur".

Pour rendre les sujets de programmation plus accessibles à des enfants, on invente alors un animal cybernétique, commandé par l'ordinateur, baptisé Tortue.

Au départ celle-ci n'a d'autre fonction que d'être bonne à programmer, LOGO étant le langage qui permet de communiquer avec elle. Certaines Tortues sont des êtres abstraits qui vivent sur des écrans d'ordinateurs, d'autres sont des sortes de jouets mécaniques se déplaçant sur le sol.

Elles sont munies d'un crayon qui permet de matérialiser la trace de leurs trajets. C'est ce que Papert appelle un de ces "objets-pour-penser-avec". Par la suite, la tortue deviendra un instrument très utile pour manipuler des objets géométriques familièrement regroupés sous le nom de Géométrie Tortue.

On appelle LOGO-TORTUE, ou LOGO-GRAPHIQUE le sous-ensemble LOGO contenant les instructions Tortue ; il existe d'autres domaines de programmation en LOGO (celui des mots et listes par exemple) mais la voie d'accès au langage la plus fréquente reste le LOGO-TORTUE.

La famille des langages LOGO (issue du LISP) se caractérise en particulier par la présence de définitions procédurales, avec des variables locales pour permettre la récursivité. C'est ainsi qu'avec LOGO il est possible de définir de nouvelles instructions et de nouvelles fonctions qui pourront être utilisées exactement comme les éléments primitifs du langage. Une autre de ses caractéristiques est d'être un langage applicatif. Il a par ailleurs une structure de liste complète, c'est-à-dire que le langage peut opérer sur des listes dont les membres sont eux-mêmes des listes, ou bien des listes de listes.

On voit par là que LOGO est loin d'être un simple jouet, réservé à l'usage des enfants. Il est d'ailleurs de plus en plus souvent utilisé à l'Université dans l'enseignement de la programmation.

Signalons cependant que le langage LOGO exige une mémoire plus spacieuse que des langages moins puissants, comme le BASIC. Jusqu'à une date récente, on ne le trouvait implanté que sur des ordinateurs de taille relativement respectable et de ce fait plus coûteux, alors que dans le même temps le langage Basic était le seul à être implanté sur tout les micro-ordinateurs. Les ordinateurs Apple ont été les premiers à disposer en France du langage LOGO. Depuis, la baisse de coûts et l'évolution très rapide du matériel informatique ont permis une meilleure diffusion du langage.

(En raison de ces difficultés techniques d'implantation, beaucoup d'enseignants qui ont fait leurs premières armes avec le BASIC gardent à ce langage un certain attachement et la polémique entre les tenants de chaque langage n'est pas tout à fait éteinte. Celle-ci n'a pas lieu d'être, ces deux langages n'ayant pas été conçus pour le même usage).

En 1970, débute au M.I.T une première expérimentation, avec des enfants, du langage LOGO. Celle-ci est jugée suffisamment encourageante pour être poursuivie avec des enfants de tous âges. Le projet LOGO est toujours en expérimentation aux Etats-Unis et a été repris par plusieurs équipes.

Les travaux des chercheurs américains sont connus en France depuis 1973. Mais c'est seulement à partir de 1980 qu'une première version du langage est disponible en France. Depuis cette date, les recherches et expériences se sont multipliées.

Il n'est pas facile de faire un bilan exhaustif de celles-ci. Comme dans toute recherche de ce type, il est illusoire, voir périlleux de prétendre mesurer avec rigueur les apports spécifiques du travail fait en Logo avec des enfants, surtout si on en attend autre chose que de simples savoir-faire.

Dans l'ensemble, tous les compte-rendus reflètent une certaine satisfaction de la part des auteurs mais peu de conclusions définitives. On pourrait dire qu'à l'heure actuelle, l'utilisation de l'outil LOGO intéresse beaucoup plus pour les questions qu'il ouvre sur notre enseignement et son adéquation aux processus d'apprentissage que pour les réponses qu'il apporte.

LES CARACTERISTIQUES DU LOGO-GRAPHIQUE

A. Les primitives.

La tortue est représentée sur l'écran par une flèche qui indique sa position et son orientation ; nous dirons que ces deux caractères définissent son **état**.

Les primitives essentielles du langage sont :

AVANCE n	(ou bien AV n)
RECULE n	(ou bien RE n)
TOURNEDROITE n	(ou bien TD n)
TOURNEGAUCHE n	(on bien TG n)

(On notera que chacune de ces primitives est désignée par un verbe à l'impératif. Ce sera toujours le cas pour les commandes du langage, qui sont assimilées ainsi à des actions, ceci permettant de rendre plus aisé l'accès à la syntaxe du langage en le rendant plus proche du langage naturel).

Chacune d'elles agit en transformant l'état de la tortue. En ce sens, on peut dire qu'elles jouent le rôle de transformations géométriques :

- AV et RE provoquent une translation de la tortue, la direction de cette translation étant définie par l'orientation initiale de la tortue.

- TG et TD provoquent une rotation de la tortue,

le centre de la rotation étant la position initiale de la tortue, l'angle étant indiqué en degré.

- Une suite d'actions provoque une succession de transformations dont le résultat est un changement d'état de la tortue: ce changement d'état, dû à la composition de ces transformations, est donc soit une rotation soit une translation.

Ces primitives permettent donc de décrire les déplacements de la tortue en termes de mouvements relatifs, chaque déplacement de la tortue est déterminé par deux choses: la primitive utilisée mais encore l'état de la tortue à l'instant où la primitive va agir.

On comprend dès lors la contrainte très forte qui s'impose dans la succession des actions et la nécessité, au moment de l'élaboration d'un programme, de connaître à chaque étape l'état de la tortue et d'anticiper sur cet état.

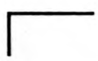
B. Leur utilisation.

Il y a dans la pratique du logo-graphique une contradiction native: alors que les primitives décrites ci-dessus ne s'expriment qu'en termes de déplacements de la tortue, on cherche, le plus souvent, à obtenir non pas un simple déplacement de celle-ci mais plutôt à faire apparaître sur l'écran un dessin. Or le dessin n'est à l'origine qu'une sorte d'effet secondaire: la trace laissée par le «crayon-baissé».

Très vite, c'est ce dessin qui devient l'effet dominant pour l'utilisateur et on a tendance à oublier qu'il est toujours associé à un changement d'état de la tortue et qu'il en dépend même étroitement. Il résulte de cet oubli des erreurs d'analyse caractéristiques, très fréquentes chez les débutants, dont voici deux exemples.

1 - un escalier

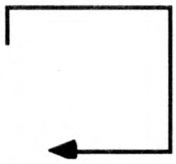
ESCALIER

On repère assez naturellement sur cette figure le motif: 

Ce motif est produit par la suite d'actions :

AV 10 TD 90 AV 20

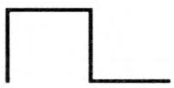
et l'on croit obtenir la figure ESCALIER en reproduisant trois fois ce motif.
Voici ce qui arrive assez souvent :



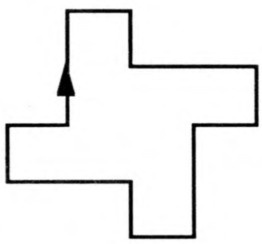
REPETE 3 [AV 10 TD 90 AV 20]

On n'a pas été assez attentif à l'orientation de la tortue à l'issue de la première séquence.

2 - des créneaux

On peut isoler le motif 

Celui-ci s'obtient par la suite d'actions:
AV 10 TD 90 AV 10 TD 90 AV 10 TG 90 AV 10
et on conclut un peu trop rapidement:
REPETE 5 [AV 10 TD 90 AV 10 TD 90 AV 10 TG 90 AV 10]
ce qui provoque des surprises quant au résultat :

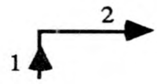


Ces exemples nous montrent que le tracé seul n'est pas suffisant pour rendre compte d'une suite d'actions puisqu'il ne figure pas le changement d'état de la tortue; nous avons donc voulu insister sur le fait qu'une suite d'actions a deux effets : un tracé **et** un changement d'état de la tortue.

Nous avons choisi d'en donner une représentation fléchée c'est-à-dire un tracé sur lequel on repère l'état initial de la tortue par une flèche 1 et son état final une flèche 2.

Par exemple :

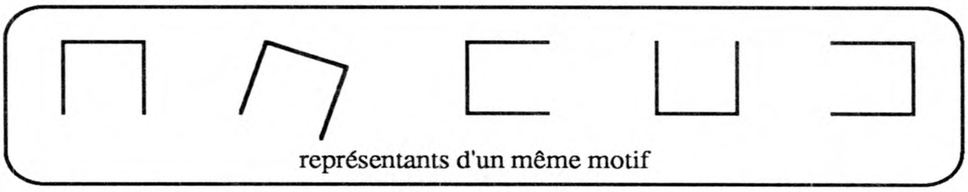
AV 10 TD 90 AV 20 aura comme représentation fléchée :



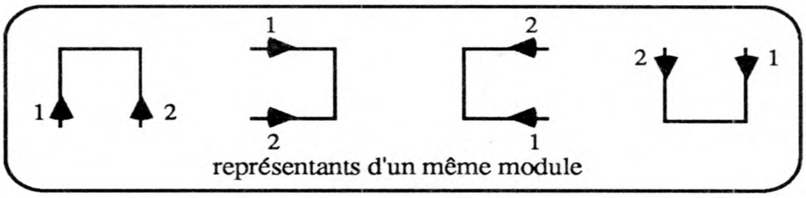
Par ailleurs, une suite d'actions ne produit pas toujours le même tracé : si l'on modifie l'état initial de la tortue, une même suite d'actions produit un nouveau tracé qui sera l'image du précédent par un déplacement (rotation, translation, ou rotation-translation).

Dans ce qui suit, nous appellerons :

- **motif** : la classe des tracés qui se correspondent par un déplacement.

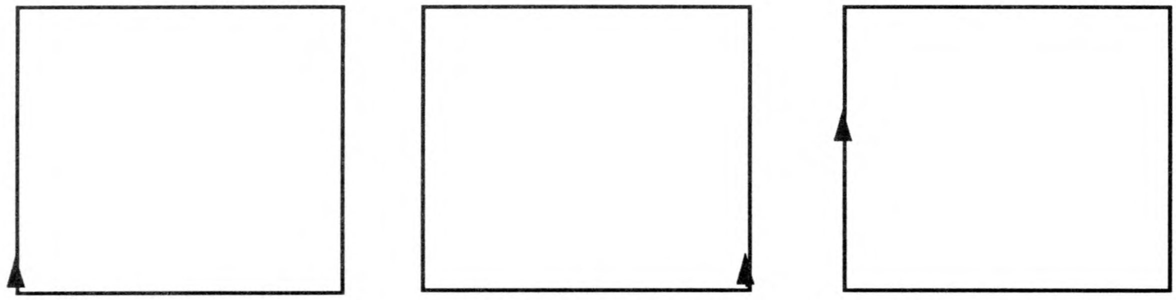

représentants d'un même motif

- **module** : la classe des représentations fléchées qui se correspondent par un déplacement.



représentants d'un même module

Remarque : il est clair qu'un même motif peut être obtenu par des suites d'actions différentes, mais les modules seront également différents. Prenons l'exemple bien connu du carré. Voici trois modules différents pour le motif "carré".



REPETE 4 [AV 20 TD 90] REPETE 4 [AV 20 TG 90] REPETE 4 [AV 10 TD 90 AV 10]

Quelques modules possibles

C. Les procédures.

Le langage Logo permet de donner un nom à une suite d'actions. On dispose alors d'une procédure.

Cette possibilité de nommer une suite d'actions pour les regrouper dans une procédure donne un type de langage «modulaire» particulièrement favorable à la pratique d'une programmation structurée.

Lorsqu'une procédure a été définie, elle s'intègre au langage au même titre qu'une primitive et peut être utilisée comme telle pour définir de nouvelles procédures. On a coutume de dire de Logo que c'est un langage «évolutif», c'est-à-dire qu'à partir d'un nombre assez restreint de primitives, il est possible, pour chaque utilisateur, d'enrichir le langage d'un lot de procédures définies pour son usage.

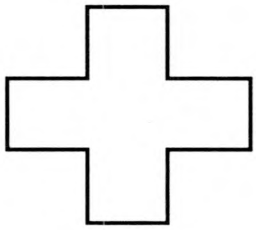
(Cette faculté du langage est intéressante pour le travail avec des enfants : l'enseignant peut fournir aux élèves ce que l'on appelle des macro-primitives afin d'élargir leurs possibilités de programmation. Cf. Grand IN n° 39).

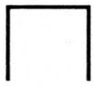
Si on utilise la terminologie décrite plus haut, on voit aisément qu'une procédure est associée à un module, c'est-à-dire que son exécution provoque un tracé et un changement d'état de la tortue. Sur le module, ce changement d'état est figuré par les flèches 1 et 2.

En modifiant l'état initial de la tortue, avant de faire exécuter la procédure, on peut obtenir, à partir de la même procédure, les différents tracés d'un même motif, c'est-à-dire ses images par rotation ou translation. (Dans ce qui suit, pour plus de commodité, nous utiliserons le mot procédure à la place du terme module).

Pour illustrer ce qui précède, nous allons analyser les propriétés géométriques d'une figure et montrer comment ces propriétés transparaissent dans la façon d'écrire une procédure.

La croix de pharmacie



On peut repérer sur cette figure le motif 

On reconnaît alors quatre tracés de ce même motif. 

(Bien sûr, bien d'autres découpages de cette figure seraient possibles).

Chacun de ces tracés se déduit du précédent par une rotation de centre O et d'angle (-90).

On peut aussi remarquer que le deuxième tracé se déduit du premier par une translation de vecteur $\overrightarrow{AB}$ suivie d'une rotation de centre B et d'angle -90, le troisième tracé se déduit du deuxième par une translation BC suivie d'une rotation de centre C et d'angle -90, etc.Ceci provient des égalités entre les transformations suivantes :

$$\begin{aligned} \mathrm{Rot}(B, -90) * t(AB) &= \mathrm{Rot}(C, -90) * T(BC) = \mathrm{Rot}(D, -90) * t(CD) \\ &= \mathrm{Rot}(A, -90) * t(DA) = \mathrm{Rot}(0, -90) \end{aligned}$$

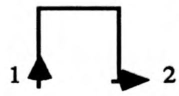
Nous appellerons R cette transformation.

Nous allons donner deux façons d'écrire une procédure Logo correspondant à cette analyse de la figure.

Méthode 1

Pour obtenir le motif repéré sur la figure, on peut utiliser la procédure :

```
POUR PONT1
AV 10 TD 90 AV 10
TD 90 AV 10 TG 90
FIN
```



Cette procédure provoque un tracé et un changement d'état de la tortue figuré ici par la transformation de la flèche 1 en flèche 2. On voit que cette transformation est égale à la transformation R.

Une première exécution de la procédure produit un premier tracé et un changement d'état de la tortue. Une nouvelle exécution de la procédure produira un autre tracé puisque l'état initial n'est plus le même.

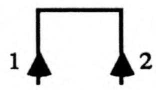
Pour obtenir le dessin complet il suffit d'écrire :

```
POUR CROIX1
REPETE 4 [ PONT1 ]
FIN
```

Méthode 2.

Une autre possibilité serait d'utiliser la procédure suivante :

```
POUR PONT2
AV 10 TD 90 AV 10
TG 90 RE 10
FIN
```



Cette procédure provoque le tracé du même motif mais le changement d'état de la tortue correspond ici à une translation. On ne retrouve pas la transformation R qui définit le passage d'un tracé au suivant. Il sera ici nécessaire, pour passer d'un tracé au suivant, de modifier l'état de la tortue avant le prochain appel de la procédure PONT2 :

```
POUR CROIX2
REPETE 4  [ PONT2 TD 90 ]
FIN
```

CONCLUSION.

L'examen attentif des caractéristiques du Logo graphique nous montre que sous des dehors «enfantins» celui-ci gère des propriétés géométriques assez complexes : orientation, notions de forme, transformations, composition de déplacements, etc.

Il n'est pas nécessaire pour l'utilisateur de maîtriser ces notions théoriques pour pouvoir aborder ce langage.

Par contre, l'analyse d'une figure incite à repérer des motifs et leurs transformés, la coordination des procédures entre elles nécessite une certaine perception de ces transformations.

Ainsi ces propriétés seront implicitement mise en œuvre dès qu'on voudra réaliser une figure un tant soit peu complexe.

PROGRAMMER POUR QUOI FAIRE ?

NOS OBJECTIFS

Programmer, ou plus généralement, planifier une tâche exige la mise en œuvre de compétences fondamentales : être capable d'anticiper des actions, de contrôler le déroulement de ces actions, être capable de manipuler des objets formels et d'opérer sur ces objets.

Il est évident que de telles compétences sont très importantes dans le développement cognitif des enfants et dépassent largement le cadre de l'informatique.

Dans notre travail à l'école élémentaire, nous avons fait l'hypothèse que la pratique de la programmation, à l'aide d'un langage approprié, serait avant tout favorable à l'amélioration de ces compétences.

Le langage LOGO, issu des recherches sur l'intelligence, offre la possibilité d'une programmation structurée, proche de notre mode de raisonnement ; il nous a paru particulièrement bien adapté à cet objectif. De plus, le micro-monde de la tortue logo étant essentiellement graphique, le thème des activités de programmation peut être choisi dans le domaine géométrique et ceci nous a donné l'occasion de progresser en même temps dans ce champ disciplinaire.

Nous avons donc travaillé avec les enfants dans une double perspective :
- programmer en utilisant une démarche algorithmique ;
- approfondir et enrichir leurs connaissances géométriques.

Dans ce qui suit, nous allons distinguer ces deux points de vue, étant bien entendu que pendant les activités réalisées en classe, ils pourront difficilement être dissociés.

I - LA DEMARCHE ALGORITHMIQUE.

L'activité de programmation nécessite la mise en œuvre de ce qu'il est fréquent d'appeler une démarche algorithmique. Il est devenu courant de mettre en parallèle cette démarche avec celle de la résolution de problème et c'est pourquoi nous avons souhaité exercer nos élèves à cette démarche.

Rappelons brièvement ce qui caractérise une démarche algorithmique et qui la rapproche de la situation de résolution de problèmes.

Pour résoudre un problème ou accomplir une tâche, un temps soit peu complexe, il sera presque toujours nécessaire de passer par une succession d'étapes :

- analyse du problème ou de la tâche ;
- décomposition de cette tâche en plusieurs parties (mise en évidence de «modules» ou «unités de sens») ;
- définition ou construction de ces différentes parties comme des entités indépendantes (ou modules) ;
- assemblage et coordination de ces parties entre elles.

Le plus souvent, la réalisation de ces différentes étapes se fait de manière implicite et ne passe pas toujours par une prise de conscience ; elle est donc, de ce fait, difficilement améliorable. Par contre, dans une activité utilisant l'ordinateur, il devient indispensable d'expliciter ces étapes pour programmer l'ordinateur à l'exécution de la tâche.

Dans notre travail en classe, nous avons constamment tiré parti de cette nécessité inhérente à l'utilisation de l'ordinateur. C'est-à-dire que, par le biais des explications à donner à la tortue-logo «qui ne sait faire que ce qu'on lui dit», les enfants ont été amenés à exprimer en langage formalisé les différentes étapes de chaque problème proposé.

Ce travail d'explicitation a été l'occasion d'un approfondissement pour améliorer la démarche choisie ; en effet, pour la plupart des exemples traités en classe le but proposé n'était pas seulement d'obtenir la solution du problème, ici une figure sur l'écran, mais nous nous sommes attachés également à l'organisation et à la structure du programme écrit pour y parvenir.

Ainsi, nous avons souvent demandé aux enfants d'écrire plusieurs programmes pour un même problème, afin de comparer les étapes définies pour chacun d'eux et de choisir le découpage le plus efficace.

Nous pensons que cette pratique devrait favoriser un comportement similaire des enfants face à des situations-problèmes rencontrées dans de tout autre domaine, en les incitant à mettre en évidence, là aussi, des «modules» à traiter de manière indépendante, avant de parvenir à la résolution du problème. (Cf. Recherche INRP «apprentissage à la résolution de problèmes»).

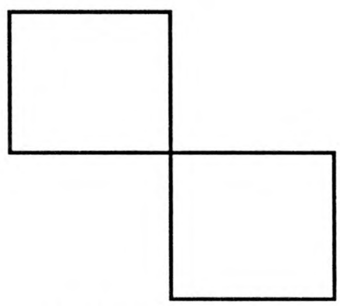
Le langage logo, qui autorise la définition de procédures et leur emboîtement se prête particulièrement bien à la description des différentes étapes de la démarche algorithmique puisque chaque module peut être réalisé de manière indépendante par l'écriture d'une procédure, ces procédures pouvant ensuite être assemblées dans le programme principal.

II - NOMMER UNE PROCEDURE.

Dans le cours de notre travail, nous avons pu vérifier que le fait de déclarer une procédure, c'est-à-dire de donner un nom à un ensemble d'actions, pour en faire une «unité de sens», n'était pas chose simple. En effet, une procédure étant définie, il est possible de lui faire exécuter sur l'écran différents dessins en modifiant seulement l'état initial de la tortue (voir Caractéristiques du langage) ; ainsi, une même procédure permet de générer en fait une «classe» d'objets, et son «nom» a donc valeur de concept, ici un concept de forme.

Si dès le départ, les enfants se font un jeu de la possibilité offerte de choisir arbitrairement le nom de la procédure et si les propositions les plus variées fleurissent, par contre la capacité à réinvestir le nom de la procédure pour construire un objet plus complexe est loin d'être immédiate et se fait avec une certaine réticence de la part de beaucoup d'enfants comme on peut le voir dans les exemples qui suivent.

Les enfants doivent faire exécuter par la tortue, le dessin suivant :



CARRE DOUBLE

Voici une solution assez caractéristique de celles proposées par beaucoup d'élèves dans des cas analogues

```
POUR CARRE
REPETE 4  [ AV 20 TD 90 ]
FIN

POUR CARREDOUBLE
CARRE TD 90 AV 20
REPETE 4 [ AV 20 TD 90 ]
FIN
```

Ici, les choses se passent comme si l'enfant acceptait d'utiliser le nom de la procédure CARRE dans une situation, mais ne retrouve pas l'usage de ce même nom dès que la situation diffère un temps soit peu. On peut interpréter cette attitude comme une difficulté d'accéder au concept de forme.

Autre exemple, il faut réaliser le dessin suivant :

CRENEAU

Voici la solution proposée par Anne-Marie

```
POUR A
REPETE 2 [ AV 10 TD 90]
FIN

POUR B
REPETE 2 [ AV 10 TG 90]
FIN

POUR CRENEAU
A B A B A B A B A B A B A B
FIN
```

On voit ici que l'enfant a commencé par définir et construire deux modules A et B selon une analyse précise, en utilisant sans difficulté la primitive REPETE.

Mais au moment d'intégrer ces modules dans le programme final, elle revient à une description linéaire, beaucoup plus sommaire, au lieu d'utiliser de nouveau la primitive REPETE en écrivant REPETE 7 [A B]

Apparemment, le degré de complexité devenait trop grand puisque qu'elle voyait, sans doute, dans les noms A et B non pas un seul objet mais une suite d'actions, ce qui conduisait à l'emboîtement de deux primitives REPETE :

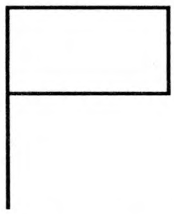
```
REPETE 7 [ REPETE 2 [ AV 10 TD 90]  REPETE 2 [ AV 10 TG 90] ]
```

De nombreuses réactions de ce type nous ont confirmé le fait qu'il y avait là ce que l'on peut peut-être appeler un obstacle didactique (compétence des enfants à utiliser un «nom» de procédure) et qu'il était tout à fait utile de les faire progresser dans ce sens étant bien entendu qu'une telle compétence est significative de progrès dans le développement de la pensée formelle.

Pour terminer, un clin d'œil d'un enfant, sans doute étonné par notre insistance à «couper les procédures en quatre».

```
POUR DRA
AV 40
FIN

POUR PEAU
REPETE 2 (AV 30 TD 90 AV 50 TD 90)
FIN
```



DRAPEAU

POUR DRAPEAU
DRA PEAU
FIN

La démarche algorithmique ne fait aucun doute !

III - LES OBJECTIFS GEOMETRIQUES.

Au cours de notre travail avec les enfants, nous avons presque toujours proposé comme situation problème de départ la réalisation d'une figure ; l'analyse de la tâche était donc essentiellement d'ordre géométrique et mettait en œuvre les connaissances des enfants dans ce domaine.

Une telle analyse repose bien sûr, sur la construction préalable d'un modèle et il est bien évident que cette construction ne peut se faire qu'en fonction des représentations que l'enfant a déjà de l'objet à construire. Notre objectif est donc de partir de ces représentations plus ou moins implicites pour les enrichir et les consolider.

Ceçi est vrai dans le cas de figures géométriques déjà rencontrées à l'école élémentaire, comme les polygones et le cercle, par exemple. Un élève qui écrit une procédure pour dessiner un carré, ne découvre pas bien sûr à cette occasion ce qu'est un carré mais il va être obligé d'en donner une description formelle qui devrait affiner et préciser sa propre représentation mentale.

Le cas du cercle est particulièrement significatif de ce processus : si les élèves savent très bien reconnaître un cercle, éventuellement le construire avec un compas, il est rare qu'ils puissent exprimer au cours des activités géométriques usuelles de l'école les diverses conceptions possibles du cercle. Or, en se mettant à la place de la tortue-logo, qui parcourt le cercle, les enfants proposent de la faire «avancer un peu, tourner un peu, avancer un peu, tourner un peu...» ce qui utilise une définition du cercle tout à fait différente de celle de la construction au compas.

(Le cercle n'est pas ici l'ensemble des points équidistants du centre mais est plutôt défini comme une courbe de courbure constante).

Il n'est pas question de faire apprendre aux enfants de telles définitions mais on voit ici comment ils peuvent en avoir une approche intuitive tout en diversifiant leur propre mode de représentation pour le rendre opérationnel.

Les notions géométriques abordées pendant notre travail ont été très diverses et découlent assez naturellement des caractéristiques du langage :

Orientation dans le plan (utilisation de TD et TG).

Notion d'angle (cette notion est le plus souvent rencontrée à l'école élémentaire au travers des secteurs angulaires et sera ici enrichie par l'aspect rotation).

Construction de polygones réguliers, valeur des angles intérieurs, extérieurs.

Repérage dans le plan.

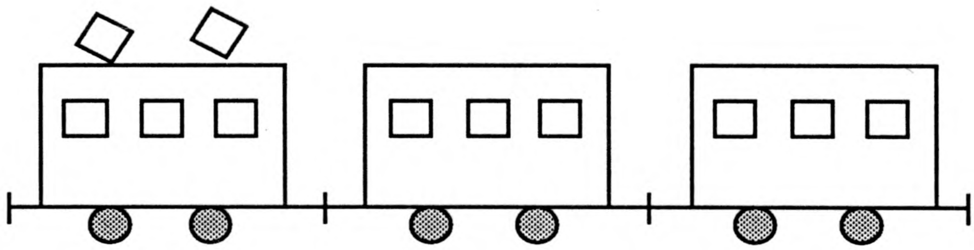
Transformations géométriques : rotation, translation, symétrie, homothétie.

Nous distinguerons, selon le type de figures proposées aux élèves, deux situations qui ne font pas appel aux mêmes compétences.

- Lorsqu'on propose, par exemple, comme figure aux enfants, une maison, l'analyse du problème peut conduire à la décomposition : mur, toit, porte. Ici, la décomposition en modules, relève d'une analyse purement figurative et fonctionnelle de l'objet et ne fait pas intervenir d'abstraction géométrique. Par ailleurs, la description de ces modules sous forme d'une procédure indépendante n'apporte pas toujours de simplification dans l'écriture du programme et peut même s'avérer contraignante au moment de la coordination des procédures entre elles (l'état de la tortue à la fin de chaque procédure n'est pas toujours le plus adapté à l'enchaînement avec la suivante).

Il est cependant utile de proposer aux enfants ce type de figures : cela permet de leur faire découvrir, dans une situation simple, sans grande difficulté d'ordre géométrique, la possibilité de donner un nom à chacune des parties et d'utiliser ce nom dans le programme final. Elles permettent également de découvrir la nécessité d'anticiper sur la position de la tortue pour coordonner les procédures entre elles.

Le projet ci-dessous réalisé par des élèves de Valence, relève de ce type de situation. Il s'agit de réaliser un dessin assez complexe, en l'occurrence un train avec ses wagons et sa locomotive ; le découpage en plusieurs parties est justifié par le besoin de partager la tâche entre différents groupes d'élèves qui devront construire, chacun, des modules indépendants. Dans cet exemple la décomposition du train en diverses parties est plutôt induite par la fonction de l'objet et plus rarement par des propriétés géométriques. Chaque groupe d'élèves devra cependant prendre soin de «livrer» son module sous forme d'une procédure en indiquant les positions initiale et finale de la tortue, pour rendre possible la coordination finale).



- La deuxième situation relève plutôt de la reconnaissance de forme et de l'utilisation des transformations géométriques.

Elle sera provoquée par des figures comportant des propriétés géométriques facilement repérables, par exemple des figures contenant un motif soumis à des déplacements.

L'enfant devra donc être capable de reconnaître ce motif, au travers de ces transformations et le reproduire au moyen des changements d'états de la tortue.

Nous avons voulu exercer les élèves à analyser une figure et les entraîner à la «voir» de différentes façons.

Nous faisons l'hypothèse que, chez les enfants, la manipulation de ces objets géométriques et l'utilisation, même implicite, de leurs propriétés devraient enrichir leur champ d'expérience et favoriser la formation de certaines images mentales, ces images mentales étant un préalable indispensable à l'émergence des concepts correspondants.

Pour parler autrement, disons que le jour où ces enfants, devenus grands, auront à faire de savants calculs sur des matrices de transformations ponctuelles, par exemple, peut-être que le souvenir lointain d'une petite tortue sur un écran pourra leur venir en aide !

EXEMPLES D'ACTIVITES A L'ECOLE ELEMENTAIRE

*** Que sait faire la tortue ?**

*** Apprendre un nouveau mot à la tortue**

*** Où la tortue travaille beaucoup**

*** Bon noël**

*** Des petits robots**

*** La route**

*** Décomposition de figures**

*** Des étoiles**

*** Des ailes et des moulins**

*** Les polygones réguliers**

*** Le cercle**

*** Le parallélogramme**

*** Papier à réseau triangulaire**

*** Des vagues et des fleurs**

*** Des figures pleines**

Introduction

Ce qui suit n'est en aucun cas une progression.

A l'heure actuelle il n'existe pas de textes officiels définissant des apprentissages obligatoires à l'école dans un langage de programmation, bien heureusement !

L'activité de programmation reste encore à l'école élémentaire un lieu de recherche et de créativité pour les enfants, où chacun doit pouvoir progresser à son rythme, sans les contraintes habituelles de contrôle d'acquisitions et d'évaluation. Nous n'avons qu'une crainte, c'est que des enseignants bien intentionnés se sentent tenus de suivre une progression impérative. Il n'en est rien.

Le langage LOGO a le grand avantage de ne nécessiter qu'un très petit nombre d'apprentissages préalables indispensables. Ces préalables une fois dépassés, chacun est libre de sa démarche, en fonction des objectifs spécifiques choisis pour chaque séance.

Il est indispensable de consacrer une ou deux séances à la découverte et à la pratique des quatre primitives AV, RE ,TD, TG, ces quatre mots constituant le vocabulaire de base de la tortue.

L'étape obligée suivante est la description et la mise en œuvre du mode procédure, avec l'utilisation indispensable de l'**éditeur.**

Il faut ensuite décrire aux élèves la possibilité de répétition offerte par la primitive REPETE .Il est nécessaire d'y passer un temps assez long, car son utilisation est délicate, sous des dehors faussement simples.

Nous conseillons d'aborder tout de suite ces contraintes techniques ; elles sont très facilement comprises par les enfants et sont indispensables pour commencer un travail de programmation proprement dit. Les balbutiements laborieux en mode direct n'apportent pas grand chose si on veut travailler dans le sens des objectifs définis plus haut.

Pour illustrer ceci, nous avons décrit, en détails, dans les trois premiers paragraphes qui suivent, la manière dont nous avons abordé ces trois étapes.

Nous relatons ensuite les exemples de notre travail dans des classes différentes, en signalant succintement les succès et les difficultés les plus notables. L'ordre de présentation est arbitraire et ne relève pas d'une difficulté croissante, il représente plutôt la diversité des approches possibles pour un même objectif et les similitudes relevées dans le comportement des enfants des différentes classes.

Les lecteurs pourront donc choisir à leur guise parmi cet éventail qui n'est pas, bien sûr, exhaustif.

Notre démarche.

Contrairement à certaines pratiques qui ont eu cours à propos du langage LOGO, les enfants n'étaient pas livrés à une découverte totalement improvisée et informe face à la tortue sur son écran. Chaque primitive et sa syntaxe était d'abord présentée par l'enseignant. (Nous n'avont pas voulu reproduire les tâtonnement empiriques qui aboutissaient à des «AVANCE, s'il te plaît » de la part des enfants, au grand attendrissement des expérimentateurs !).

De même, les enfants ne «jouaient» pas avec l'ordinateur ; ils avaient à chaque séance une tâche précise, en l'occurrence un dessin à faire excécuter par la tortue. Par contre, chacun était libre de choisir sa démarche et sa méthode, avec la possibilité à chaque instant de regarder faire la tortue sur l'écran, pour valider ou corriger son travail.

Nous avons pu constater, dans toutes les classes, sans exception, que les enfants adhéraient volontiers à cette contrainte et faisaient preuve d'une persévérance à toute épreuve.

Que sait faire la tortue ?

Dans chaque classe, nous avons consacré une première séance à la découverte des primitives de base.

Elle peut avancer ou reculer ; pour cela il faudra taper au clavier :

AV *suivi d'un nombre* (AV *est l'abrévation de* AVANCE)

RE *suivi d'un nombre* (RE *est l'abréviation de* RECULE)

La première question qui se pose est celle de l'unité . Nous convenons de dire qu'il s'agira du nombre de "pas de tortue".

Les enfants font quelques essais, ce qui donne l'occasion de découvrir les limites verticales de l'écran : AV 100 et RE 99.

Elle peut aussi changer de direction, en utilisant les mots :

TD *suivi d'un nombre* (abréviation de TOURNE A DROITE)

TG *suivi d'un nombre* (abréviation de TOURNE A GAUCHE)

Là encore, il faut savoir ce que doit être le nombre qui suit. Nous avons procédé de différentes façons pour trouver les valeurs de ce nombre.

«Actuellement, sur l'écran, la tortue regarde vers le haut, essayez de la faire aller vers la porte».

Dans certaines classes nous avons essayé de ne pas faire référence à la mesure des angles en degrés, pour deux raisons :

* dans les classes de CM1 les enfants ne connaissaient pas d'unité de mesure pour les angles ;
* en CM2, nous avons préféré qu'ils retrouvent la mesure de l'angle droit, eux-mêmes.

Nous avons pu constater que les enfants ne proposent pas spontanément des mesures comme 100, 50, ou 90, mais essaient des mesures comme 81, 67, 89...(ce qui nous démontre, au passage,tout le travail à faire sur la notion d'encadrement).

Pour finir le nombre 90 a été adopté, comme résultat expérimental !

Dans une autre classe, nous avons choisi de couper court à ces tâtonnements, en définissant tout de suite les commandes TD 90 et TG 90.

De cette manière, le nombre 90 est pris dès le départ comme une sorte d'unité de référence .

le quart de tour de la tortue : 90
le demi-tour : 90 + 90 = 180
le tour complet : 90 + 90 + 90 + 90 = 360.

Ces quatre mots étant définis, nous avons complété cette liste avec :

- VE (abréviation de VIDE ECRAN)

«la tortue efface tout et revient se mettre au centre, la tête en haut».

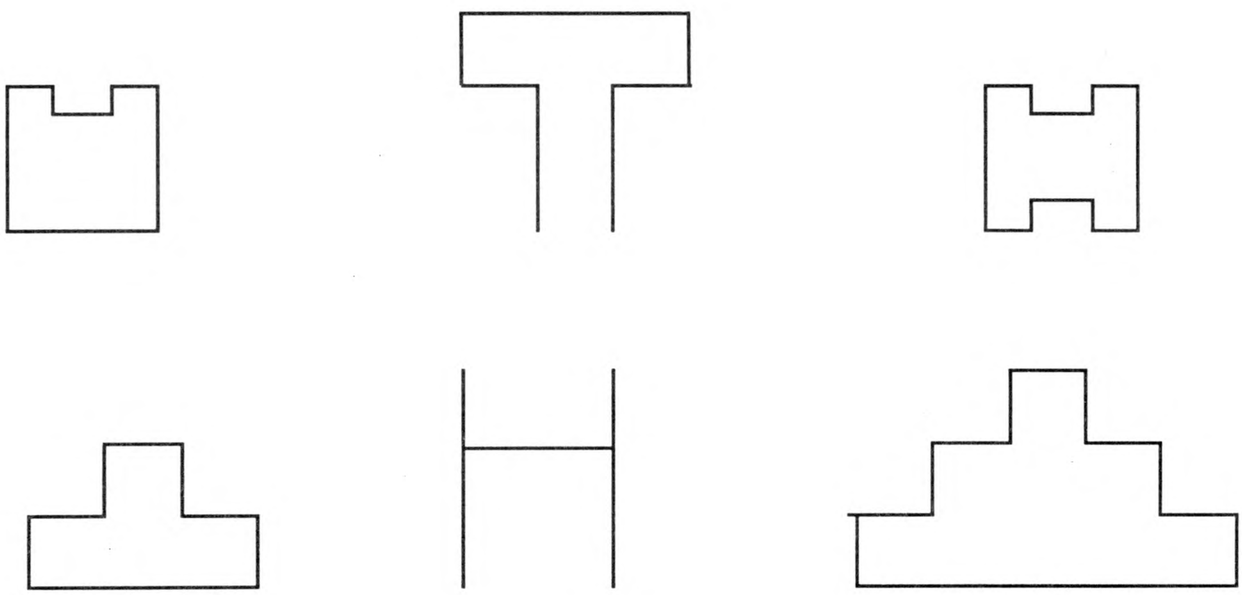
- LC (abréviation de LEVE ton CRAYON)

«la tortue ne laisse pas de trace en se déplaçant».

- BC (abréviation de BAISSE ton CRAYON)

«la tortue retrouve son crayon».

Cette introduction étant faite, les enfants s'entrainent à partir de quelques modèles dessinés au tableau ; ces exemples n'utilisent pour le moment que l'angle droit. Chaque enfant arrive à en réaliser un ou plusieurs, selon sa rapidité.



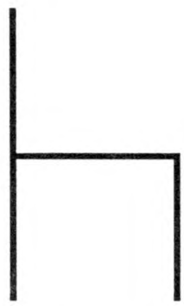
Ce travail et cette première découverte peuvent être avantageusement prolongés en classe, sans ordinateur :

Les enfants utilisent les mêmes commandes pour faire déplacer un de leur camarade en suivant un parcours dans la classe. On insiste alors sur le fait que «droite» et «gauche» sont relatifs à l'enfant qui se déplace et non à celui qui parle. Nous avons pu constater que cette mise au point sur la relativité des directions "droite" et "gauche" était tout à fait utile pour les élèves de CE et souvent même encore pour des élèves de cours moyen.

Apprendre un nouveau mot à la tortue

Dans la plupart des classes, dès la deuxième (ou troisième séance) nous avons jugé plus utile et plus efficace de décrire tout de suite aux enfants la manière d'utiliser l'éditeur.

Nous sommes parti d'un exemple très simple, baptisé CHAISE.



Ce dessin une fois réalisé par tout le monde, on l'efface par VE. On voudrait bien le refaire. Comment ? Courageusement, les enfants proposent de récrire la suite des commandes.

«Ne pourrait-on pas dire à la tortue de refaire la même chose, c'est-à-dire une chaise ?».

«C'est possible, il suffit de lui apprendre un nouveau mot, le mot CHAISE.*Voici comment faire".*

:Nous avons alors donné aux enfants la petite notice suivante.

(Cette notice se rapporte au LOGO Thomson).

Il nous parait très formateur d'entraîner les enfants à se servir le plus souvent possible d'une documentation ; après quelques explications orales supplémentaires, tout le monde parvient à ses fins et dispose d'une première procédure .

Apprendre un nouveau mot à la tortue

1 .

Tu dois d'abord changer de page et passer sur une page de couleur blanche, appelée Editeur
Pour cela, tape **ED** (en terminant par la touche **ENTREE**)

2 .

Si tu veux apprendre à la tortue le mot CHAISE, par exemple, il faut d'abord taper :
POUR CHAISE (en terminant par la touche **ENTREE**)

3 .

Tu expliques alors à la tortue tout ce qu'elle doit faire
(en terminant par la touche **ENTREE**)

4 .

Tu écris ensuite le mot **FIN**
(en terminant par la touche **ENTREE**)

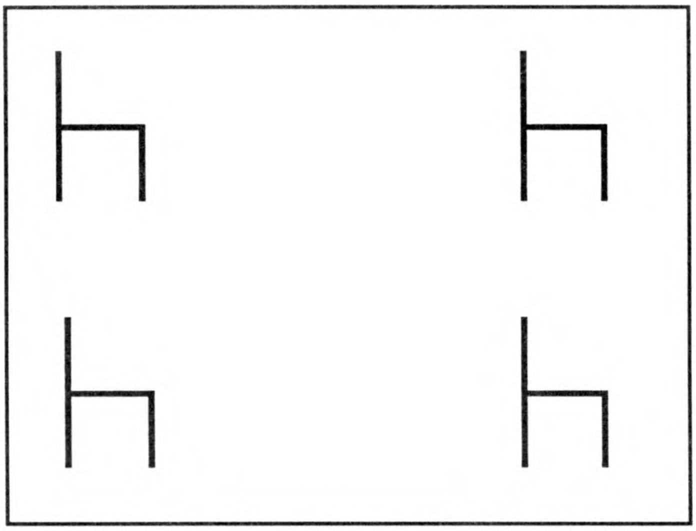
5 .

Tu dois maintenant quitter la page de couleur blanche pour retrouver la tortue.
Pour cela enfonce simultanément les touches **CNT** et **Q**

1	ED	**ENTREE**
2	POUR ------	**ENTREE**
3	------------ ------------	**ENTREE**
4	FIN	**ENTREE**
5	CNT Q	

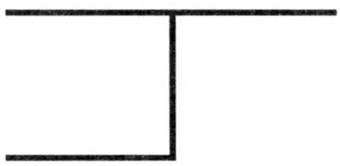
Maintenant CHAISE *est un mot connu de la tortue. Tape* CHAISE *et regarde.*

Dans un deuxième temps, on montre aux enfants comment le nouveau mot peut être utilisé.
«Essayez de dessiner une chaise aux quatre coins de l'écran».

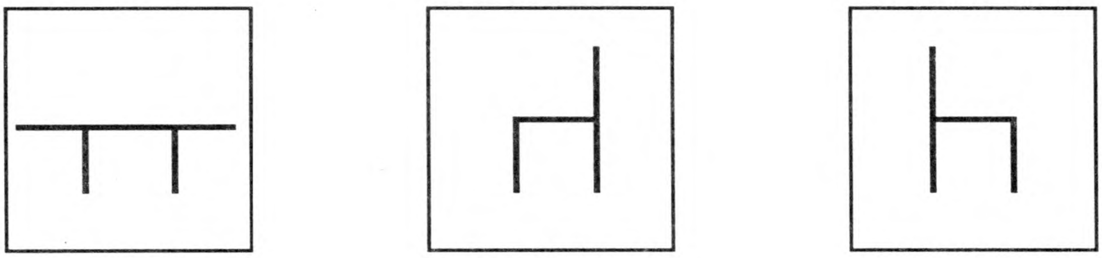


Cet exercice est utile pour faire découvrir aux enfants que le dessin produit par CHAISE n'est pas lié à une position fixée sur l'écran .

De même, on montre que la chaise peut aussi "être tournée". Voici ce qu'on obtient si on tape : TD 90 CHAISE



On s'entraîne ensuite à apprendre d'autres mots à la tortue, en lui faisant dessiner les objets suivants :



On s'accorde pour y voir une table et deux chaises.

On découvre ensuite que la tortue ne peut pas dessiner les deux chaises avec la même suite d'actions; il faudra définir deux noms de procédures différents.

Voici des exemples de descriptions possibles :

```
POUR CHAISE
AV 20 RE 10
TD 90 AV 10
TD 90 AV 10
FIN
```

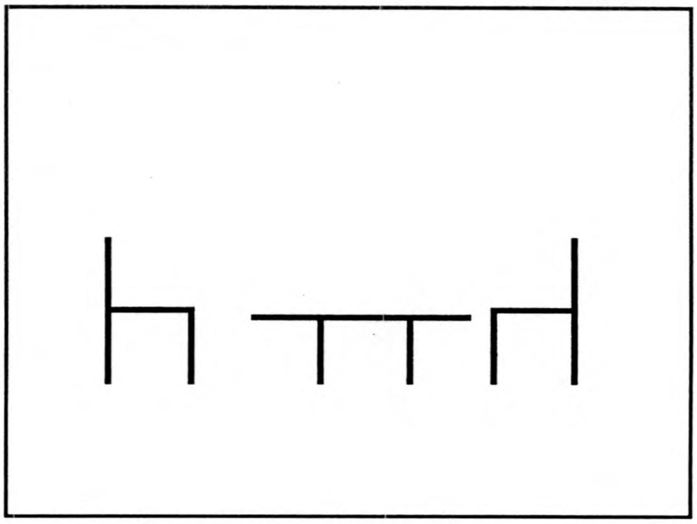
```
POUR FAUTEUIL
AV 20 RE 10
TG 90 AV 10
TG 90 AV 10
FIN
```

Enfin, on montre aux enfants une possibilité fondamentale offerte par le langage tortue.

"Les nouveaux mots du vocabulaire tortue peuvent être utilisés pour en définir un nouveau . Nous allons lui faire dessiner les objets précédents dans une seule figure que nous appellerons SALON, *par exemple".*

Après quelques tâtonnements, les enfants réalisent tous une figure complète. On aboutit à des procédures diverses .Par exemple :

```
POUR SALON
 LC TG 90 AV 80 TD 90
BC  CHAISE
 LC TG 90 AV 20 TG 90
BC TABLE
 LC TG 90 AV 20 TG 90
BC FAUTEUIL
FIN
```



Cet exemple de décomposition d'une figure, très simple à analyser, nous a permis de donner aux enfants, en peu de temps, les règles de base du langage : la définition d'une procédure et la possibilité d'emboîtement de ces procédures.

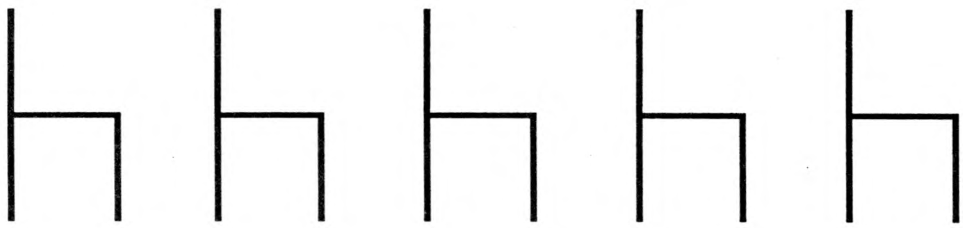
Où la tortue travaille beaucoup

Objectif : Présentation de la primitive REPETE
Son utilisation

Premier exemple

Pour susciter le besoin d'une procédure de répétition, nous avons prolongé l'activité précédente en posant le problème suivant :

«La tortue sait maintenant dessiner une chaise quand on lui dit le mot CHAISE. On voudrait lui faire dessiner une rangée de cinq chaises. Est-ce que ce sera facile ?».



Les enfants proposent alors de répéter plusieurs fois le mot CHAISE.

On leur fournit alors une nouvelle information :

«La tortue comprend le mot "répète" *à condition de l'utiliser d'une façon très précise».*

REPETE nbre [liste d'actions]

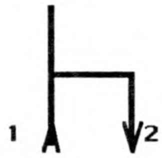
«On indique à la tortue combien de fois elle doit répéter la même chose et on écrit entre crochets tout ce qu'elle doit répéter».

On laisse ensuite les enfants résoudre eux-mêmes le problème de la rangée de chaises.

Bien entendu, on commence par beaucoup d'erreurs du genre :

```
POUR RANGEE
REPETE 5 [CHAISE]
FIN
```

Cette procédure est, bien sûr, inefficace, si on utilise le module CHAISE précédemment étudié.



Chaque groupe d'élèves analyse son erreur et tente de la rectifier avec notre aide éventuelle. On découvre, ou bien on redécouvre qu'il est indispensable de positionner la tortue avant de lui faire dessiner la chaise suivante.

Beaucoup d'enfants écrivent alors :

```
POUR RANGEE
REPETE 5 [CHAISE] LC TG 90 AV 30 TG 90 BC
FIN
```

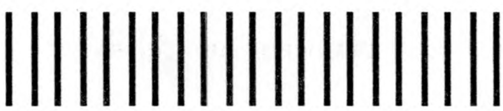
Là encore, déception quant au résultat.

Ce type d'erreur étant très fréquent quand on utilise la primitive REPETE, nous avons passé un certain temps à l'analyser et le corriger avec toute la classe. On aboutit à la conclusion qu'il faut bien mettre entre crochets **toute** la suite d'actions qui doit être répétée.

Les enfants persévèrent et tous les groupes aboutissent finalement à une rangée de chaises parfaitement d'aplomb.

Nous avons relaté un peu longuement cette séance pour insister sur une démarche qui fait l'intérêt essentiel de l'utilisation du langage LOGO : face à un problème posé, il n'y a pas une réponse juste ou fausse validée par le maître, mais au contraire une suite d'essai-erreur au cours de laquelle chaque essai sur l'écran permet à l'enfant de découvrir et d'analyser lui-même son erreur et de progresser vers la solution.

Deuxième exemple



HERBE

Un enfant plein d'imagination croit y voir de l'herbe , pourquoi pas!

Ce second exemple est volontairement très simple du point de vue de l'analyse géométrique ; il est extrêmement facile d'y repérer un motif qui se reproduit. Les enfants connaissent la primitive REPETE, il s'agit surtout de déjouer les difficultés encore présentes dans l'utilisation de cette primitive.

On trouve encore des erreurs du type :

```
POUR HERBE
REPETE 21 [ AV 10 RE 10 ]
FIN
```

Ces erreurs sont découvertes puis corrigées assez facilement par tous.

```
POUR HERBE
REPETE 20  [ AV 10 RE 10 LC TD 90 AV 2 TG 90 BC]
AV 10 RE 10
FIN
```

Certains enfants proposent même d'écrire une procédure intermédiaire, baptisée BRIN, ce qui n'est pas indispensable dans le cas présent, mais montre qu'ils sont capables d'utiliser un nom de procédure.

```
POUR HERBE                                      POUR BRIN
REPETE 20  [ BRIN LC TD 90 AV 2 TG 90 BC ]      AV 10 RE 10
BRIN
FIN                                             FIN
```

Bon... noël

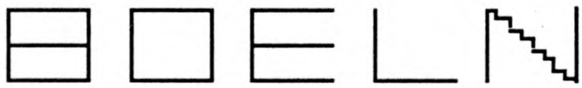
Niveau : **CE - CM**

Objectifs : décomposition d'un problème
coordination de procédures
prise en compte de l'orientation

Durant le mois de décembre nous avons réalisé un petit projet collectif dans une classe de CE_2 et une classe de CM : faire écrire à la tortue BON NOEL.

1ère étape (sans ordinateur).

a) Il a fallu tout d'abord choisir collectivement la graphie de chaque lettre. Ce qui a donné :



La graphie du «N» a posé quelques problèmes : les enfants ont à leur actif environ huit heures de LOGO, ils ne connaissent que l'angle droit et c'est pourquoi la barre du N a été représentée p par un escalier (l'escalier avait été réalisé lors d'une séance précédente).

b) Il faut ensuite déterminer la taille des lettres pour que «BON NOEL» tienne sur l'écran, ce qui nous donne l'occasion de faire avec les enfants un petit travail d'arithmétique.

2ème étape (sans ordinateur).

C'est celle de l'écriture des procédures et donc du choix des modules.

La classe est répartie en 5 groupes. Chaque groupe est chargé d'écrire une lettre. Avant de commencer l'écriture de la procédure, les enfants doivent décider de la position initiale et finale de la tortue.

Voici le déroulement de cette recherche :

Un membre de l'équipe «B» (chargée de réaliser la procédure B) vient au tableau et marque ce qu'il a choisi comme flèche départ et comme flèche arrivée.

Exemple du choix selon les enfants.

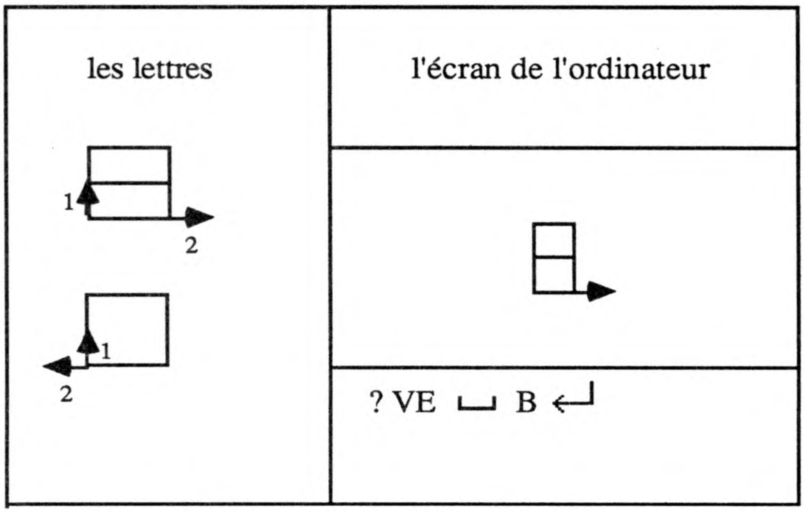
(CE_2) (CM_1 et CM_2)

On remarquera

- que la position de la flèche-1 est toujours une flèche dirigée vers le haut. (Est-ce due à l'influence de la position de la tortue après VE ?) ;
- que chez les «plus grands» la flèche-2 est dirigée selon le sens du crayon lorsqu'on dessine B sur une feuille de papier.

Puis un élève de l'équipe «O» vient marquer ses flèches.

Voici ce qu'on obtient sur le tableau noir de la classe.



Nous disons alors aux enfants : supposons que l'équipe «B» et l'équipe «O» aient fini leur travail, c'est-à-dire aient écrit les procédures B et O réalisant ces deux modules. Si je tape : V E B $\boxed{\text{ENTREE}}$ que va-t-il se passer ?

- «La tortue va dessiner B».
- «Où se trouve maintenant la tortue, selon la flèche-2 du module B?».

Un élève vient marquer en rouge la tortue.

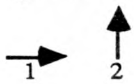
- «Si je veux maintenant que la tortue dessine le O, où la tortue doit-elle se trouver ?».

Un élève vient marquer en vert la position désirée.

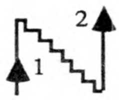
- «Que faut-il faire pour amener la tortue dans la bonne position ?».

Un élève répond :

- «Il faut faire un espace».
- «Oui, nous allons créer une procédure ESPACE. Que fera cette procédure ?».

L'élève qui avait suggéré de faire «un espace» vient au tableau et marque (oh merveille !). 

On décide d'appeler cette procédure ESP.

Puis on continue avec l'équipe «N». L'enfant marque 

«Que faut-il faire entre le O et le N ? Où se trouve la tortue à la fin du O ? Où se trouve la tortue au début du N ?».

Pour que la procédure espace «marche» entre chaque lettre, on s'aperçoit alors qu'il faut que les différents groupes s'entendent entre eux...

Les modules choisis sont alors

On décide que le groupe L écrira aussi la procédure ESP qui réalise l'espace.

3ème étape (avec ordinateur).

Chaque groupe écrit sa procédure et la vérifie sur l'écran.

Quand arrive le moment de l'assemblage final les enfants tapent :

B ESP O ESP N ESP ESP N ESP O ESP E ESP L ENTREE

Tous sont contents au début de voir que «ça marche» mais bien vite ils s'aperçoivent que NOEL est écrit de travers.

Les enfants ne comprennent pas très bien **pourquoi** il y a une erreur mais ils savent où se situe l'erreur.

On cherche à analyser cette erreur en se posant des questions :

«Que fait ESP ?».

«Que fait ESP ESP ?».

Les enfants cherchent les réponses en dessinant les positions successives de la tortue. Ils proposent alors de créer une nouvelle procédure : ESD (espace double) en écrivant :

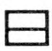
POUR ESD	OU	POUR ESD
ESP		AV 40
TD 90		TD 90
ESP		FIN
FIN		

Par la suite, les enfants s'amusent beaucoup à écrire :

LE BON NOEL, LE BON LEON, LE BON BONBON et ont envie de définir la procédure A pour écrire à la rentrée BONNE ANNEE !

Quelques remarques.

1. Nous n'avons pas cherché à utiliser les propriétés géométriques du tracé des différentes lettres bien que certains enfants se soient aperçus que l'on pouvait faire le

 à partir du  ou du  etc.

2. Dans cette activité l'effort a porté sur le **choix** de la position de départ et de la position d'arrivée, la décomposition en module étant ici évidente (sauf peut-être la procédure ESP). Au départ nous avons laissé les différents groupes libres de leur choix, mais il a fallu ensuite unifier ces choix... même si ce choix n'était pas le meilleur... ce qui a permis d'exploiter l'erreur du «ESP ESP».

3. Cette activité a favorisé le travail en équipes et a contribué à la solidarité des différentes équipes (ce qui n'était pas évident à priori compte tenu de la réaction de certains enfants très individualistes qui prétendaient au début «je ferai toutes les lettres).

Ce partage de la tâche entre plusieurs équipes a permis de montrer la nécessité d'une spécification précise de chaque module, pour que la coordination des résultats soit possible.

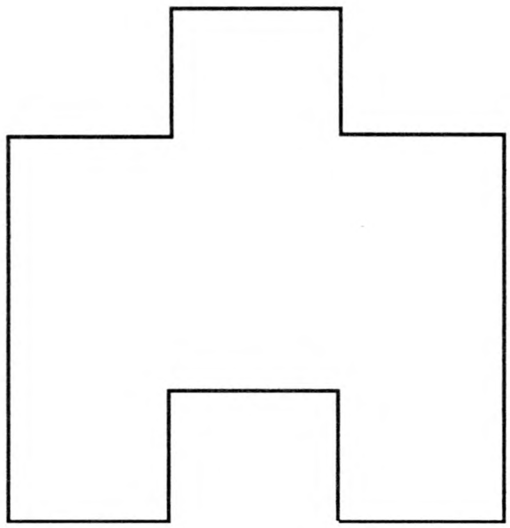
Des petits robots

Niveau : CE2 - CM1

UN ROBOT

Objectifs : Mesurage - choix d'une unité de mesure.
Changement d'unité.
Utilisation de droite et gauche.

On propose aux enfants le dessin suivant, en leur demandant d'écrire une procédure pour le réaliser. On décide d'un commun accord qu'il s'agit d'un robot !



Ce dessin peut être présenté aux élèves sur une feuille unie ou sur du papier quadrillé. Le choix de ce support détermine un travail de mesurage différent pour les enfants.

- papier uni : utilisation d'un double décimètre pour mesurer le modèle.
- papier quadrillé : il suffit de compter les carreaux, le support servant ainsi d'instrument de mesure.

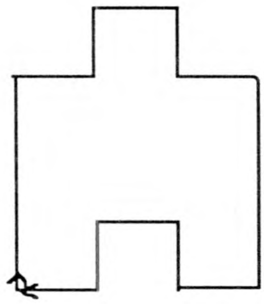
Dans les deux cas, il faudra établir une correspondance entre les mesures faites sur le modèle et l'unité de mesure sur l'écran, c'est-à-dire le "pas de tortue".

Le plus simple est de choisir 10 pas de tortue pour représenter 1 centimètre ou un carreau, mais pourquoi pas 8 pas de tortue pour un carreau, si on veut provoquer un petit exercice de calcul mental.

La recherche d'une procédure pour obtenir le dessin ne va pas sans mal. En effet les problèmes d'orientation et d'utilisation des commandes TD et TG ne sont pas résolus d'emblée par nombre d'enfants, surtout lorsque la tortue a la tête en bas.

Après plusieurs essais, qui font progresser les choses, on obtient une majorité de réussites. Voici, par exemple, une des procédures obtenues (et le module correspondant).

```
POUR ROBOT
AV 40 TD 90 AV 10 TG 90
AV 10 TD 90 AV 10 TD 90
AV 10 TG 90 AV 10 TD 90
AV 40 TD 90 AV 10 TD 90
AV 10 TG 90 AV 10 TG 90
AV 10 TD 90 AV 10
FIN
```



Il est possible de poursuivre cette activité par un travail sur la proportionnalité : on pourrait demander aux élèves de modifier leur procédure pour agrandir ou réduire le modèle proposé.

DES ROBOTS

Objectifs : Utilisation d'un nom de procédure.
Coordination entre appels de procédure.
Arithmétique : problème ouvert.

On a pu constater auparavant que la dimension horizontale de l'écran est de 320 pas de tortue. On propose alors aux enfants le problème suivant :

"Essayez d'aligner horizontalement et bien régulièrement, le plus possible de petits robots. Bien sûr, chaque robot doit rester entier".

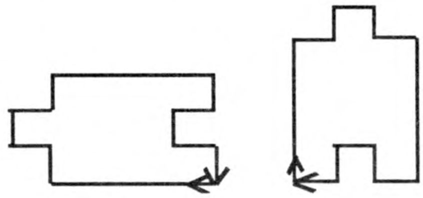
Les enfants sont alors confrontés à deux difficultés de type différent :

- difficulté géométrique : quel est l'état de la tortue lorsqu'elle a fini de dessiner un robot ? Comment la déplacer pour démarrer le robot suivant ?

- difficulté arithmétique : combien peut-on dessiner de robots au maximum ? quel écartement choisir entre chaque robot ? Où placer le début et la fin de la ligne ?

Pour éviter une telle complexité, il vaut mieux procéder en deux étapes

- On pourra, dans un premier temps, s'essayer à placer deux robots côte à côte, afin de trouver remède à des surprises du genre :



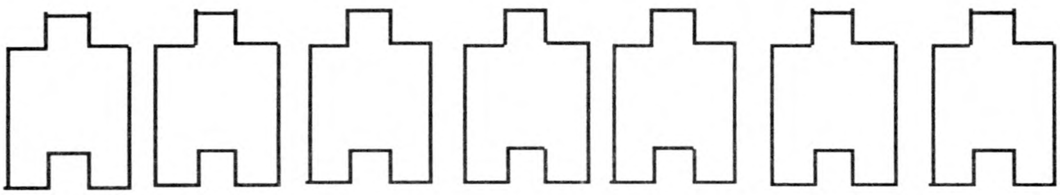
ROBOT LC AV 20 BC ROBOT

- Dans un deuxième temps, une fois réglée la difficulté géométrique, on pourra faire une recherche sur la question du nombre maximum de robots et leur écartement.

Ce problème est très ouvert et autorise des modes de résolutions très divers.

Il est bon de laisser chaque groupe à son tâtonnement, mais beaucoup d'enseignants retrouveront là une vieille connaissance : le traditionnel problème d'intervalles !

Cette recherche pourra bien sûr se prolonger en séance de mathématiques, le travail en Logo servant essentiellement de point de départ et de validation.



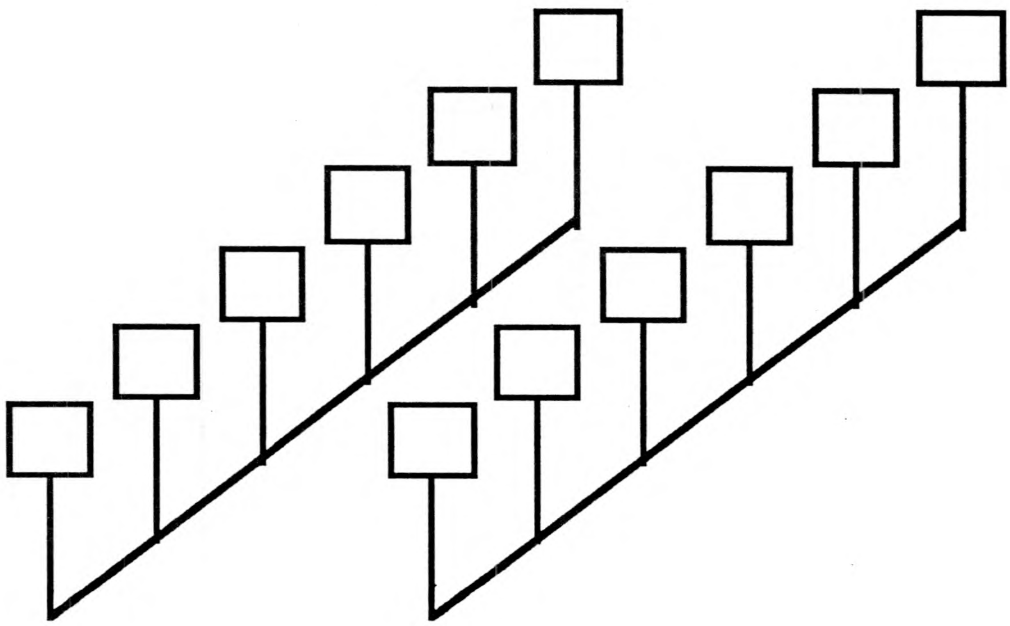
FRISE DE ROBOTS

La route

Niveau : **CM2**

Objectifs : - Décomposition d'un problème.
- Orientation - direction de droites.

Nous avons proposé aux enfants le dessin suivant :



Comment faire exécuter ce dessin par la tortue ?

Si on ne commence pas par réfléchir au problème avec toute la classe, les enfants s'empressent sur leur clavier pour pianoter une succession désordonnée de LC AV TD TG .... en tentant de réaliser le dessin en pas à pas, comme s'ils dessinaient avec un crayon.

Il est préférable de ne pas les laisser perdre trop de temps dans cette voie qui est sans grand intérêt.

On leur demande, avant tout , de **dire ce que l'on voit** sur ce dessin.

Selon les sensibilités, on peut y voir une route bordée d'arbres ou de lampadaires ! Les enfants ont opté pour des arbres.

Il faudra donc apprendre à la tortue à dessiner un arbre. Pour cela, il sera nécessaire d'écrire une procédure ARBRE.

Cela suffira-t-il pour composer cette route ?

On pourrait en effet s'en contenter, mais on peut aussi décomposer la route en deux bords, en écrivant aussi une procédure BORD.

Dans un deuxième temps, après cette mise au point préliminaire, les enfants ont travaillé pour écrire chacune de ces procédures.

La procédure ARBRE n'a posé aucun problème. Ils ont ensuite cherché à dessiner chaque bord de la route indépendamment de sa position relative sur l'écran ; il a suffi de passer correctement d'un arbre à l'autre.

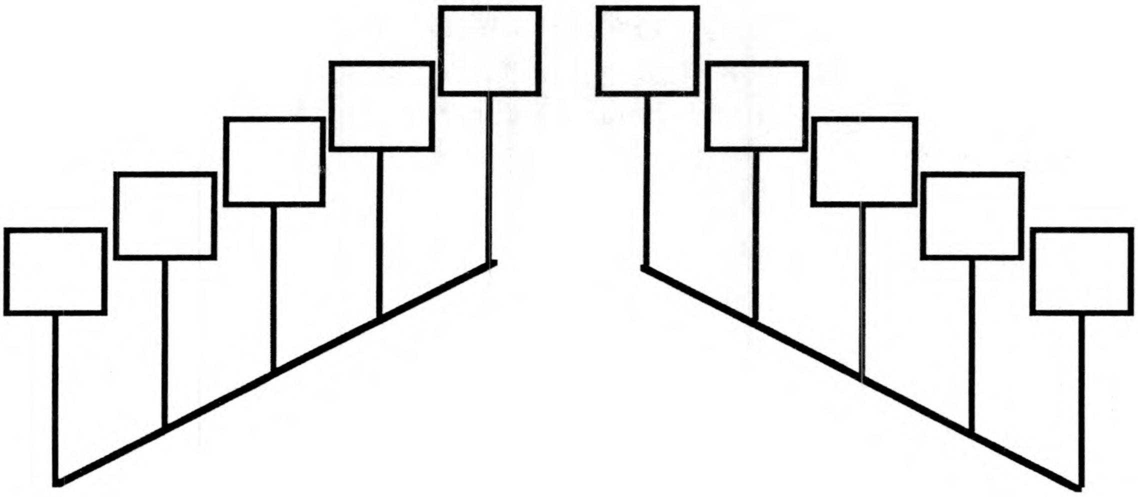
Dans un troisième temps, il s'est agit de coordonner ces procédures entre elles pour réaliser la route ; ceci a demandé beaucoup de soin pour ramener et positionner correctement la tortue après le tracé du premier bord.

Grâce à la discussion initiale, on a abouti à des programmes bien structurés du type

```
POUR ROUTE
DEBUT BORD
DEPLACEMENT BORD
FIN
```

Une autre façon de voir la route

Pour les enfants les plus rapides, nous avons proposé une variante du dessin précédent, la même route vue sous un autre angle.



Après les essais infructueux de certains enfants qui croyaient réussir, tout le monde constate que le bord droit de la route ne peut pas être tracé avec la procédure BORD. Il faut donc écrire une nouvelle procédure BORD2 ; ici le calcul de l'angle du bord droit donne lieu à une recherche très intéressante.

Enfin là encore, le travail final de coordination a été difficile, mais très fructueux sur le plan du raisonnement.

Remarque

Le lecteur désireux de dessiner une route où la perspective soit respectée, devra utiliser des procédures paramétrées (ce que nous n'avons pas fait avec les enfants) et, pourquoi pas, la récursivité. (cf. p.130).

Décomposition de figures

Niveau: **CE - CM**

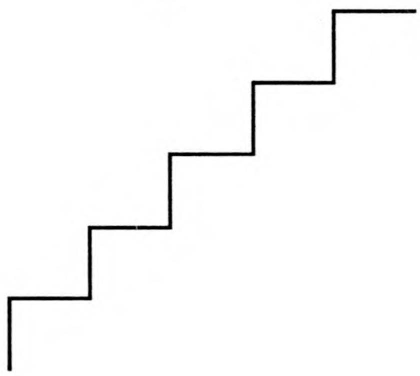
Objectifs : - analyse d'une figure géométrique
- reproduction d'un motif par un déplacement (translation - rotation)
- reconnaissance de formes

Les figures qui suivent ont été construites avec une particularité : dans chacune d'elles on peut repérer un motif qui se reproduit par rotation ou par translation.

Pour faire exécuter ces figures par la tortue, les élèves peuvent utiliser différentes stratégies, l'une d'elles étant de tracer pas à pas chaque trait , sans aucune analyse.

Ce procédé étant coûteux en temps et souvent peu efficace, il nous a été facile de montrer aux enfants l'intérêt d'une analyse préalable, pour aboutir à une procédure plus facile à écrire. Ensuite "c'est la tortue qui fait tout le travail !"

Ces figures vous sont présentées dans un ordre quelconque, elles ne présentent pas de difficulté croissante. Pour chacune d'elles, plusieurs analyses sont possibles.



Ici, on repère assez facilement un motif qui se reproduit par translation. Il suffit d'écrire :

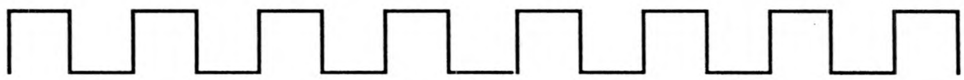
POUR ESCALIER
REPETE 5 [AV 10 TD 90 AV 10 TG 90]
FIN

Ou encore, d'une façon plus élaborée :

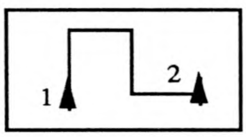
POUR MARCHE
AV 10 TD 90 AV 10 TG 90
FIN

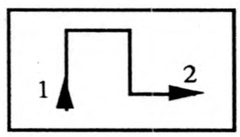
POUR ESCALIER
REPETE 5 [MARCHE]
FIN

Comme nous l'avons relaté dans notre introduction, une erreur fréquente consiste à négliger la commande TG 90. Une fois cette erreur corrigée, remarquons que la procédure MARCHE produit alors à la fois le motif **et** la translation de la tortue qui se trouve ainsi en "bonne position" pour démarrer le tracé de la marche suivante.

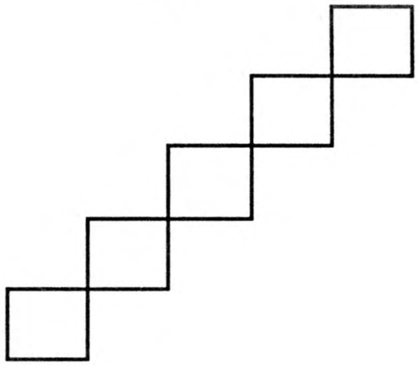


Sur cette figure, il s'agit encore d'une translation, mais le choix du module de base n'est pas aussi simple que dans l'exemple précédent. On peut prendre, par exemple, l'un des deux modules suivants :

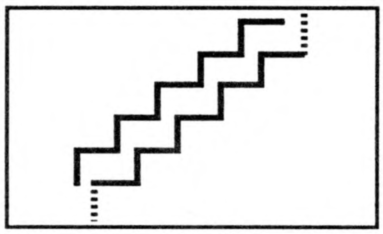




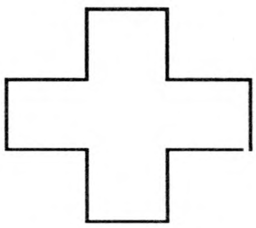
La difficulté pour les enfants a été de faire clairement ce choix pour pouvoir ensuite coordonner correctement la position de la tortue. Il faut les inciter à représenter une suite d'actions par un motif, c'est-à-dire à représenter les positions 1 et 2 de la tortue.



Dans cette figure, selon les modules construits précédemment, on peut voir, par exemple, soit une succession de carrés, soit le motif ESCALIER reproduit deux fois, à une petite modification près, comme indiqué ci-dessous.



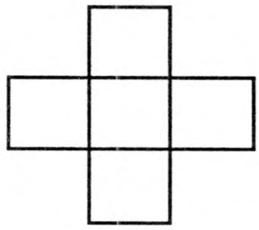
La difficulté, une fois le motif reconnu, sera encore de positionner la tortue entre les deux réalisations du motif ESCALIER.



Sur cette figure, il est possible de repérer un motif qui se reproduit quatre fois par rotation. Le choix du motif n'est pas unique, la difficulté sera encore d'orienter correctement la tortue à chaque étape.

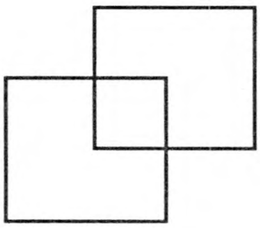
Il en sera de même pour la figure suivante, qui peut être vue comme formée de deux rectangles, mais aussi comme formée par quatre carrés, reproduits par rotation. Cette dernière analyse nous parait refléter le mieux la structure de la figure. Elle a été trouvée sans notre aide, par quelques enfants.

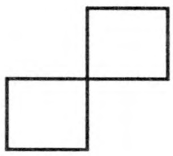
```
POUR CARRE
REPETE 4 [AV 20 TD 90 ]
FIN
```

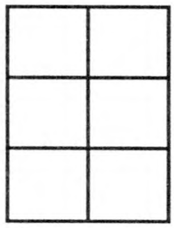


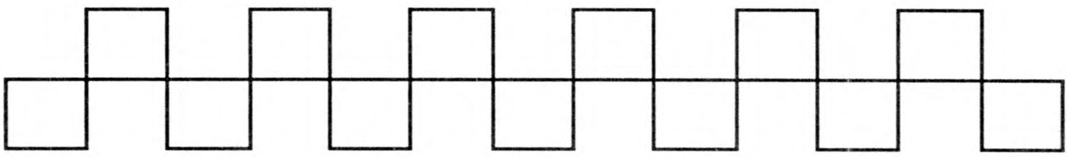
```
POUR FIGURE
REPETE 4 [ CARRE TD 90 AV 20 ]
FIN
```

En utilisant encore des carrés, on peut réaliser les dessins suivants :

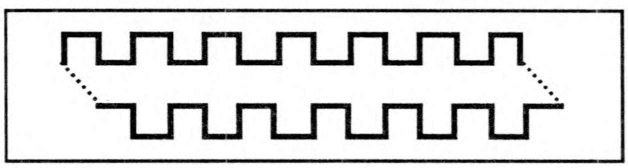




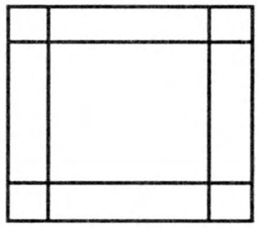




Bien entendu, il est possible d'utiliser d'autres décompositions. Ce dernier dessin, par exemple peut être décomposé à l'aide du motif CRENEAU élaboré précédemment, avec une petite modification, comme on peut le voir ci-dessous.



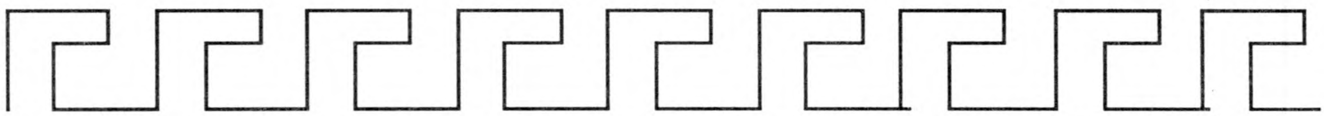
Dans cet exemple, on peut décomposer de multiples façons la figure en différents carrés et rectangles. Il existe cependant une analyse très efficace, trouvée par quelques enfants, qui utilisent un seul module RECTANGLE reproduit par quatre rotations:



```
POUR RECTANGLE
REPETE 2 [ AV 50 TD 90 AV 10 td 90 ]
FIN
```

```
POUR MOUCHOIR
REPETE 4 [ RECTANGLE AV 50 TD 90 ]
FIN
```

Et pourquoi ne pas terminer par une de ces jolies frises qui ornent si bien le bas des cahiers !

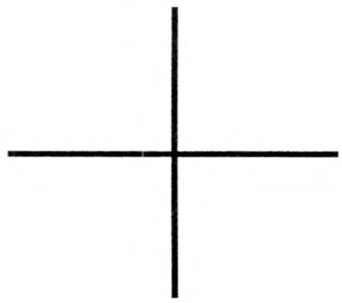


Des étoiles

Niveau : **CM1-CM2.**

Objectifs :
- Analyse d'une figure.
- Mesure d'angles.
- Reconnaître une situation de division.

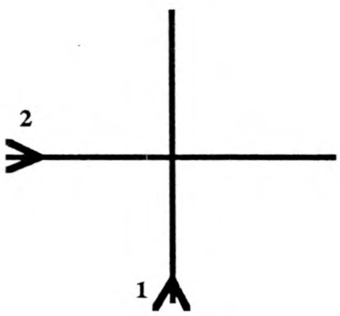
Dans un premier temps, nous avons proposé ce dessin aux élèves.



Comme la tortue n'est pas représentée sur notre modèle, le choix de la position initiale de la tortue est fait par les enfants. Cela aboutit à des procédures variées, dont voici des exemples.

1er exemple

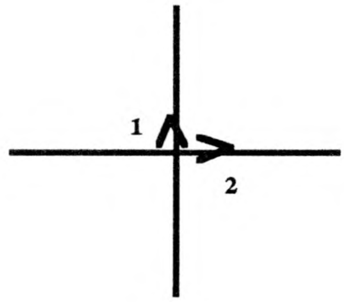
```
POUR ETOILE
AV 80 RE 40
TD 90 AV 40 RE 80
FIN
```



La tortue parcourt l'étoile en pas à pas ; la figure n'est pas structurée.

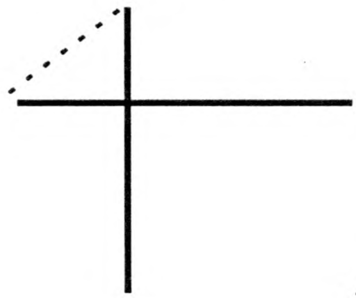
2ème exemple

REPETE 2 [AV 40 RE 80 AV 40 TD 90]



Ici la figure est analysée comme faite de deux segments perpendiculaires: l'élève a même découvert un centre de rotation qu'il a su utiliser.

Par contre, à partir de la même analyse, on a aussi des tentatives infructueuses :



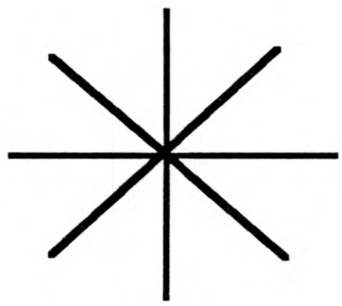
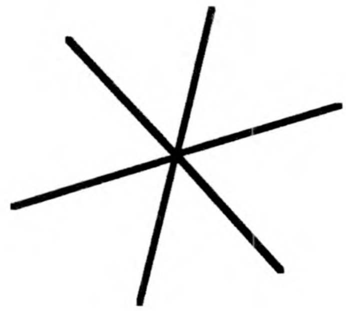
AV 80 LC ....

On voit ici que l'enfant essaie de tracer deux segments perpendiculaires, comme on le ferait «à main levée», en tentant de «viser» de façon approximative,vers l'extrémité du second segment.

Après ces premiers essais, nous avons comparé les procédures produites par les différents groupes et nous avons choisi la suivante en précisant que, pour la suite ce dessin serait une étoile à 4 branches :

```
POUR ETOILE4
REPETE  4  [ AV 40 RE 40 TD 90]
FIN
```

Dans un deuxième temps, nous avons demandé aux élèves de réaliser les étoiles suivantes :

- Il y a peu de difficultés pour ETOILE 8, l'angle de 45° est très vite proposé par les enfants, permettant de tracer «le milieu» de l'angle droit (on en profite pour glisser le mot de bissectrice, plus difficile à dire, mais tellement plus mathématique !)

Par ailleurs, une analyse très intéressante, que nous n'attendions pas, apparaît :

```
POUR ETOILE 8
ETOILE 4
TD 45 ETOILE 4
FIN
```

- L'étoile à 6 branches va provoquer beaucoup plus d'hésitations. On revient à l'étoile à 4 branches pour aider les enfants.

«La tortue a tourné 4 fois de 90. Elle a donc tourné de combien, à la fin ?»
«Elle a tourné de 360»
«Ici, elle tourne 6 fois. Mais il faut encore que cela fasse 360, comment faire ?»

Il s'ensuit une recherche sur les multiples de 6, pour trouver celui qui sera égal à 360. Le nombre 60 est assez vite trouvé puisque $6 \times 60 = 360$.

Par la suite, nous demandons aux enfants de faire des étoiles à 10 branches, 12 branches, 5 branches, 7 branches. Cette fois-ci, les enfants n'ont pas de modèle sous les yeux. Ils doivent partir d'une description verbale et faire référence à une image de l'étoile faite à partir des exemples précédents.

L'angle de la rotation est assez facilement trouvé pour l'étoile de 12 branches.

```
REPETE 12 [ AV 40 RE 40 TD 30 ]
```

Par contre les choses se compliquent de nouveau pour l'étoile à 5 branches

$5 \times \ldots = 360$

Le tâtonnement est plus laborieux et beaucoup d'essais sont nécessaires.

Nous avons pu constater, et ce, dans plusieurs classes de CM2, que très peu d'enfants utilisent spontanément la division pour répondre à cette question. Ce fait est assez remarquable et montre combien il est utile de placer les enfants dans des situations nouvelles pour renforcer les acquisitions arithmétiques

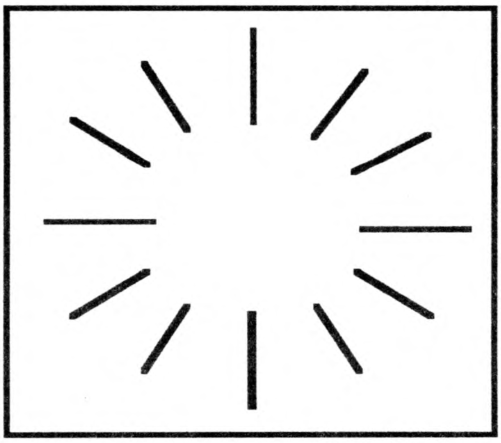
PROLONGEMENT : UNE HORLOGE.

Objectif : - Décomposition d'une figure.
- Utilisation des procédures ÉTOILE.

Nous avons prolongé ce travail, en utilisant les étoiles nouvellement trouvées, pour tracer les graduations de notre horloge.

Ce travail a donné lieu à des analyses très intéressantes pour tracer les douze graduations qui ne sont pas toutes de même longueur. On peut :

- les décomposer en quatre quadrants identiques
- tracer les quatre grandes graduations puis insérer les huit petites, ce qui demande des calculs d'angles délicats
- ou encore tracer douze petites graduations régulièrement, puis les quatre grandes par-dessus



Des ailes et des moulins

Niveau : CM1 - CM2

Objectifs : - Familiariser les enfants avec la notion d'angle, à partir des rotations.
- Evaluation de la mesure d'un angle.
- Décomposition d'un problème.

Jusqu'ici les enfants ont travaillé sur quadrillage, en n'utilisant que les commandes TD 90 et TG 90 (production d'angles droits). Nous voulons maintenant leur faire découvrir et utiliser d'autres angles.

Le point de départ (sans ordinateur).

Voici ce que Gino, un enfant de la classe, a écrit pour déplacer la tortue dans un programme :

LC AV 10 TD 90 TG 90 AV 5 TG 90 TG 90 BC.

«Peut-on simplifier cette écriture ?»

Les enfants remarquent que TD 90 TG 90 ne sert à rien, on peut donc le supprimer
Mais alors AV 10 AV 5 peut être remplacé par AV 15.
Par quoi a-t-on envie de remplacer TG 90 TG 90 ? La réponse est unanime :
TG 180

On convient donc que :

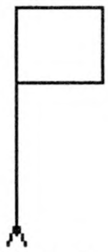
Le nombre qui suit TD ou TG indique "de combien " tourne la tortue.

Dans les activités qui vont suivre, nous allons mettre en oeuvre cette découverte.

Le moulin à quatre ailes.

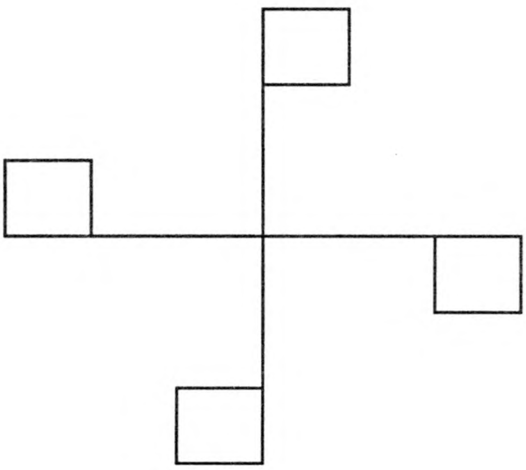
On demande aux enfants d'écrire la procédure AILE associée au module suivant :

```
POUR AILE
AV 30 TD 90 AV 10 TD 90
AV 10 TD 90 AV 10 TD 90
RE 20
FIN
```



$\overrightarrow{F_{12}}$

Puis de réaliser un moulin à 4 ailes, c'est-à-dire d'obtenir sur l'écran :



Remarque.

On a suggéré d'emblée le module AILE avec le «bon départ» et la «bonne arrivée» pour la tortue, ce qui permet la réalisation de « MOULIN4» sans trop de difficultés pour tous. On obtient pour une majorité d'enfants:

```
POUR MOULIN 4
REPETE 4 [AILE TD 90]
FIN
```

Le moulin avec beaucoup d'ailes.

Il semble difficile a priori de faire faire à la tortue quelque chose qu'on n'a pas encore dessiné soi-même... c'est pourtant ce que l'on va demander aux enfants maintenant, puisqu'on leur demande d'obtenir sur l'écran un moulin à 6 ailes, 8 ailes, 10 ailes, etc.

Cette fois-ci l'enfant n'a pas la figure sous les yeux, il doit imaginer l'objet à partir du moulin précédent. (Il est implicitement admis que ces nouveaux moulins ont la même structure que MOULIN4, c'est-à-dire que l'angle entre deux ailes est constant).

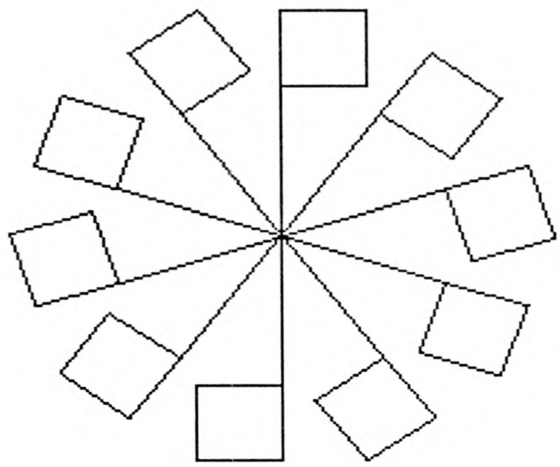
Les enfants procèdent par tâtonnement. La plupart écrivent tout d'abord :

REPETE 6 [AILE TD 90].

Ensuite chaque enfant imagine une stratégie pour corriger l'erreur qu'il vient de constater : il remplacera par exemple 90 par 70, 75 etc. et à chaque étape il sera obligé de modifier ses instructions en fonction de ce qu'il voit sur l'écran.

L'erreur devient donc productive puisque chaque nouvel essai est fonction de l'essai précédent.

Les enfants sont très actifs, ils ont envie de réussir, ils sont en position d'acteurs et l'ordinateur leur permet de découvrir, sans aide extérieure, la solution au problème posé.



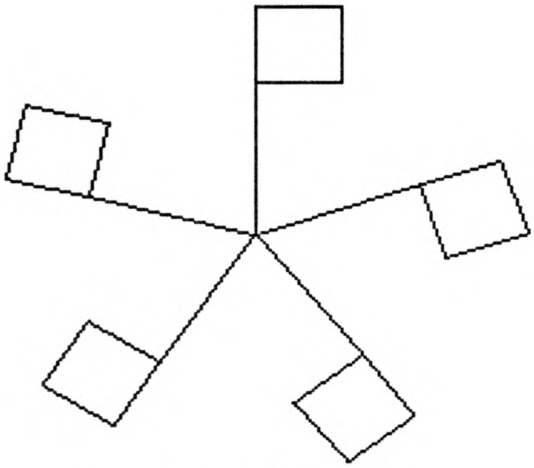
Validation du résultat

Il faut toutefois remarquer que l'image que renvoit l'ordinateur est appréciée «à l'œil» et jugée comme bonne ou mauvaise par l'enfant lui-même; ce n'est donc pas, à proprement parler, une validation au sens de preuve mathématique mais plutôt un support à la recherche empirique du résultat. Par ailleurs, l'imprécision due à la qualité de l'écran ne facilite pas la tâche.

Il a donc fallu quelquefois inciter les élèves à une vérification supplémentaire.

Exemple :

Un enfant était satisfait de REPETE 5 [AILE TD 71] comme instruction réalisant MOULIN5.

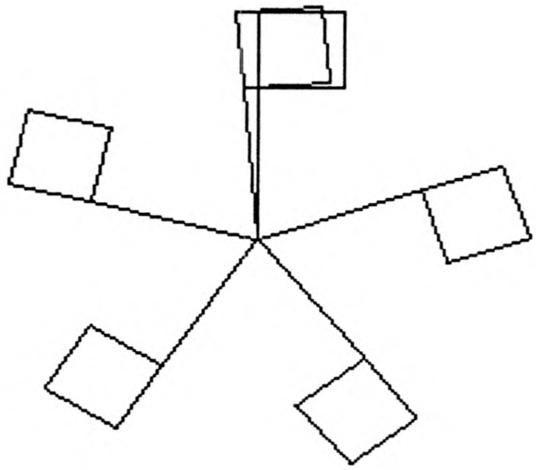


REPETE 5 [AILE TD 71]

Il est en effet parfaitement impossible de déceler à "l'oeil nu" une quelconque différence entre chaque angle.

Nous lui avons alors demandé :
«Comment peux-tu vérifier que ta solution est bonne ?».

L'enfant a alors fait dessiner une aile de plus : celle-ci étant décalée par rapport à la première aile, il en a conclu à l'inexactitude de son premier dessin.



REPETE 5 [AILE TD 71] AILE

Conclusion

Après cette séance, la mesure d'angle devient une chose que l'on a expérimentée et manipulée.

La multiplicité des figures réalisées fait qu'à l'issue de cette activité l'enfant a une représentation visuelle d'un angle de 30°, de 45°, etc. La «tortue» exécute les ordres rapidement en respectant les données et la précision des figures obtenues, sans être parfaite, est bien meilleure que ce que l'on pourrait dessiner à la main même avec un rapporteur !

Remarque.

Il faut constater que les enfants procèdent toujours par «essai-erreur» et qu'ils ne pensent à aucun moment à diviser 360 par le nombre d'ailes !

Certains remarquent :

«si on double le nombre d'ailes, l'angle de rotation doit être divisé par deux».

D'autres :

«si on multiplie la valeur de l'angle de rotation par le nombre d'ailes on trouve 360» c'est-à-dire que le nombre que l'on cherche vérifie la multiplication à trou :

nombre d'ailes x ... = 360.

On remarque que même en CM2 la situation de division n'est pas encore bien acquise et que les enfants ne réinvestissent pas le mécanisme opératoire.

Il faut donc laisser les enfants maîtres de leur expérience et ne pas «divulguer» trop tôt la relation qui existe entre le nombre d'ailes et la valeur de l'angle de rotation. Il sera nécessaire après plusieurs séances utilisant l'ordinateur, de faire le point «en classe».

En partant des expériences faites avec la tortue, on essayera d'aboutir à la modélisation, en établissant le théorème :

"Pour un moulin à n ailes, la valeur de l'angle de rotation est égale au quotient de 360 par n"

Les polygones réguliers

Niveau : **CM2**

Objectifs : - découverte des propriétés géométriques des polygones réguliers
- mesure d'angles
- situation de proportionnalité

Les enfants savent tous faire dessiner un carré à la tortue.

A partir de cet exemple, nous expliquons aux enfants que le carré est un polygone régulier à quatre côtés.

Régulier signifie : tous les côtés égaux,
tous les angles égaux.

On leur demande alors de faire tracer par la tortue, un polygone régulier à 8 côtés.

Le problème est difficile ; ici les enfants n'ont pas sous les yeux un modèle, il doivent faire référence à une description verbale de l'objet à dessiner.

Par analogie avec le carré, les élèves proposent :

REPETE 8 [AV 10 TD ...]

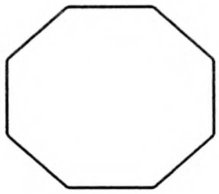
Un premier groupe d'élèves profitant des possibilités de l'ordinateur recherchent par essais successifs plusieurs mesures d'angles jusqu'à l'obtention d'une procédure qui leur parait exacte. Cette démarche toute expérimentale est intéressante car si au début les enfants essaient n'importe quoi, petit à petit ils se forgent une conjecture sur l'encadrement de l'angle comme :

$$30 < n < 90$$

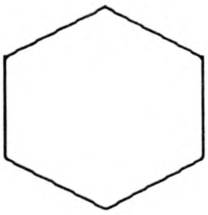
Ici l'ordinateur joue son rôle d'outil donnant à la géométrie un éclairage expérimental inconnu à ce jour.

Un deuxième groupe d'élèves, peu soucieux de précision, pense que l'angle de rotation étant plus petit que 90 degrés, doit être "la moitié"! Ils font mouche du premier coup !

Enfin un dernier groupe d'élèves faisant le rapprochement avec le carré, induisent que si à 4 côtés sont associés des angles de 90°, à 8 côtés doivent être associés des angles de 45°.



Nous avons repris la même recherche pour un polygone régulier à six côtés.



Nous avons ensuite essayé de faire une synthèse afin de découvrir une règle générale

Nous avons proposé le tableau suivant qui était à compléter par les élèves :

nombre de côtés (n)	4	6	8	9	12	20
angle (a)	90	60	45			
n * a	360	360	360			

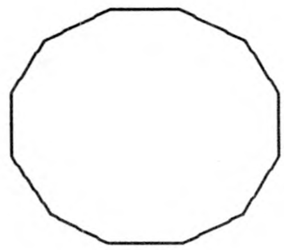
Il s'agit là d'observer dans un premier temps, les résultats obtenus empiriquement sur les écrans.

Il n'est pas possible avec les enfants de faire une démonstration déductive pour trouver la relation entre le nombre de cotés et la valeur de l'angle. Par contre, il nous a paru possible et souhaitable de développer chez eux une démarche inductive, c'est-à-dire de leur demander de découvrir cette relation à partir de ce qu'ils peuvent observer dans ce tableau.

Ils utilisent ensuite le fait observé (le produit n * a est égal à 360), pour compléter le tableau.

Nous avons ensuite repris le travail avec la tortue-logo. Ainsi lorsque nous avons demandé la procédure du polygone à douze cotés (dodécagone), la plupart des élèves n'ont pas hésité à écrire :

```
POUR POLY 12
REPETE 12 [AV 30 TD 360 / 12]
FIN
```



Ce travail a été repris et prolongé par la suite au cours des activités sur le cercle.

Le cercle

Niveau : **CM2**

Objectif : - Découvrir une nouvelle définition du cercle
- Utiliser les propriétés du cercle.

Très vite, dans toutes les classes, les enfants ont envie de savoir faire un cercle.

Après avoir mené des activités sur les angles et les polygones réguliers, nous nous sommes posé la question :

«Comment tracer un cercle ?».

Nous avons abordé ce problème de deux façons différentes :

- Définition classique du cercle : la tortue dessine chaque point du cercle à distance égale du centre, mais elle doit pour cela faire des aller-retour.

ou bien

- La tortue parcourt "la circonférence du cercle". Cette approche est beaucoup plus nouvelle, le centre et le rayon ne sont pas matérialisés.

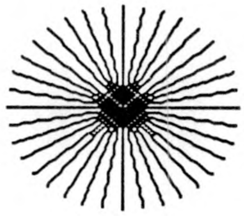
Il n'est pas indispensable de présenter ces deux méthodes aux enfants : ce sont deux analyses du cercle différentes, elles sont complémentaires mais indépendantes.

la première décrit le cercle comme ensemble des points équidistants d'un point fixe ;
la seconde décrit le cercle comme une courbe de courbure constante.

1ère analyse.

On prend tout d'abord l'image d'une roue de bicyclette ; les enfants ont l'idée de ne tracer que les rayons de cette roue. Pour cela, on peut écrire par exemple :

```
POUR RAYONS
REPETE 36 [AV 20 RE 20 TD 10]
FIN
```



Pour tracer une roue pleine, il faudra «resserrer les rayons pour qu'ils se touchent».

```
POUR DISQUE
REPETE 360 [AV 20 RE 20 TD 1]
FIN
```

Mais comment ne tracer que le pneu ? (C'est-à-dire le cercle).

Les enfants ont l'idée d'aller jusqu'au pneu en "lève crayon", puis de dessiner quelque chose... on suggère de faire une petite croix pour réaliser un pneu clouté.

On écrit alors

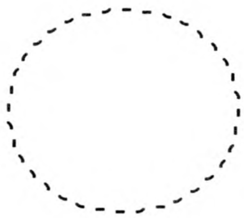
```
POUR CLOU
REPETE 4 [AV 1 RE 1 TD 90]
FIN
```

```
POUR PNEUCLOU
LC
REPETE 36 [AV 20 BC CLOU LC RE 20 TD 10]
BC
FIN
```

Après avoir réalisé ces différents dessins sur l'écran, un enfant dit : «Si je veux faire un pneu sans clou je vais supprimer la barre verticale de la croix». Le «clou» devient alors «trait». On écrit :

```
POUR TRAIT
TD 90
AV 1 RE 2 AV 1
TG 90
FIN
```

```
POUR PNEU
LC
REPETE 36 [AV 20 BC TRAIT LC RE 20 TD 10]
BC
FIN
```



Jusque là tous les tracés se font avec la position de départ de la tortue au centre de la roue.

Lorsqu'on réalise PNEU on obtient un pneu en pointillé ou bien un pneu où les différents traits se chevauchent (selon la dimension de «trait» et du pneu). Cela ne satisfait pas pleinement les enfants.

2ème analyse.

N'est-il pas possible de tracer seulement le bord de la roue, sans faire faire tous ces aller-retour à la tortue ?

On imagine alors que la tortue «marche sur le pneu de la bicyclette».
Comment fait-elle ?
«Elle avance en tournant».

Des enfants proposent alors de faire faire des pas très petits à la tortue. On écrit, par exemple :

```
POUR PNEU5
REPETE 200 [AV 1 TD 1 AV 1]*
FIN
```

Ceci ne permet pas d'obtenir un cercle complet, on essaye donc avec un nombre plus grand que 200 ; (Il est difficile de justifier mathématiquement pourquoi la rotation totale de la tortue doit être 360).

Par analogie avec le travail fait sur les polygones, on finit par s'entendre sur la procédure :

* Nous avons préféré cette description (AV 1 TD 1 AV 1) à l'habituelle (AV 2 TD 1). Ce choix est longuement justifié dans le chapitre "Cercles et arcs de cercles" (p. 91)

```
POUR CERCLE
REPETE 36 [AV 1 TD 10 AV 1]
FIN
```

Par la suite, nous avons posé une question qui provoque une recherche intéressante. On a demandé aux enfants de réaliser avec cette méthode des cercles de différentes grandeurs.

Un enfant tape :

```
REPETE 20 [AV 4 TD 18 AV 4]
```

puis

```
REPETE 40 [AV 2 TD 9 AV 2]
```

et il est surpris de dessiner deux cercles «identiques» à l'œil.

On explique alors le phénomène après avoir fait faire ces deux tracés par tous les groupes.

On demande : «combien de pas la tortue a-t-elle fait de pas dans le 1er cas ? Dans le 2ème cas ?».

Quel est le périmètre du 1er «cercle» ?

Quel est le périmètre du 2ème «cercle» ?

Ceci nous permet d'avoir une idée approximative de la taille du cercle.

Nous n'entrerons pas, avec les enfants, dans le problème du calcul approximatif du rayon. Cette question délicate est abordée, pour les grands, dans le chapitre "Cercles et arcs de cercles " (p.91).

Le parallélogramme

Niveau : **CM1-CM2**
Objectif : - Propriétés des parallélogrammes.
- Propriétés des droites parallèles.

Avant d'aborder ce travail avec l'outil informatique, il est nécessaire que le mot parallélogramme évoque un objet connu des enfants.

Notre objectif n'est pas de le définir ici, mais plutôt,à partir de la nécessité de le faire tracer par la tortue, de mettre en œuvre ses propriétés et de les expliciter.

Les enfants savent déjà que le parallélogramme a ses côtés opposés parallèles et de même longueur.

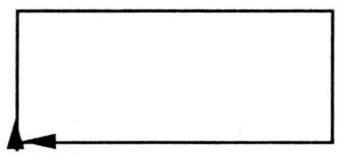
Nous avons commencé par poser le problème suivant :

«faire tracer à la tortue un parallélogramme dont les côtés mesurent 30 et 50».
(Unité : pas de tortue).

Première recherche.

Ce problème, volontairement imprécis, laisse les enfants perplexes. Un groupe propose enfin la solution suivante :

AV 30 TD 90 AV 50 TD 90
AV 30 TD 90 AV 50



Nous en profitons pour faire évoluer la recherche de tous.

«Que dessine cette procédure ?».

Avec ou sans l'aide de l'écran, tout le monde reconnaît un rectangle.

«Est-ce que cela répond au problème posé, c'est-à-dire, est-ce un parallélogramme ?»

Aussitôt s'ouvre un débat très révélateur : peu d'enfants admettent qu'il s'agit aussi d'un parallélogramme. Cela nous donne l'occasion d'insister sur ce point et de confirmer que le rectangle **est** un parallélogramme particulier.

«Qu'est-ce qui fait que ce parallélogramme est un rectangle ?».

«L'angle de 90°».

A partir de ce constat, on décide de tracer un parallélogramme qui ne comporte pas d'angle droit.

Deuxième recherche.

Pour aider les enfants, on propose de commencer la procédure par :

AV 30 TD 60 AV 50

Les difficultés se retrouvent au moment de tracer le second côté de longueur 30. Comment orienter la tortue ? Certains enfants tournent la difficulté en écrivant

AV 10 TD 60 AV 50
TG 60 RE 30 **TD 60** RE 50

Après quelques essais, les enfants trouvent qu'il faut la faire tourner de 120° vers la droite. En effet, elle aura alors fait un demi-tour par rapport à sa direction initiale.

Le même travail se renouvelle pour tracer le deuxième côté. Nous devons signaler que cette démarche a été difficile et que peu d'enfants en sont venu à bout sans notre aide.

On propose à tous la solution suivante :

```
POUR PARALLELOGRAMME
REPETE 2 [AV 30 TD 60 AV 50 TD 120]
FIN
```

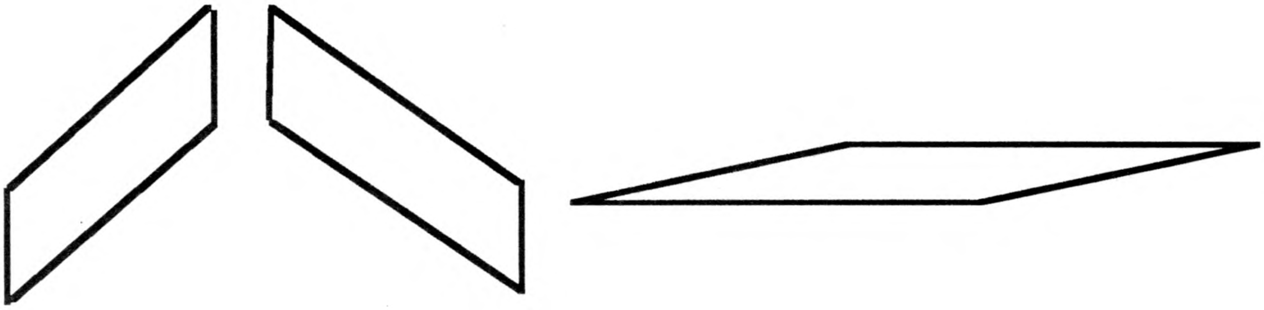
On analyse cette procédure en observant que les 2 rotations de la tortue sont telles que leur somme vaut 180° (Nous ne sommes pas loin des angles alternes-internes).

Troisième recherche.

Nous avons voulu montrer aux enfants que la seule donnée des longueurs des côtés ne définit pas entièrement un parallélogramme.

Nous voulions également leur faire réutiliser la procédure précédente, qui n'avait pas été facile à trouver.

On leur a demandé de faire tracer par la tortue plusieurs parallélogramme différents avec les mêmes longueurs de côtés. On obtient par exemple :



Ce travail s'est fait assez facilement.

Pour renforcer les différentes notions rencontrées pendant ces recherches, nous avons fait tracer en classe des parallélogrammes avec une régle et un compas. Les enfants ont ainsi pu vérifier que selon l'outil utilisé il fallait prendre en compte des propriétés différentes.

Papier à réseau triangulaire

Niveau : **CM_1 et CM_2.**

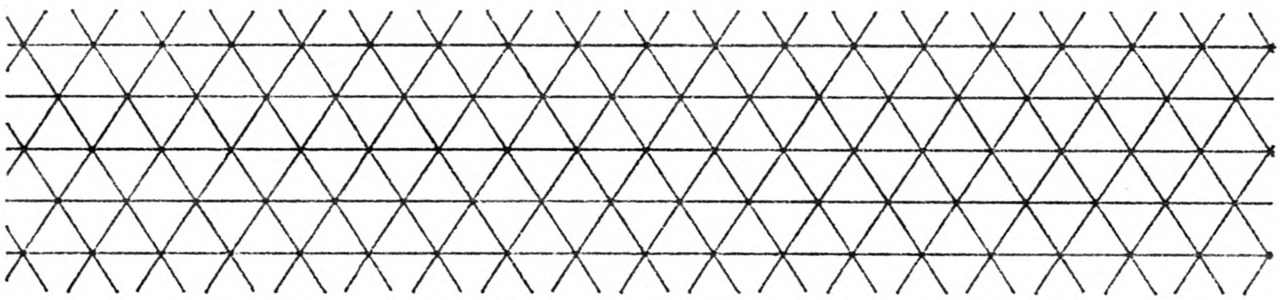
Objectifs : Mesure d'angles
Homothétie
Symétrie

Comme tout un chacun peut le constater, les figures dessinées par la tortue logo, qu'elles soient inventées par les enfants ou proposées par l'enseignant, abondent en angles rigoureusement droits.

Il y a à cela plus d'une raison, l'une d'entre elles étant qu'il est délicat de faire reproduire par les élèves des figures contenant des angles quelconques, puisque qu'ils ne disposent pas d'un instrument de mesure adéquat.

(le rapporteur n'étant pas d'un usage très judicieux à l'école élémentaire compte tenu des acquis peu sûrs à propos de la notion d'angle).

Nous avons donc essayé de varier quelque peu les figures proposées en les présentant dessinées sur un papier à réseau dont la maille est un triangle équilatéral (vous en trouverez en annexe divers modèles pour reproduction); ce support a l'avantage de fournir au élèves un instrument de mesure aisé à la fois pour les longueurs et les angles.



Dans un premier temps, l'observation de ce réseau avec les élèves nous a permis d'évaluer la mesure de chaque angle y figurant. Après discussion et vérification on aboutit aux conclusions suivantes :

- on y retrouve l'angle plat, déjà connu des enfants comme le demi-tour de la tortue (TD 180 ou TG 180) ;

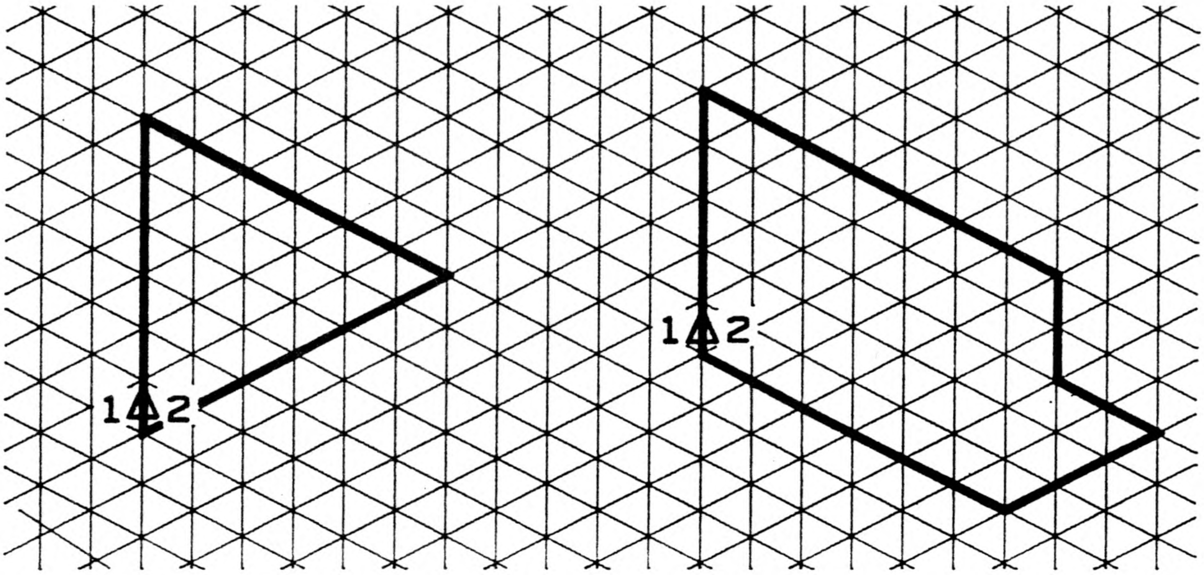
- on peut observer trois angles adjacents égaux, dont la somme vaut 180° ; les élèves trouvent alors une mesure de 60° pour chacun de ces angles.

- on en déduit facilement la mesure de tous les angles qui figurent sur ce réseau
- on découvre ou bien on redécouvre que les angles intérieurs d'un triangle équilatéral mesurent 60°.

Dans les exemples qui suivent l'objectif est tout d'abord de faire utiliser aux enfants des angles autres que l'angle droit. Mais nous avons aussi travaillé sur d'autres notions géométriques tout en utilisant le même support papier.

Titi et Toto

Objectifs: agrandissement, réduction, homothétie.



Les enfants n'ont pas eu de difficulté pour écrire une procédure TITI. Par contre, les choses se compliquent nettement avec le dessin TOTO, comme en témoigne la procédure suivante d'un élève :

```
POUR TOTO
AV 50 TD 120 AV 70 TD 60 AV 20 TG 120 AV 20
TD 80 AV 30 TD 120 AV 60
FIN
```

Nous pouvons alors constater que des enfants qui avaient mal maîtrisé précédemment les rotations à angle droit de la tortue se trouvent ici devant un problème plus complexe. Nous avons dû faire d'autres dessins du même type avant d'aller plus loin.

Dans un deuxième temps, nous avons demandé aux élèves d'écrire **deux** procédures pour avoir la figure TOTO dans deux «tailles» différentes (la consigne restant volontairement vague).

Voici deux résultats caractéristiques parmi les productions des élèves :

POUR PERSIL	POUR SKIP
AV 40 TD 120 AV 60	AV 50 TD 120 AV 70
TD 60 AV 10	TD 60 AV 20
TG 60 AV 10	TG 60 AV 20
TD 120 AV 20	TD 120 AV 30
TD 60 AV 50	TD 60 AV 60
FIN	FIN

POUR Z	POUR X
AV 25 TD 120	AV 50 TD 120
AV 35 TD 60	AV 70 TD 60
AV 10 TG 60	AV 20 TG 60
AV 10 TD 120	AV 20 TG 120
AV 15 TD 60	AV 30 TD 60
AV 30	AV 60
FIN	FIN

Nous avons utilisé ces deux exemples en classe pour découvrir avec les enfants une règle permettant de réduire ou d'agrandir une figure en conservant sa forme.

(Pourquoi ne pas les appeler figures semblables...).

- Est-ce que SKIP et PERSIL donnent deux figures semblables ? Et les procédures Z et X ?
- Comparons les écritures de SKIP et de PERSIL : les angles sont identiques, les longueurs sont toutes augmentées de 10.
- Est-ce la bonne méthode pour agrandir une figure ?
- Quelle semble être la bonne méthode ?
- Comparons les écriture de Z et de X : les longueurs ont été «multipliées» par deux.
- Conclusions ?

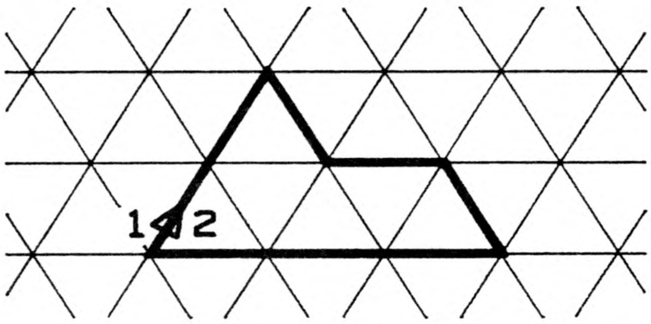
Après ce travail, il ne reste plus qu'à appliquer sur d'autres exemples, avec ou sans tortue logo, la règle ainsi découverte pour vérifier et retenir ces conclusions :

"pour agrandir (ou réduire) une figure sans changer sa forme, il faut multiplier (ou diviser) la longueur des cotés et conserver les angles"

Le sphinx.

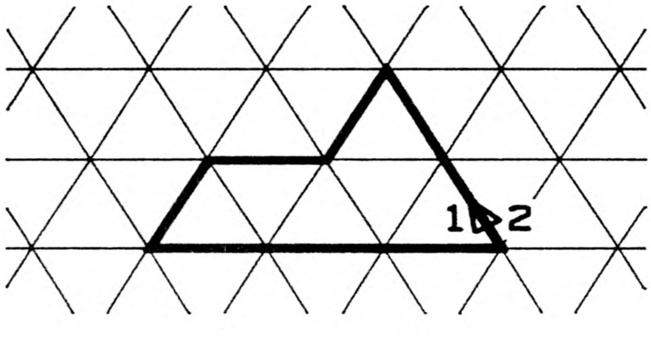
Objectif : - reconnaître des figures symétriques
- propriétés des figures symétriques

Nous avons proposé aux enfants le dessin ci-dessous aussitôt baptisé «sphinx» (à cause de la lecture de Tintin en Egypte, sans doute...).



Après plus ou moins de tâtonnements, tout le monde parvient à une procédure qui produit ce dessin.

Dans un deuxième temps, on fournit aux enfants le dessin suivant.



«Que peux-ton dire de ce nouveau sphinx, par rapport au premier ?».
«Ils sont symétriques».
La chose ne semble faire de doute pour personne.
On pose ensuite la question :
«Comment écrire une procédure pour le sphinx de droite ?».
Avec une quasi unanimité, ils proposent :
«On n'a qu'à changer les TD en TG, et les TG en TD ! ».

Voilà notre découverte des propriétés des figures symétriques qui tourne court !!! Apparemment, dans cette classe du moins, les enfants ont une idée intuitive de la transformation des angles par symétrie ; il reste à le vérifier sur d'autres exemples.

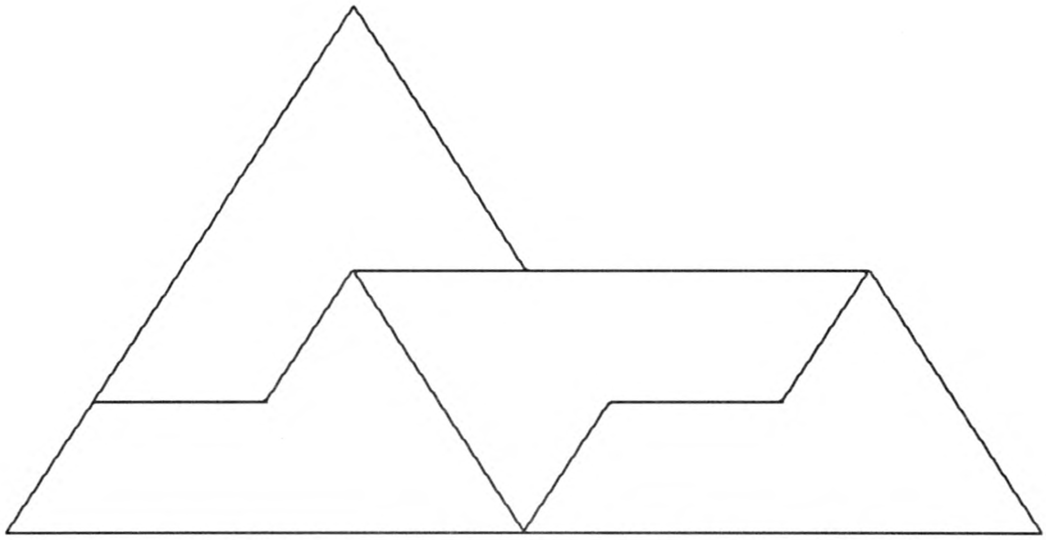
Ici, le fait d'avoir, sur le modèle, un axe de symétrie verticale est sans doute très incitatif. Nous n'avons pas d'autre expérience pour en dire plus.

Ce travail étant fait, les enfants ont chacun à leur disposition deux procédures : SPHINX.DROIT et SPHINX.GAUCHE. En comparant ensuite avec toute la classe l'écriture de ces deux procédures, on retiendra que les longueurs sont égales ainsi que les angles, mais que ceux-ci n'ont pas le même «sens».

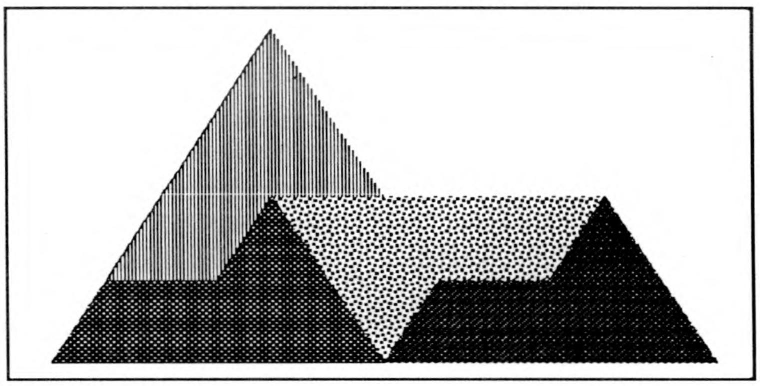
(Il n'est pas question à l'école primaire d'aller plus loin dans le formalisme et notre imprécision choquera peut-être certains ; notre seule ambition est de faire toucher du doigt aux enfants le fait que la symétrie est une isométrie négative afin qu'ils en conservent une première connaissance intuitive).

Des sphinx

Dans une autre séance, nous leur avons proposé le dessin suivant :



Une certaine analyse de cette figure montre qu'elle peut se décomposer en quatre sphinx.



On décide de dire de ce dessin que c'est un «grand sphinx».

Est-ce suffisamment précis comme dénomination ?

"Non, puisque comme les petits, les grands sphinx peuvent être «droit» ou «gauche» ".

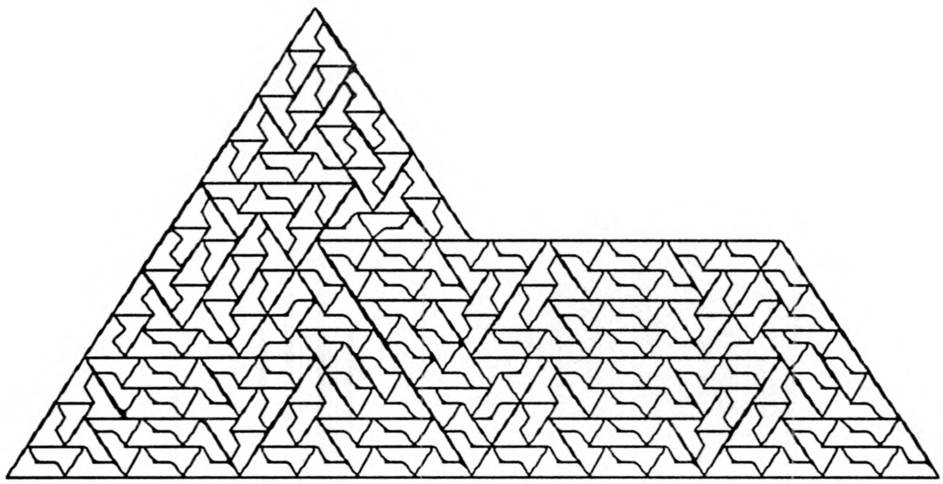
Par comparaison avec les petits sphinx dessinés précédemment, on appellera ce dessin : grand sphinx droit.

Après ces remarques préliminaires, on propose aux enfants le travail suivant :
«Ecrire une procédure G.SPHINX.DROIT en utilisant les procédures SPHINX. DROIT et SPHINX.GAUCHE».

Notre objectif était de leur faire retrouver, parmi les quatre petits sphinx de la figure les SPHINX.DROIT et les SPHINX.GAUCHE. Il nous faut avouer que la réussite à ce travail a été quasi nulle !
Les raisons en sont diverses mais la première est que les enfants trouvaient plus économique d'écrire une procédure pas à pas, sans revenir à des sous-procédures ; on ne peut pas leur donner tort !

Pour aller plus loin

Nous invitons le lecteur courageux à prolonger lui-même cet exercice : un «grand» sphinx étant composé de quatre sphinx «plus petits», pourquoi ces «petits» ne feraient-ils pas eux aussi des «petits» ?

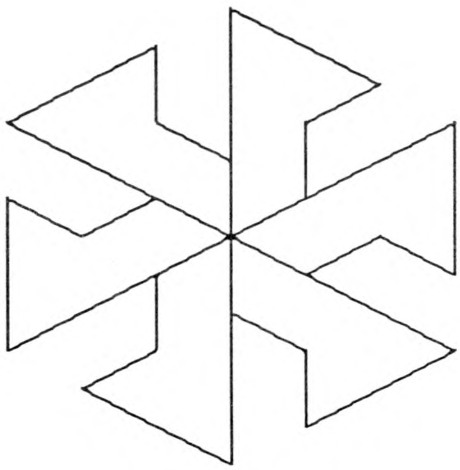


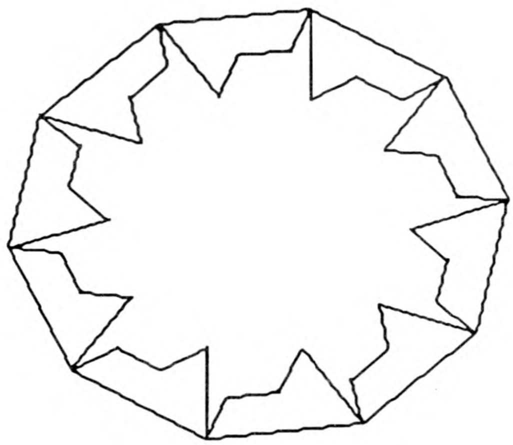
On voit poindre ici la récursivité ! (cf. p 99)

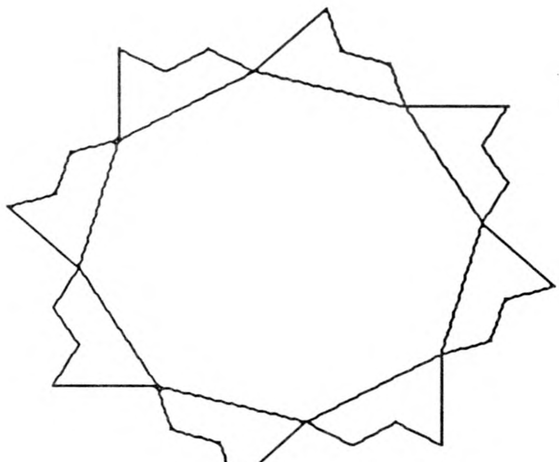
Des petits sphinx qui font la ronde

Nous avons proposé ensuite d'autres figures sur le même thème, mais plus faciles et qui ont eu plus de succès !

Cependant, nous avons pu constater qu'il restait toujours une difficulté pour différencier les deux sphinx et trouver la bonne orientation de la tortue. Ce travail d'analyse de figures est donc utile pour renforcer la capacité à reconnaître les éléments symétriques d'une figure.







Des vagues et des fleurs*

Niveau : **CE-CM**

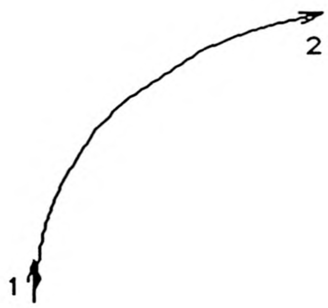
Objectifs : - Décomposition d'une figure
- Orientation
- Mesure d'angles

On peut pensé que les primitives du langage ne sont pas incitatrices au découpage en blocs signifiants car elles sont exprimées en terme de déplacements de la tortue et non de tracés.

Nous les avons donc augmenter d'un lot de procédures toutes faites (généralement appelées macro-primitives) utilisables par les enfants comme des primitives. Elles sont présentées aux élèves comme un nouveau mot du langage tortue.

Ces procédures tracent des dessins plutôt figuratifs, pour rendre plus facile la décomposition : n'est-il pas naturel de voir dans une fleur des pétales ?

Situation "Pétale"



DEMIPETALE 8

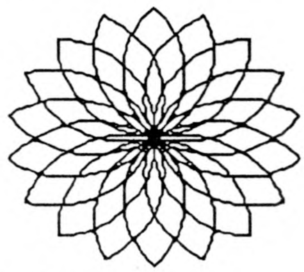
PETALE 6

* Ces activités ont été décrites dans la revue Grand N (n° 39-40)

Dans un premier temps, les enfants prennent connaissance de ces procédures en les utilisant librement. Ils doivent découvrir l'effet du paramètre qui modifie la taille de chacune d'elles et surtout le problème que pose la procédure DEMIPETALE qui n'est pas "transparente".

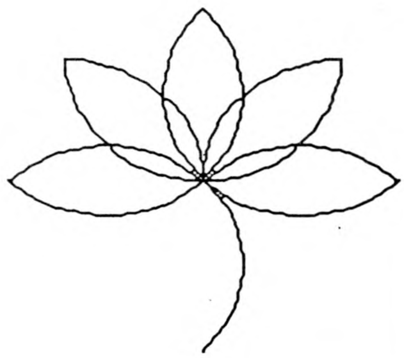
Très vite des compositions variées apparaissent.

```
POUR  COSMOS
TG 90
REPETE 6 [ PETALE 5 TD 35]
FIN
```



```
POUR DIAMANT
 REPETE 18 [PETALE 4 TD  20]
FIN
```

```
POUR ROSE
TG 90
REPETE 5 [PETALE 5 TD 45]
DEMIPETALE 5
FIN
```

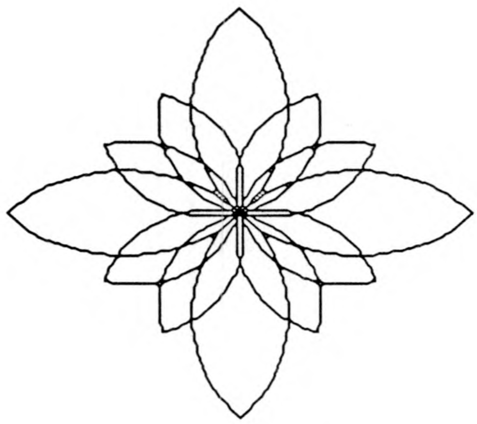


Nous avons pu constater que ces procédures étaient proposées par des enfants qui ne connaissaient jusqu'ici que les commandes TD 90 et TG 90, ce qui montre combien cette situation favorise la découverte de nouveaux angles.

Remarque

Un intéressant petit problème d'arithmétique se pose si on regarde de plus près le dessin COSMOS. Les pétales sont-ils bien "répartis", c'est-à-dire symétriquement par rapport à la verticale ? Est-ce que l'angle de 35° découvert par tâtonnements est satisfaisant ? Peut-on le déterminer plus précisément par le calcul ? (Le nombre 36 serait plus adéquat).

Dans un deuxième temps, nous avons proposés aux élèves des figures à analyser, dont voici un exemple .



DAHLIA

Le travail consiste ici à découvrir une structure répétitive et à l'exprimer correctement. On peut voir qu'il y a plusieurs décompositions possibles.

Les résultats des enfants sont assez différents ; même si tous parviennent au dessin final, le mode d'écriture de la procédure est plus ou moins bien structuré. On obtient, par exemple une description simplement séquentielle comme :

```
POUR DAHLIA
PETALE 5 TG 30 PETALE 3 TG 30 PETALE 3 TG 30 PETALE 5 TG 30 PETALE 3 TG 30
PETALE 3 TG 30 PETALE 5 TG 30 PETALE 3 TG 30 PETALE 3 TG 30 PETALE 5 TG 30
PETALE 3 TG 30 PETALE 3 TG 30
FIN
```

Ou bien, au contraire, une description avec emboîtement, comme:

```
POUR  DA
PETALE 5 TD 30
REPETE  2 [ PETALE 3 TD 30]
FIN
```

```
POUR DAHLIA
REPETE  4 [  DA ]
FIN
```

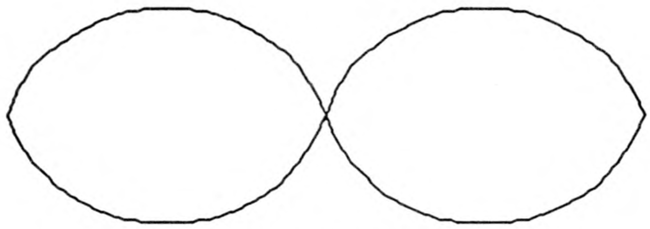
Situation " Arc "

ARC 12

Nous avons utiliser la même démarche avec ARC (n).

Après des manipulations libres, nous avons proposé le problème suivant :

Des lunettes



LUNETTES

La décomposition de ce dessin en modules ARC a été bien réussie par tous.

Nous avons également proposé aux élèves, le dessin suivant :

Des vagues

VAGUE

"Peut- on réaliser cette vague avec la procédure ARC ? "

Il faut signaler que ce problème est difficile et n'a pas été résolu, en général .
Après quelques échecs, un enfant a proposé cette solution, vite reprise par tous:

```
POUR VAGUE
ARC 8
LC ARC 8 TD 180 BC
ARC 8 TD 180
FIN
```

"Maintenant que nous avons la procédure VAGUE, peut-on faire des vagues ?".

VAGUES

Après la recherche précédente et en utilisant la même idée que dans la procédure VAGUE, les enfants résolvent assez facilement ce nouveau problème. Ils écrivent, par exemple :

```
POUR VAGUES
REPETE 4 [VAGUE 4 LC ARC 4 BC]
FIN
```

*****A l'usage des enseignants**

Voici les procédures qui produisent les modules DEMIPETALE, PETALE, et ARC*

.

POUR PETALE :L TG 45 DEMIPETALE :L TD 90 DEMIPETALE :L TD 135 FIN	POUR DEMIPETALE :L TD 3 REPETE 15 [AV :L TD 6] TG 3 FIN	POUR ARC :L TD 20 REPETE 15 [AV :L TD 10] TG 170 FIN

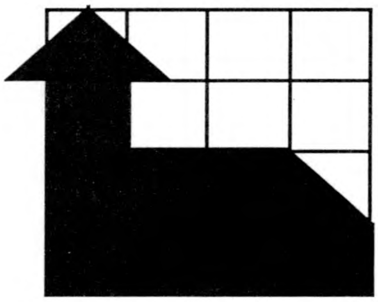
* Les calculs savants qui ont conduit à l'orientation de la tortue dans chacune de ces procédures sont largement expliqués dans le chapitre "Cercles et arcs de cercle" .

Des figures pleines

Niveau : CP - CE

Objectif : Décomposer une «surface géométrique» à l'aide de surfaces élémentaires.

Nous avons rassemblé ici des exemples dans lesquels les figures à réaliser sont "pleines", c'est-à-dire que l'intérieur du contour est coloré. Par exemple :



Pour cela, il est nécessaire de fournir aux enfants ce qu'on appelle des macro-primitives, c'est -à-dire des procédures que l'enseignant aura enregistrées au préalable et qui augmentent ainsi le vocabulaire de la tortue.

Chacune de ces macro-primitives produit sur l'écran une surface "pleine" que nous appelerons module (Vous trouverez p.90 les procédures utilisées dans ces exemples).

Le fait que la figure soit pleine, permet de découper la «surface» à réaliser comme on le ferait avec une surface en carton.

Dans les petites classes, les enfants ont effectivement travailler sur des modèles en carton. Cette manipulation leur a permis de concrétiser le découpage d'une procédure en sous-procédures.

On peut faire ce travail de deux façons différentes :

1. Demander aux enfants de choisir un découpage de la figure puis de le faire

ou bien

2. Demander aux enfants de reconstituer la surface à réaliser à l'aide de morceaux de carton déjà découpés et ayant une forme imposée (puzzle).

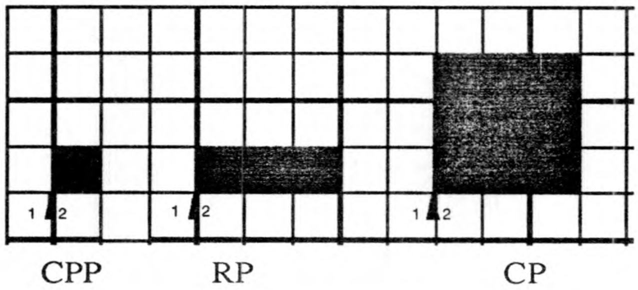
Dans les classes, nous avons exploitées des situations différentes, en fonction des modules fournis aux enfants.

(Cf. les numéros 42 et 44 de la revue Grand N)

Situation 1

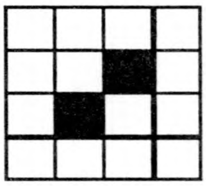
Les modules sont les suivants

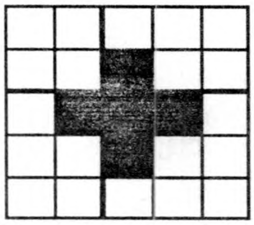
CPP : carré plein petit
RP : rectangle plein
CP : carré plein

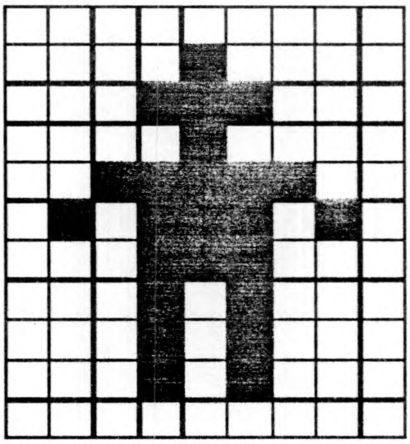


Avant tout, il est indispensable de faire découvrir aux élèves que, sur chacun de ces modules, la position de la tortue est essentielle.

Dans un premier temps, pour se familiariser avec ces modules, ils les utilisent librement pour inventer des figures. On leur impose ensuite des modèles, dont voici quelques exemples :





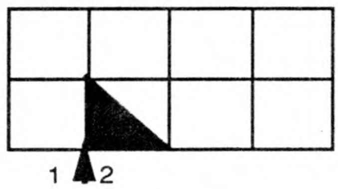


Situation 2

Les seuls modules autorisés sont :

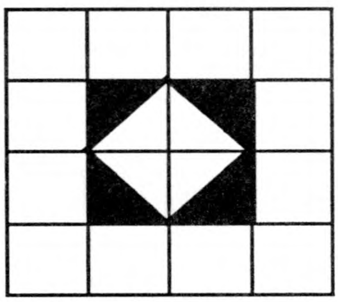
CP : carré plein

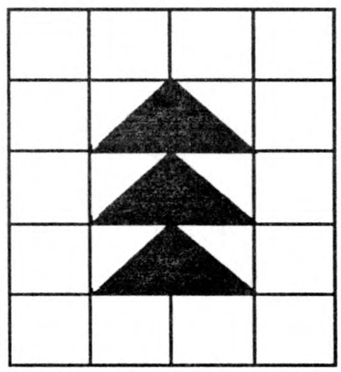
T : triangle plein

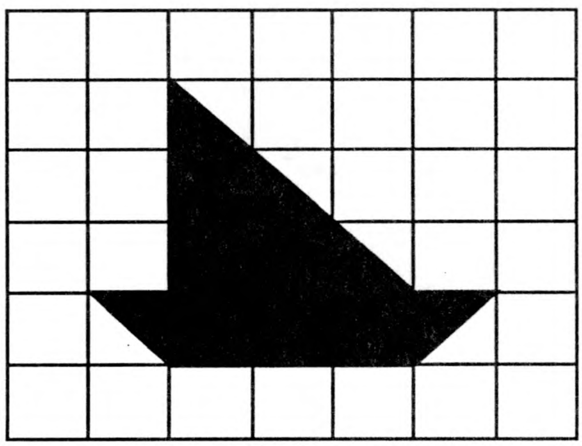


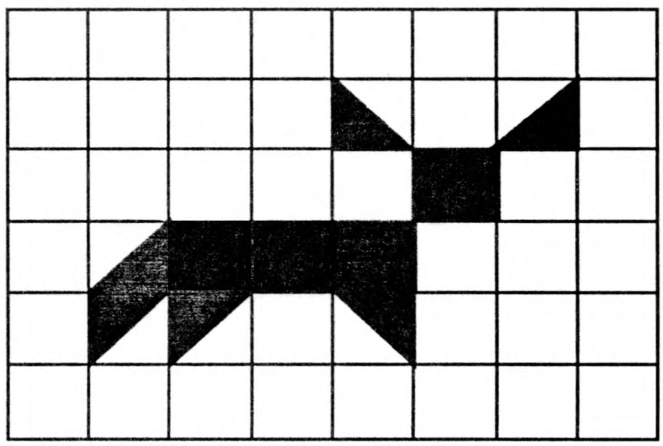
T

Voici quelques dessins facilement réalisables.







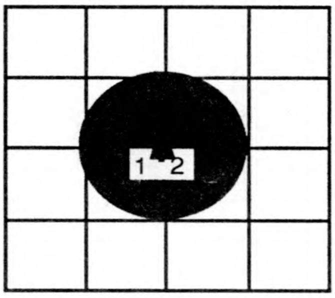


Situation 3

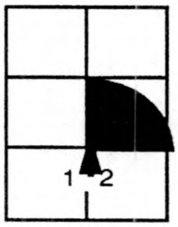
On enrichit les formes possibles grâce à deux nouveaux modules :

C : cercle plein

QC: quart de cercle plein

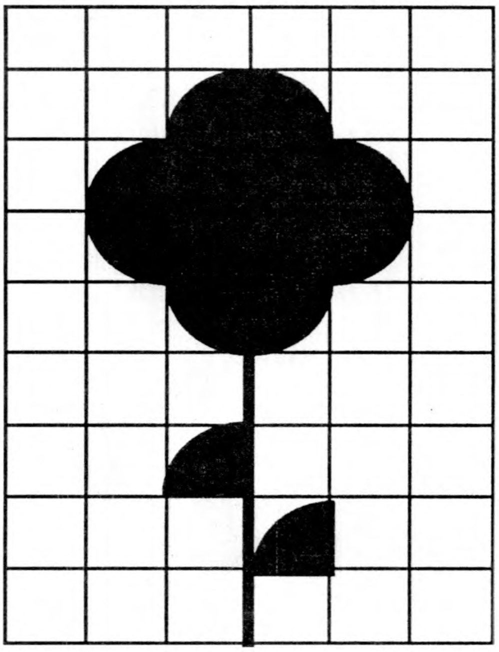


C

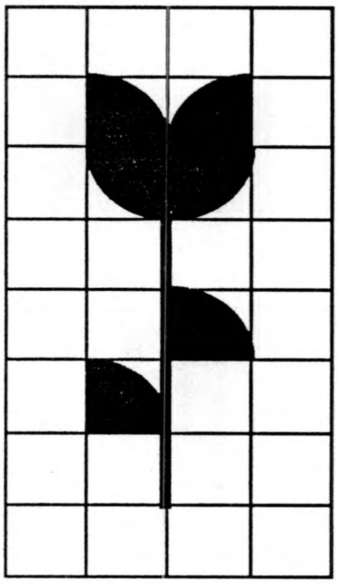


Q C

On peut alors dessiner, par exemple :



COQUELICOT



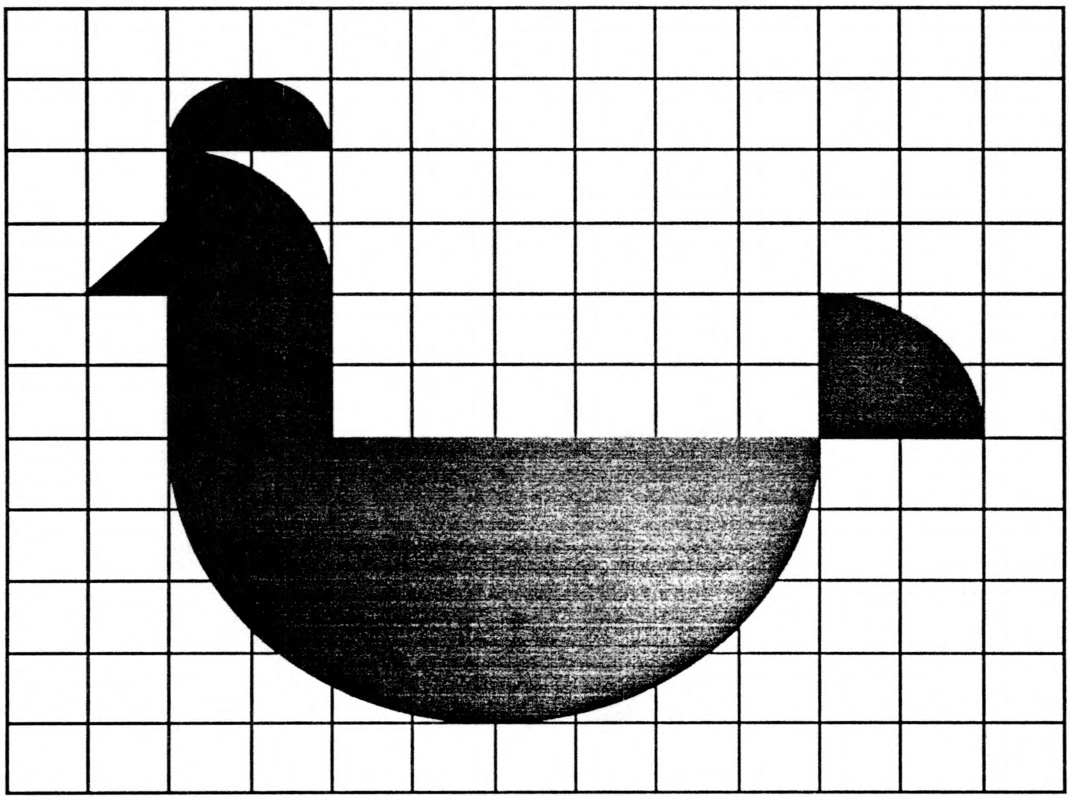
TULIPE

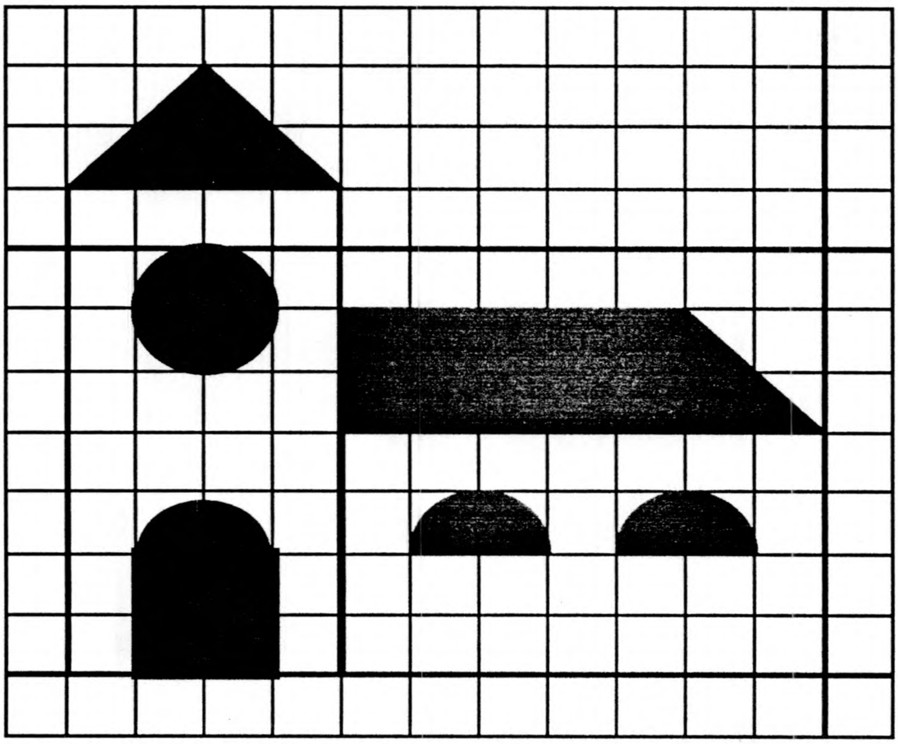
Situation 4

Il est possible de disposer des mêmes formes, mais avec un paramètre, c'est-à-dire qu'on peut en faire varier la taille ; nous avons créé, par exemple:

TR n	triangle rectangle isocèle plein (n = côté de l'angle droit).
QCR n	quart de cercle plein de rayon n.
CR n	cercle plein de rayon n.
RPP p q	rectangle plein de côtés p et q (RP = RPP 10 30).

On peut alors réaliser des compositions très variées. A vous d'en imaginer beaucoup d'autres !





*** A l'usage des enseignants :

Voici les procédures utilisées dans ce qui précède.

```
POUR RPP :P :Q
REPETE (:Q - 1) [AV :P RE :P TD 90 AV 1 TG 90]
AV :P RE :P TD 90 RE :Q TG 90
FIN
```

```
POUR  CPP
RPP 10 10
FIN
```

```
POUR CP
RPP 30 30
FIN
```

```
POUR RP
RPP 10 30
FIN
```

```
POUR TR :N
SI :N = 0 [STOP]
AV :N RE :N
TD 90 AV 1 TG 90
TR :N - 1
TD 90 RE 1 TG 90
FIN
```

```
POUR T
TR 10
FIN
```

```
POUR CR :R
REPETE 360 [AV :R RE :R TD 1]
FIN
```

```
POUR C
CR 10
FIN
```

```
POUR QCR :R
REPETE 90 [AV :R RE :R TD 1]
AV :R RE :R TG 90
FIN
```

```
POUR QC
QCR 10
FIN
```

CERCLES ET ARCS DE CERCLE

Un cercle-Logo qu'est-ce ?

Lorsqu'on veut faire tracer un cercle par la tortue, le plus souvent on se contente d'écrire :

REPETE 360 [AV 1 TD 1]

Il apparaît alors sur l'écran un polygone régulier ayant 360 côtés, de périmètre 360 pas.

Pour que «le cercle» ne soit pas trop long à tracer on peut aussi dessiner un polygone régulier ayant seulement 36 côtés et on écrit :

```
POUR CERCLE
REPETE 36 [AV 1 TD 10]
FIN
```

Dans la suite de ce chapitre on a choisit d'approcher un cercle par un polygone régulier ayant **36** côtés.

Tout le monde s'accorde à dire que le polygone sur l'écran est un "cercle", mais les difficultés commencent si on cherche à en savoir plus sur celui-ci.

- Où est le centre ?
- Quel est son rayon ?
- Comment est-il situé par rapport à l'orientation initiale de la tortue ?

(c'est-à-dire par rapport à la flèche $\overrightarrow{F_{12}}$)

I - LES DEFAUTS D'UN TEL CERCLE

Le centre du polygone régulier ainsi tracé sera bien sûr le centre du « cercle».

Par contre, on pense (à tort !) que le polygone défini de cette façon produit à l'écran un «cercle» de rayon R, **tangent** à la position départ et arrivée (flèche $\overrightarrow{F_{12}}$) de la tortue.

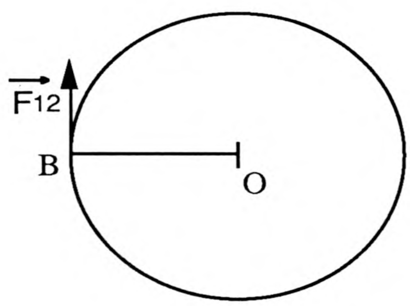
Autrement dit on pense que le polygone régulier que l'on trace a pour propriétés :

$OB \perp \overrightarrow{F_{12}}$ et $OB = R$

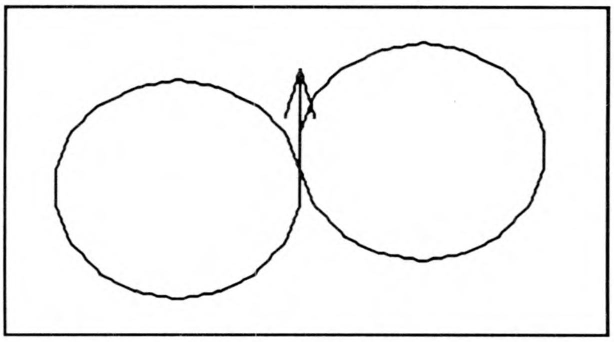
O étant le centre du polygone

B étant le point de départ et d'arrivée de la tortue

$\overrightarrow{F_{12}}$ étant la direction de départ et d'arrivée de la tortue



Or, il n'en est rien, comme on peut le voir avec l'exemple suivant :



CERCLE TD 180 CERCLE TD 180

Visiblement, les deux cercles sont tangents, mais la flèche $\overrightarrow{F_{12}}$ n'est pas la tangente commune.

Le centre du cercle n'est pas sur la demi-droite perpendiculaire en B à $\overrightarrow{F_{12}}$.

Nous appelerons "défaut α ∀ l'angle formé par BO et cette demi-droite.

Pour mieux illustrer ce défaut, faisons des approximations du cercle plus grossières.

On va représenter le cercle (C), tangent en B à $\overrightarrow{F_{12}}$, de circonférence 120 pas par différents polygones réguliers à n cotés ayant comme périmètre 120 pas.

Pour chaque cas, nous calculerons la longueur BC qui représente le diamètre du «cercle» .

BC est calculé par la formule $BC = 2\ BO = 2 \times \dfrac{L}{2 \sin \pi/n} = \dfrac{L}{\sin \pi/n}$

où L représente la longueur d'un côté du polygone régulier.

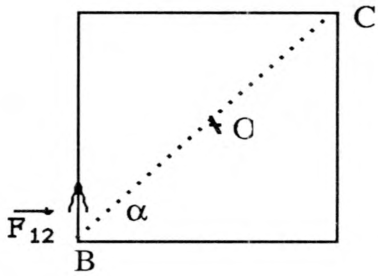
On écrit successivement :

REPETE 4 [AV 30 TD 90]

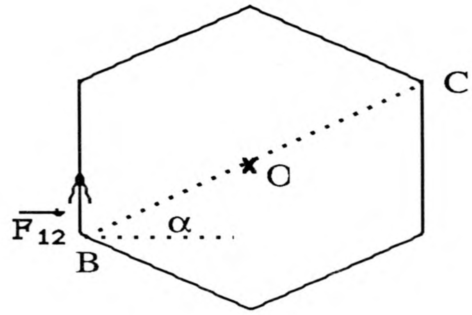
périmètre = 120 pas

α = 45°

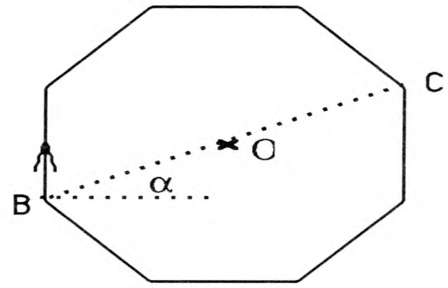
BC = 42,4 pas



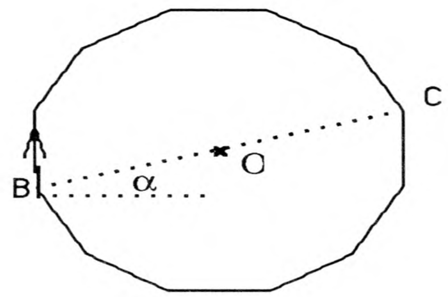
REPETE 6 [AV 20 TD 60]
périmètre = 120
α = 30°
BC = 40 pas



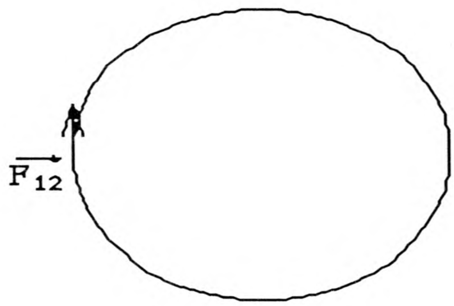
REPETE 8 [AV 15 TD 45]
périmètre = 120
α = 22,5°
BC = 39,2 pas



REPETE 12 [AV 10 TD 30]
périmètre = 120
α = 15°
BC = 38,6 pas



REPETE 36 [AV 3.3 TD 10]
périmètre = 118,8 pas
α = 5°
BC = 37,8 pas



Nous remarquons que

- le «défaut α» diminue lorsque n augmente .
- le cercle (C) ayant pour périmètre 120 pas, son rayon devrait être $R = 120 / \pi$.

On s'aperçoit que OB n'est pas égal à R, c'est ce que nous appellerons le défaut R.

II - COMMENT NE PLUS AVOIR LE DEFAUT α.

Le problème à résoudre se ramène au suivant:

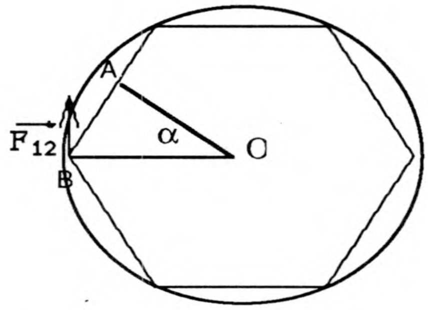
Comment construire un polygone régulier, de centre O, tel que BO soit perpendiculaire à $\overrightarrow{F_{12}}$.

Il existe plusieurs n-polygones qui répondent à la question (n fixé). Voici deux solutions.

(Pour plus de clarté, les dessins seront faits avec n=6 alors que les calculs correspondent à n=36)

1ère solution.

On choisit de tracer le polygone inscrit dans le cercle



La tortue part de B, sommet du polygone.

On a $(\overrightarrow{BA}, \overrightarrow{F_{12}}) = \alpha = \frac{1}{2} \times$ (angle extérieur du polygone)

et $(\overrightarrow{BO}, \overrightarrow{F_{12}}) = 90°$.

$L = 2\ AB = 2\ OB \sin \alpha$

pour $n = 36$ $\alpha = 5°$

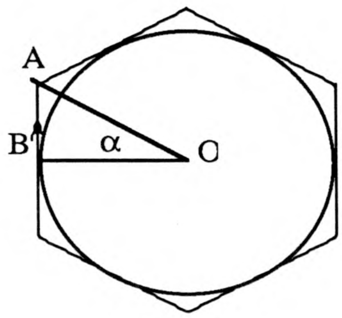
$2 \sin \alpha = 0{,}175.$

On écrira, si L désigne la longueur du côté du polygone

```
POUR CER1 :L
TD 5
REPETE 36 [AV :L TD 10]
TG 5
FIN
```

2ème solution

On choisit de tracer le polygone circonscrit au cercle



La tortue part de B milieu d'un côté et est dirigée selon ce côté.

On a $(\overrightarrow{F_{12}}, \overrightarrow{BD}) = 90°$.

$M = AB = OB \text{ tg } \alpha$

pour n = 36 on a

$\alpha = 5°$

$\text{tg } \alpha = 0{,}088$.

On écrira, si M désigne **la moitié** du côté du polygone

```
POUR CER2 :M
REPETE 36 [AV :M TD 10 AV :M]
FIN
```

III - COMMENT ATTENUER LE «DEFAUT R».

En général on préfère écrire une procédure CERCLE paramétrée par le rayon (OB=R)

Avec la 1ère solution on a : $L = R \times 0{,}175$.

On écrira

```
POUR CERCLE1 :R          avec     POUR CER1 :L
CER1 :R * 0.175                   TD 5
FIN                               REPETE 36 [AV :L TD 10]
                                  TG 5
                                  FIN
```

Avec la 2ème solution on a : $M = R \times 0{,}088$.

On écrira

```
POUR CERCLE2 :R          avec     POUR CER2 :M
CER2 :R * 0.088                   REPETE 36 [AV :M TD 10 AV :M
FIN                               FIN
```

Dans ces deux solutions, on sait en théorie calculer OB, mais les résultats numériques pris en compte par le langage Logo ne sont que des valeurs approchées.

De ce fait nous accumulons plusieurs erreurs : non seulement on remplace le cercle par un polygone mais

- la description de ce polygone se fait à l'aide de la commande AV (:R × 0,175) (ou AV :R × 0,088)

où 0,175 est une approximation de $2 \sin \frac{\pi}{36} = 0{,}1743\ldots$

où 0,088 est une approximation de $\operatorname{tg} \frac{\pi}{36} = 0{,}08748\ldots$

- dans le langage Logo la primitive AV n n'utilise que la première décimale de n, ce qui aboutit à :

$$OB = \frac{L}{2 \sin 5} = \frac{\text{tronque1 } (R \times 0{,}175)}{2 \sin 5} \cong R$$

ou bien

$$OB = \frac{M}{\operatorname{tg} 5} = \frac{\text{tronque1 } (R \times 0{,}088)}{\operatorname{tg} 5} \cong R$$

(tronque1 est la fonction qui tronque un nombre réel à partir de sa 1ère décimale).

En général le «défaut R» n'est pas perceptible ; mais on s'en aperçoit en essayant de tracer par exemple un cercle inscrit dans un carré.

Il n'est pas simple de remédier à ce défaut.

On peut, si on veut «minimiser» le défaut R, créer la fonction ARRONDI1 qui arrondira tout réel à sa 1ère décimale mais en tenant compte de la 2ème décimale. On se sert pour cela de la fonction ARRONDI qui arrondit tout réel à l'entier le plus proche.

```
POUR ARRONDI :A
SI PLP? DIFF :A ENT :A 0.5 [RENDS ENT :A] [RENDS SOMME 1 ENT :A]
FIN

POUR ARRONDI1 :A
RENDS DIV ARRONDI :A * 10 10
FIN
```

Les procédures CERCLE1 et CERCLE2 s'écrivent alors

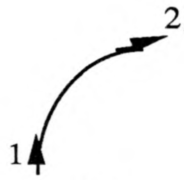
```
POUR CERCLE1 :R                  avec      POUR CER1 :L
CER1 ARRONDI1 :R * 2 * sin 5               TD 5
FIN                                        REPETE 36 [AV :L TD 10]
                                           TG 5
                                           FIN

POUR CERCLE2 :R                  avec      POUR CER2 :M
CER 2 ARRONDI :R * sin 5 / cos 5           REPETE 36 [AV :M TD 10 AV :M]
FIN                                        FIN
```

Il faut alors remarquer que CERCLE1 fait 36 erreurs d'arrondi alors que CERCLE2 en fait 72.

IV - ARCS DE CERCLE

Nous avons choisi d'écrire deux procédures associées à chacun des modules ARC1 et ARC2.



ARC1

ARC2

Dans ARC1 les positions de départ (1) et d'arrivée (2) sont tangentes à l'arc.

Dans ARC2 les positions de départ (1) et d'arrivée (2) sont dirigées selon la corde AB de l'arc AB.

Ces deux procédures ont deux paramètres

1. le rayon de l'arc : R
2. l'angle au centre de l'arc : A (en degrés)

Nous allons étudier deux cas :

A est un multiple de 10° (cas simple)

A est quelconque.

A. L'angle au centre A est un multiple de 10°.

On a $A = 10 \times D$

On écrit

1-
```
POUR ARC :D :L
TD 5
REPETE :D [AV :L TD 10]
TG 5
FIN
```

```
POUR ARC1 :A :R
 ARC QUOT :A 10 :R × 0.175
FIN
```

2- Si l'on veut se servir d'arcs de cercle avec des enfants il est plus agréable de choisir le départ et l'arrivée de la tortue dirigés selon la corde de l'arc.

La corde AB fait avec la tangente au point de départ un angle de A/2 degrés.On écrit

```
POUR ARC2 :A :R
 TG :A / 2
ARC QUOT :A 10 R × 0.175
TG :A / 2
FIN
```

Exemple.

Pour obtenir un arc d'angle au centre 70° et de rayon R ($L = R \times 0.175$) on écrit

```
POUR EXEMPLE :L
TG 30
REPETE 7 [AV :L TD 10]
TG 40
FIN
```

En effet TG 30 = TG 70 / 2 TD 5
et TG 40 = TG 70 / 2 TG 5

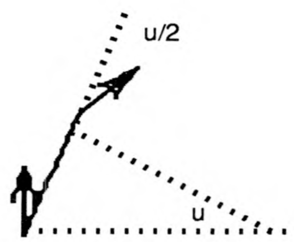
B. L'angle au centre A est quelconque.

Si A n'est pas un multiple de 10° on aura

$A = 10 * D + U$ avec $U < 10$

En LOGO :D = QUOT :A 10
:U = RESTE :A 10

Il faut donc tenir compte du reste u en traçant un «petit arc» à la fin du polygone régulier.



PETITARC 20 10

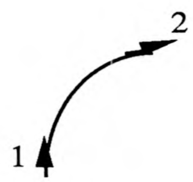
On écrit

```
POUR PETITARC :U :L
TD :U / 2
AV :L
TD :U / 2
FIN
```

```
POUR ARCELEM :U :R
PETITARC :U :R * 2 * sin (:U / 2)
FIN
```

Ce qui donne

```
POUR ARC1 :A :R
ARC QUOT :A 10 :R × 0.175
ARCELEM RESTE :A 10 :R
FIN
```




```
POUR ARC2 :A :R
TG :A / 2
ARC1 :A :R
TG :A / 2
FIN
```

UNE INTRODUCTION A LA RECURSIVITE

Introduction

Logo est un langage qui permet l'écriture de procédures récursives.

De tels procédures sont très utiles pour décrire des algorithmes complexes, pour lesquels une description séquentielle serait longue et compliquée.

Cependant, avant de pouvoir écrire une procédure récursive, il faut avoir analysé le problème à résoudre avec un point de vue particulier, que nous appellerons analyse récurrente (ou récursive). Mais il n'est pas facile d'adopter ce point de vue tant qu'on n'a pas acquis une certaine familiarité avec la récursivité.

Dans ce qui suit, nous proposons une série d'exemples pour habituer le lecteur à cette démarche. Il nous a paru plus facile de donner d'abord des exemples graphiques.

Comment peut-on découvrir qu'il existe une analyse récursive pour un dessin D ?

On aura une vision intuitive d'une analyse récursive si dans un dessin D on retrouve une ou plusieurs fois le dessin D en «plus petit».

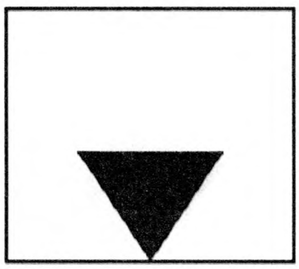
Quelques exemples sont bien connus de tous :

Un bébé tient une boîte de lait et sur cette boîte se trouve un bébé qui tient une boîte de lait... Les bébés sont de plus en plus petits jusqu'à devenir non visibles.

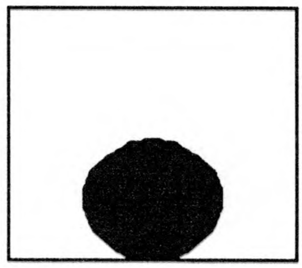
Pour comprendre comment on peut générer un dessin récursif nous allons prendre l'exemple d'une caméra de télévision qui filme sa propre image.

Des télévisions récursives

Voici deux écrans de télévision, bien sûr très stylisés: le triangle représente la tête du journaliste de la première chaîne et le disque celle du journaliste de la deuxième chaîne .



première chaîne



deuxième chaîne

Maintenant supposons que
- sur la 1ère chaîne, derrière le journaliste il y ait **un** téléviseur B,
- sur la 2ème chaîne, derrière le journaliste il y ait **deux** téléviseurs C et D.

Que se passe-t-il si on règle les téléviseurs (B, C, D) sur une des deux chaînes (1 ou 2).

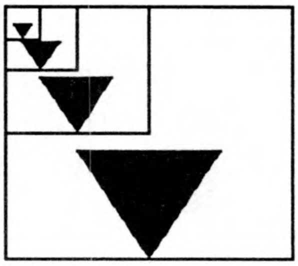
Nous allons faire les dessins correspondant à la situation :
- B est réglé sur la 1ère chaîne,
- C et D sont réglés sur la 2ème chaîne

avec l'hypothèse: «lorsque le téléviseur devient petit (à préciser) on ne le voit plus».

1. Que voit-on sur la 1ère chaîne lorsque B diffuse la 1ère chaîne ?

Sur notre écran A on verra une tête et l'écran B .

Sur B on verra ce que l'on voit sur notre écran en plus petit, c'est-à-dire une tête et l'écran B, etc...



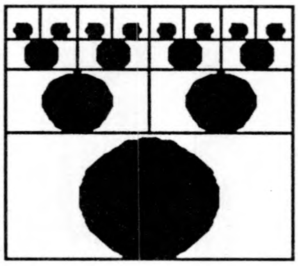
Nous obtenons ainsi un dessin ayant **1** appel récursif; le dessin est constitué d'un écran (le grand), d'une tête et d'un écran (le même que le grand en plus petit).

2. Que voit-on sur la 2ème chaîne lorsque C et D diffusent la 2ème chaîne ?

Sur notre écran A on verra une tête et les écrans C et D.

Sur C on verra une tête et les écrans C et D .

Sur D on verra une tête et les écrans C et D , etc...



Nous obtenons ainsi un dessin ayant **2** appels récursifs : le dessin est constitué d'un grand écran, d'une tête et de deux écrans (les mêmes que le grand en plus petit).

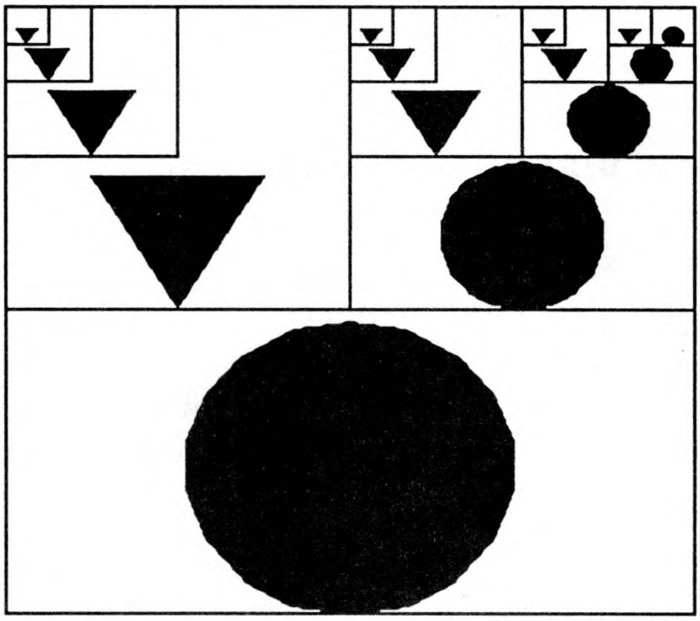
3. Variations sur le même thème.

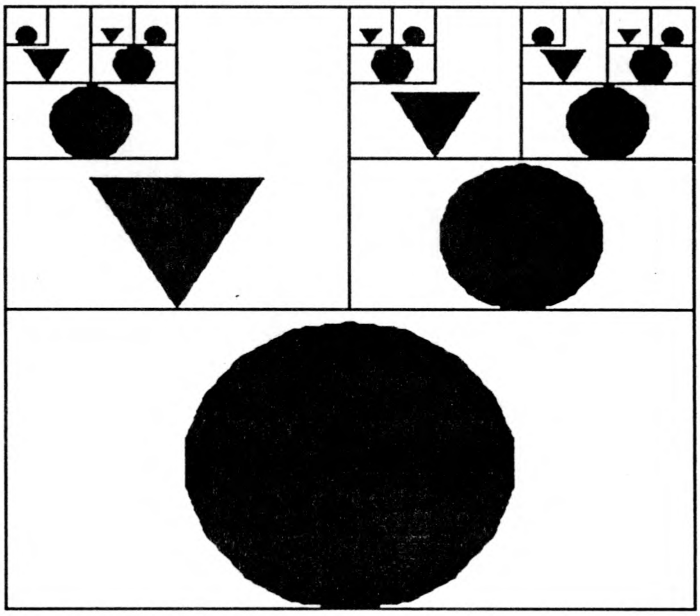
1) Que voit-on sur la 1ère chaîne lorsque B, C, D diffusent la 2ème chaîne ?

2) Que voit-on sur la 2ème chaîne lorsque B, C, D diffusent la 1ère chaîne ?

3) Pouvez-vous dessiner ce que l'on voit sur la 1ère et sur la 2ème chaîne si B diffuse la 2, C diffuse la 1 et D diffuse la 2 ?

4) Que représentent les dessins suivants ?

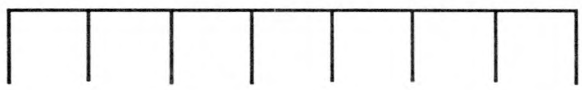




Le peigne dans tous ses états

Dans ce qui suit, nous allons étudier une figure simple en s'efforçant de la voir de différentes façons volontairement compliquées. L'usage de la récursivité est ici artificiel et purement didactique puisque cette figure peut se décrire beaucoup plus facilement par une répétition.

Voici cette figure.



Dans ce qui suit nous analyserons successivement deux modules différents associés à cette figure : Le peigne **non transparent** et le peigne **transparent.**
On appelle "transparent" un module pour lequel l'état final de la tortue est identique à son état initial.

A. Le peigne non transparent.

Le module ci-dessous est "non transparent".

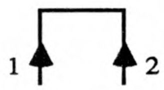
Nous allons chercher à écrire différentes procédures associées a ce module.

Nous l'appellerons *peigne(n)*, *n* représente la longueur du peigne, l'unité étant l'écartement entre deux dents (le dessin ci-dessus correspond à *peigne* (6)). On choisira l'écartement entre 2 dents égal à 10 pas de tortue.

1ère analyse :

C'est celle qui vient le plus naturellement à l'esprit :

On peut considérer que *peigne(n)* est composé de n modules identiques, égaux au module P.




```
POUR P
AV 10 TD 90 AV 10
TG 90 RE 10
FIN
```

P correspond en fait à *peigne (1)* .

peigne (n) est la juxtaposition de n modules P .

Cette description n'est évidemment pas récursive.

On écrira alors :

```
POUR PEIGNE :N
REPETE :N [P]
FIN
```

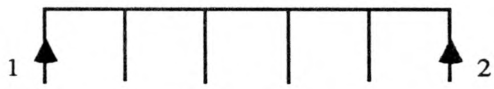
Peigne (n) est défini pour n ≥ 0 et *peigne (0)* ne produit aucun tracé.

2ème analyse.

On peut considérer que *peigne (n)* est composé de 2 modules :

Premier module : P

Deuxième module : *peigne (n - 1)*



peigne (n) est alors la juxtaposition de ces **2** modules. Ici comme pour les suites récurrentes on définit *peigne (n)* à partir de *peigne (n - 1)* , c'est-à-dire qu'on a trouvé une relation du type $U_n = f(U_{n-1})$ reliant *peigne (n)* à *peigne (n - 1)* .

On aura une définition récursive, à condition de définir le premier peigne, qui peut être *peigne (0)*

On a vu que *peigne (1)* était égal au module P .

Si l'on veut que *peigne (1)* vérifie la relation trouvée, il doit être la juxtaposition de P et de «rien». On peut donc considérer que *peigne (0)* est «rien» :

On traduira cela par :

```
SI :N = 0 [STOP]
```

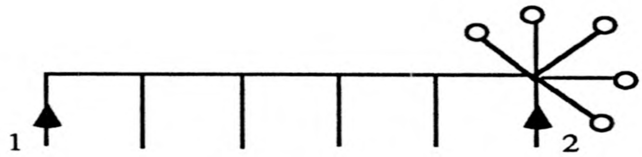
On écrira alors

```
POUR PEIGNE :N
  SI :N = 0 [STOP]
  P
  PEIGNE :N - 1
  FIN
```

Peigne (n) est défini pour n ≥ 0 et *peigne (0)* ne produit aucun tracé.

Remarque.

Voici un autre peigne de longueur n :

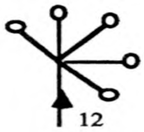


La relation de récurrence est **encore** :

peigne(n) est la juxtaposition de P et de *peigne (n -1)*

mais cette fois *peigne (1)* est la juxtaposition de P et du motif :

```
POUR MOTIF
---------------
---------------
FIN
```



Ce qui équivaut à dire que *peigne(0)* est égal à MOTIF.

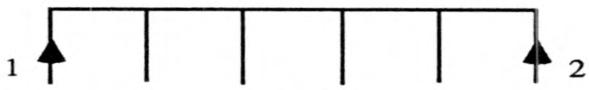
On écrira alors

```
POUR PEIGNE :N
SI :N = 0 [MOTIF STOP]
P   PEIGNE :N - 1
FIN
```

3ème analyse.

On peut considérer que *peigne (n)* est composé de 2 modules :

1er module : *peigne (n - 1)*



2ème module : P

peigne (n) est alors la juxtaposition de ces 2 modules.

Et *peigne (1)* est la juxtaposition de «rien» et de P.

Ici on a encore *peigne (0)* = «rien» et donc la même condition d'arrêt.

On écrira alors

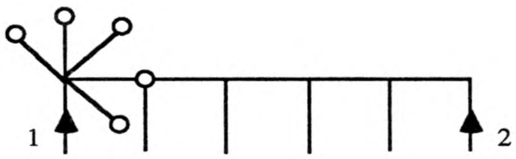
```
POUR PEIGNE :N
SI :N = 0 [STOP]
PEIGNE :N - 1
P
FIN
```

Là encore *peigne (n)* est défini pour n $\geq$ 0.

Remarques.

- Ici l'appel récursif n'est pas terminal

- le peigne avec motif de la remarque précédente ne peut pas être analysé de cette façon car il ne vérifie pas la même relation de récurrence ; avec cette troisième analyse le motif sera placé comme ci-dessous :



et il s'écrira

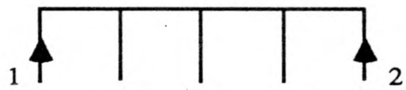
```
POUR PEIGNE :N
SI :N = 0 [MOTIF STOP]
PEIGNE :N - 1
P
FIN
```

4ème analyse.

On peut considérer que *peigne (n)* est composé de 3 modules :

1er module : P

2ème module : *peigne (n - 2)*



3ème module : P

Cette fois la relation de récurrence va de 2 en 2 : *peigne (n)* est défini à partir de *peigne(n -2)*, il faudra donc deux valeurs initiales pour définir entièrement *peigne (n)*, par exemple *peigne (0)* et *peigne (1)* :

* *peigne (2)* est la juxtaposition de P, de «rien» et de P donc *peigne (0)* sera «rien»
* *peigne (3)* est la juxtaposition de P, de P et de P, donc *peigne (1)* sera P.

On écrira alors

```
POUR PEIGNE :N
SI :N = 0 [STOP]
SI :N = 1 [P STOP]
P
PEIGNE :N - 2
P
FIN
```

Remarques.

- là encore la récursivité n'est pas terminale,
- aucun des deux peignes avec motif ne peut être analysés de cette façon.

5ème analyse.

On peut considérer que *peigne (n)* est composé de deux peignes :

- lorsque n est pair (**n = 2p**)

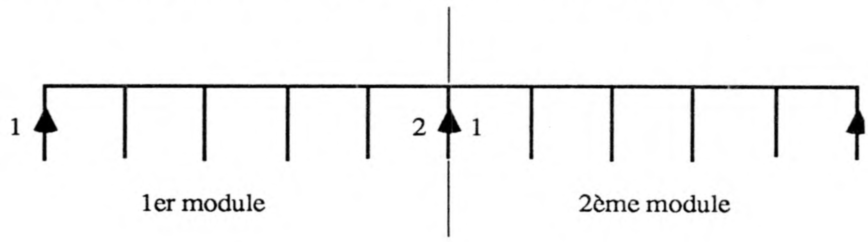
deux peignes de longueur **p**

- lorsque n est impair (**n = 2p + 1**)

deux peignes l'un de longueur **p**

l'autre de longueur **p + 1**

peigne (n) est la juxtaposition de *peigne (p)* et de *peigne (n - p).*



Dans cette analyse on a :

peigne (3) est la juxtaposition de *peigne (1)* et de *peigne (2)* ;

peigne (2) est la juxtaposition de *peigne (1)* et de *peigne (1)* ;

peigne (1) est la juxtaposition de *peigne (0)* et de *peigne (1).*

Le programme va tourner «en rond» puisque *peigne (1)* est défini à partir de *peigne (0)* et de *peigne (1)* c'est-à-dire de lui-même...

On voit donc qu'il n'est pas possible de définir *peigne (1)* de cette manière , il faudra donc le prendre comme une valeur initiale .

On écrira alors

```
POUR PEIGNE :N
SI :N = 1 [P STOP]
PEIGNE QUOT :N 2
PEIGNE :N - QUOT :N 2
FIN
```

Remarques.

- *peigne (n)* est défini pour n ≥ 1.
- La procédure PEIGNE comporte deux appels récursifs.

6ème analyse.

On peut aussi utiliser deux relations différentes selon que n est pair ou impair.

Si n est pair on considère que *peigne (n)* est la juxtaposition des deux peignes de longueur n/2.

Si n est impair on considère que *peigne (n)* est la juxtaposition de P et de *peigne (n - 1)*.

En fait si n est pair on a la 5ème analyse et si n est impair on a la 2ème analyse.

On a alors :

peigne (3) est la juxtaposition de P et de *peigne (2)* ;

peigne (2) est la juxtaposition de *peigne (1)* et de *peigne (1)* ;

peigne (1) est la juxtaposition de P et de *peigne (0)*.

On amorcera donc cette relation simplement par la définition de *peigne (0)* = rien et on écrira alors :

```
POUR PEIGNE :N
SI PAIRE? :N [PEIGNE :N / 2 PEIGNE :N / 2] [P PEIGNE :N - 1]
FIN
```

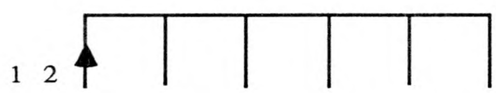
avec la définition de la fonction booléenne PAIRE?

```
POUR PAIRE? :N
SI EGAL? RESTE :N 2 0 [RENDS VRAI] [RENDS FAUX]
FIN
```

B. Le peigne transparent*.

Nous allons introduire une modification importante qui concerne la position finale de la tortue: nous allons dessiner le même motif mais nous voulons que la procédure PEIGNE soit transparente c'est-à-dire que l'état initial de la tortue soit le même que l'état final.

Par exemple :

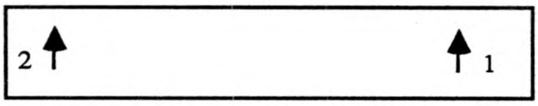


1ère analyse.

peigne (n) se décompose en n + 1 modules :

- **n** modules égaux à P
- **un** module servant à ramener la tortue au départ :

* Attention bien qu'il ne s'agisse pas du même module nous continuerons à l'appeler peigne (n). Ne vous mélangez pas les peignes !



LC TG 90 AV :N * 10 TD 90 BC

On écrira alors

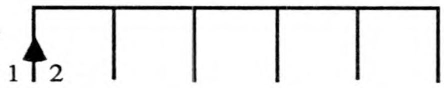
```
POUR PEIGNE :N
REPETE :N [ P ]
LC TG 90  AV :N * 10 TD 90 BC
FIN
```

2ème analyse.

peigne (n) se décompose en 3 modules.

1er module : P

2ème module :*peigne (n - 1)*

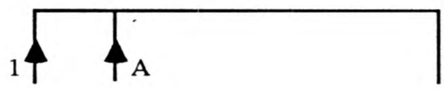


3ème module : T


```
POUR T
LC TG 90 AV 10
TD 90 BC
FIN
```

Attention.

Puisque la procédure PEIGNE est transparente, *peigne (n - 1)* ramène la tortue là où elle était avant l'appel de *peigne (n - 1)*.



Avant l'appel de *peigne (n - 1)* la tortue est selon la flèche A donc après *peigne (n - 1)* elle se retrouve en A ; il faut ensuite la ramener selon la flèche 1 d'où la nécessité du troisième module T.

Cherchons maintenant la valeur initiale de cette relation de récurrence :

On a *peigne (1)* est la juxtaposition de P de «rien» et de T donc *peigne (0)* sera «rien».

On écrit alors

```
POUR PEIGNE :N
SI :N = 0 [STOP]
P
PEIGNE :N - 1
T
FIN
```

L'écriture de cette procédure donne souvent lieu à des erreurs dont voici les plus courantes.

Erreurs les plus courantes.

1. On a parfois tendance à écrire

```
POUR PEIGNE :N
SI :N = 0 [STOP]
P PEIGNE :N - 1
LC TG 90 AV 10 * :N TD 90 BC          * erreur ! *
FIN
```

car on n'a pas compris que la dernière ligne est exécutée autant de fois que PEIGNE a été appelé .

2. Il arrive aussi qu'on écrive :

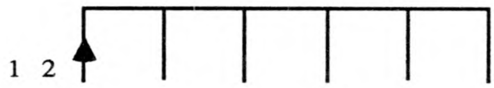
```
POUR PEIGNE :N
SI :N = 0 [LC TG 90 AV  10 * :N TD 90 BC STOP]        * erreur ! *
P PEIGNE :N - 1
FIN
```

car on pense que lorsque toutes les dents du peigne auront été dessinées, la tortue étant au bout du peigne, il suffira alors de la ramener au début du peigne ... mais ce qu'on oublie c'est que cette instruction est faite lorsque :N = 0 autrement dit AV :N est égale à AV 0

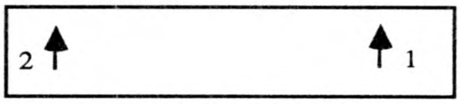
3ème analyse.

C'est la version qui consiste à voir *peigne (n)* comme *peigne (n - 1)* suivit de P. Cette analyse n'est pas intéressante ici car *peigne (n - 1)* ramène la tortue au départ. II faudrait transporter la tortue «à la fin du peigne» avant de tracer le module P puis retransporter la tortue «au début du peigne...» ce qui donnerait 4 modules :

1) *peigne (n - 1) :*

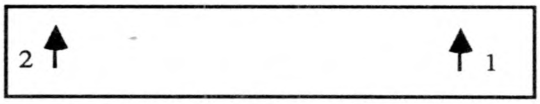


2) *côted (n - 1)* :



3) Le module P :

4) *côteg (n)*



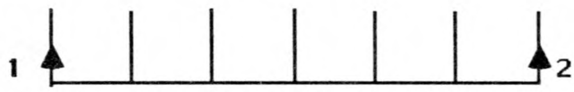
4ème, 5ème, 6ème analyses.

Même remarque que pour la 3ème analyse : ces analyses ne sont pas intéressantes dans ce cas de figures.

3. Exercices : le peigne à l'envers

(voir correction en annexe).

Si vos cheveux se sont dressés sur votre tête, à la lecture de ce qui précède, essayez le peigne à l'envers, à titre de remède!



PEIGNE A L'ENVERS

Où le peigne devient vraiment récursif

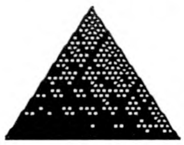
Tous les dessins qui suivent sont fabriqués à partir du même module, que nous appelerons TRAIT.

```
POUR TRAIT :N
AV :N RE :N
FIN
```

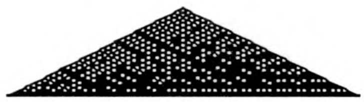

TRAIT

Il s'agit de figures pleines, réalisées à partir de "traits" suffisamment serrés, de longueur décroissante ou croissante, par exemple, diverses formes de triangles :

TRI.RECT



TRI.EQUI



TRI.ISO

(Dans les représentations qui suivent nous avons volontairement espacé les "traits" pour faire apparaître la décomposition)

Comme le module **P** dans le peigne, TRAIT se répète mais ici sa taille varie à chaque tracé. La primitive REPETE peut donc plus difficilement être utilisée .

La récursivité n'est plus un artifice didactique, mais devient très utile ici.

1er exemple.

On veut réaliser un triangle rectangle isocèle plein.

On place la tortue comme la flèche $\overrightarrow{F_{12}}$. On retrouve alors un peigne transparent à dents décroissantes cf. 2ème analyse dans "peigne" B).

On appelle TRI.RECT (N) ce triangle , N étant la longueur des côtés de l'angle droit.

On écrit alors la procédure :

```
POUR TRI.RECT :N
SI :N = 0 [STOP]
TRAIT :N  TD 90 AV 1 TG 90
TRI.RECT :N - 1
TD 90 RE 1 TG 90
FIN
```

TRI.RECT 60

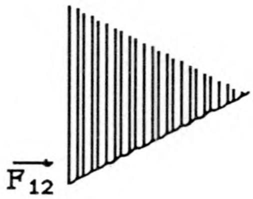
2ème exemple.

On veut réaliser un triangle équilatéral plein. On place la tortue selon $\overrightarrow{F_{12}}$. Il s'agit encore d'un peigne transparent à dents décroissantes(cf. 2ème analyse dans "peigne".B)

On l'appelle TRI.EQUI (N), N étant la longueur des côtés.

On écrit alors la procédure:

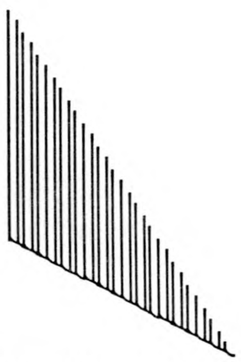
```
POUR TRI.EQUI :N
SI :N = 0 [STOP]
TRAIT :N TD 60 AV 1 TG 60
TRI.EQUI :N
TD 60 RE 1 TG 60
FIN
```



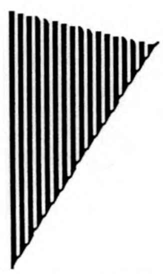
3ème exemple.

On veut réaliser un triangle isocèle plein, cet exemple est du même type que les deux précédents mais nous allons mettre l'angle au sommet du triangle isocèle comme paramètre supplémentaire.

On l'appelle TRI.ISO (N,A) : - N étant la longueur des côtés
- A l'angle au sommet



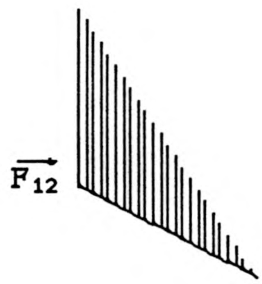
TRI.ISO 90 120



TRI.ISO 100 30

On écrit alors la procédure:

```
POUR TRI.ISO :N :A
 SI :N = 0 [STOP]
 TRAIT :N
 TD :A  AV 1 TG :A
 TRI.ISO :N - 1 :A
 TD :A  RE 1 TG :A
FIN
```



Notons que cette procédure généralise les deux précédentes. On retrouve ainsi

TRI.RECT (n) = TRI.ISO (n, 90)

TRI.EQUI (n) = TRI.ISO (n, 60).

4ème exemple.

On veut réaliser un triangle rectangle plein quelconque, en choisissant les longueurs A et B des cotés de l'angle droit. (Ci-dessous A= 30 et B=90)

- Nous pouvons choisir comme paramètres A et B, longueurs des côtés de l'angle droit.

On écrit alors la procédure :

```
POUR TR1 : A :B
 SI :B = 0 [STOP]
 TRAIT :A
 TD 90 AV 1 TG 90
 TR1  :A  / :B * (:B - 1) :B - 1
 TD 90 RE 1 TG 90
FIN
```

TR1 30 90

Pour éviter de calculer à chaque appel le rapport A / B, il est préférable d'utiliser d'autres paramètres.

- On peut choisir comme paramètre N et T

TR2 (N, T) : - N désigne la longueur d'un côté
- T désigne la tangente de l'angle aigu adjacent à ce côté

```
POUR TR2 :N :T
 SI :N = 0 [STOP]
 TRAIT :N * :T
 TD 90 AV 1 TG 90
 TR2 :N - 1 :T
 TD 90 RE 1 TG 90
FIN
```

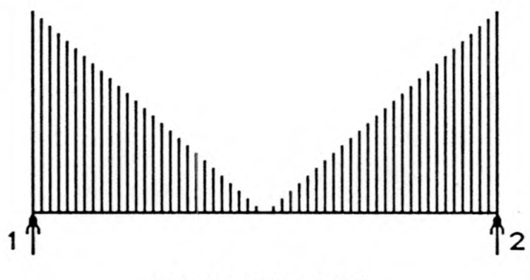
TR1 90 1/3

Dans ce cas, on peut encore utiliser la longueur des cotés en écrivant:

```
POUR TR1 :A :B
TR2 :B :A / :B
FIN
```

Ici encore on peut retrouver le triangle rectangle isocèle car TRI.RECT (N) = TR2 (N, 1).

5ème exemple.

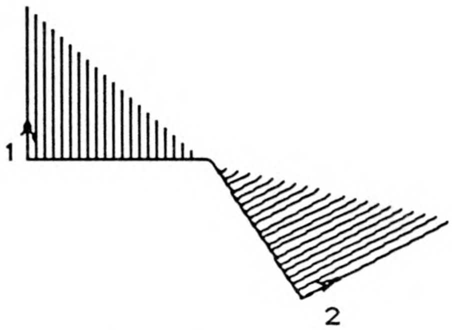


BARRIERE 80

Vous avez bien sûr reconnu sur cette figure un peigne non transparent!
On peut lui appliquer l'analyse n° 4 , ce qui conduit à la procédure:

POUR BARRIERE :N
  SI :N = 0 [STOP]
  TRAIT :N   TD 90 AV 1 TG 90
   BARRIERE :N - 1
  TD 90 AV 1 TG 90 **TRAIT :N**
FIN

6ème exemple



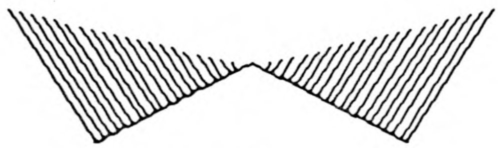
PAPILLON 60

L'analyse est la même que pour BARRIERE, on écrit la procédure:

POUR PAPILLON :N
  SI :N = 0 [TD 60 STOP]
  TRAIT :N TD 90 AV 1 TG 90
  PAPILLON :N - 1
  TD 90 AV 1 TG 90 TRAIT :N
FIN

Remarque.

Barrière (n) et papillon (n) vérifie la **même** relation de récurrence, seule la condition d'arrêt est différente.



TG 30 PAPILLON 60

7ème exemple.

Ici il s'agit d'une variante de la procédure BARRIERE .

```
POUR PORTAIL :A :B
SI :A > :B [STOP]
SI :A = :B [TRAIT :A STOP]
 TRAIT :A TD 90 AV 1 TG 90
PORTAIL :A + 1 :B
TD 90 AV 1 TG 90  TRAIT :A
FIN
```

Les paramètres A et B permettent de choisir la longueur du premier trait et du trait central.

Notons, par comparaison avec le peigne, qu'ici le portail est de longueur 2(B - A).

Remarque.

On peut dessiner avec cette procédure un triangle isocèle de hauteur h .

Il suffit de faire :

PORTAIL 0 h

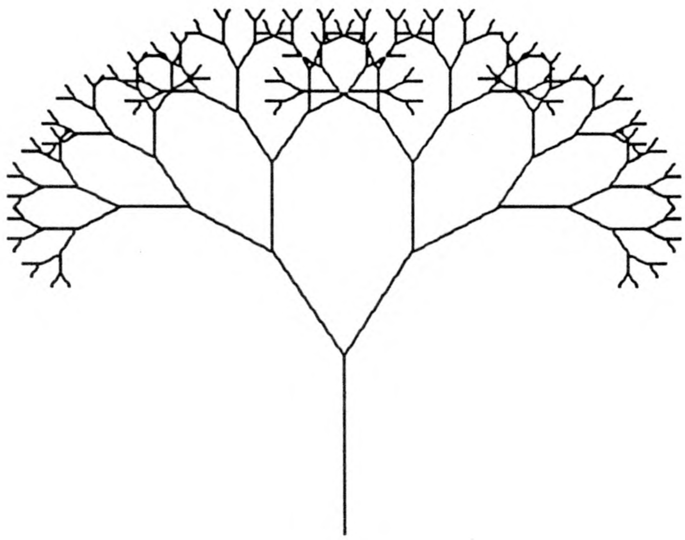
On voit qu'il faut deux paramètres à la procédure, alors qu'un tel triangle ne dépend que d'une variable.

Pour pallier cet inconvénient, on peut écrire:

```
POUR  ISOCELE :B
PORTAIL 0 :B
FIN
```

Des arbres

Nous voulons écrire une procédure réalisant le dessin suivant :



ANALYSE DU DESSIN

Nous conviendrons d'appeler ce dessin un arbre. Il est constitué d'une branche initiale et de deux arbres «plus petits» : l'arbre-gauche et l'arbre-droit.

Puisqu'il s'agit d'un dessin récursif, chacun de ces arbres «plus petits» est lui-même un arbre, seuls sa taille et sa position diffèrent.

Pour traduire en Logo cette analyse il nous faut, en plus, choisir, pour chaque partie de cette décomposition, l'état initial et final de la tortue.

CHOIX DES FLECHES DEPART ET ARRIVEE

Nous allons, comme d'habitude, représenté l'état de la tortue par une flèche.

Prenons comme position de départ, le pied de l'arbre; le plus simple est de tracer un arbre «transparent», c'est-à-dire que l'état final de la tortue sera le même que l'état initial.

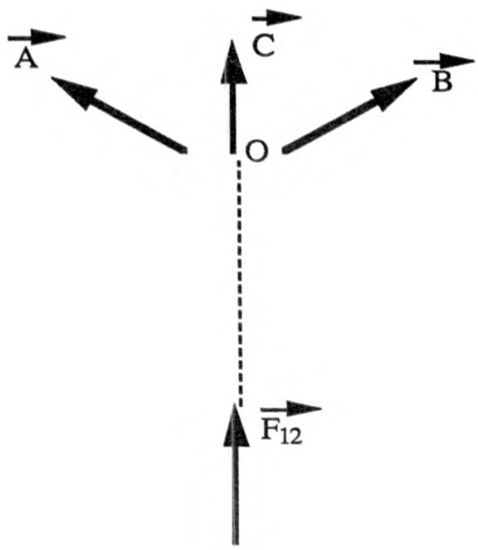
On choisit donc :
flèche départ = flèche arrivé (notée $\overrightarrow{F_{12}}$)

Ce choix étant fait, il ne faut pas oublier que l'arbre-gauche et l'arbre-droit sont alors, eux aussi, «transparents», ce qui permet de passer facilement de la fin de l'arbre-gauche au début de l'arbre-droit .

DESCRIPTION DE L'ALGORITHME

Ainsi on aura la description suivante :

0- la tortue est selon $\overrightarrow{F_{12}}$ (début et fin de l'arbre).
1- On fait un trait :
la tortue est selon $\vec{C}$.
2- On tourne vers la gauche :
la tortue est selon $\vec{A}$ (début de l'arbre-gauche).
3- On fait l'arbre-gauche :
la tortue est selon $\vec{A}$ (fin de l'arbre-gauche).
4- On tourne vers la droite :
la tortue est selon $\vec{B}$ (début de l'arbre-droit).
5- On fait l'arbre-droit :
la tortue est selon $\vec{B}$ (fin de l'arbre-droit).
6- On oriente la tortue :
la tortue est selon $\vec{C}$.
7- On fait un trait de C vers F :
la tortue est selon $\overrightarrow{F_{12}}$ (fin de l'arbre).



On a déjà trouvé la structure générale de la procédure ARBRE car on sait comment la branche initiale donne «naissance» à deux autres arbres. (en quelque sorte, on a défini la relation de récurrence entre deux arbres successifs) Il reste encore à préciser ce qu'est l'arbre initial.

CHOIX DE LA CONDITION D'ARRET

Nous pouvons décider par exemple :

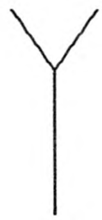
Solution 1 :
un arbre de branche **initiale** «petite» (par exemple L < 5) est composé de cette seule branche, ce que l'on traduit en LOGO par :

SI :L < 5 [AV :L RE :L STOP]

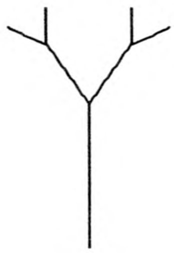
Où L désigne le paramètre représentant la longueur de la branche initiale.

Solution 2 :
On fixe ce qu'on pourrait appeler le nombre d'embranchements ou encore «la profondeur» de l'arbre à dessiner.

Ainsi un arbre de profondeur 1 sera :



un arbre de profondeur 2 sera :



etc.

Il faudra alors introduire un paramètre supplémentaire P, sorte de compteur, qui indiquera à chaque étape la profondeur de l'arbre restant à dessiner. L'arbre initial est alors l'arbre de profondeur 0, ce que l'on traduit en LOGO par :

SI :P = 0 [AV :L RE :L STOP]

CHOIX DES PARAMETRES

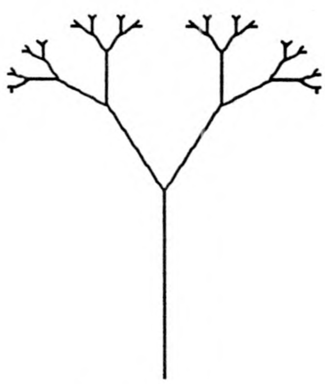
Selon la condition d'arrêt choisie on utilisera:
soit un paramètre : la longueur L de la branche initiale

soit deux paramètres : la profondeur P et la longueur L de la branche initiale.
On définit alors pour le premier cas :

arbre-gauche = arbre(L/2)
arbre-droit = arbre(L/2)

Ce qui donne la procédure :

```
POUR ARBRE1 :L
SI :L < 5 [AV :L RE :L STOP]
AV :L
 TG 30
ARBRE1 :L / 2
 TD 60
ARBRE1 :L / 2
 TG 30
RE :L
FIN
```

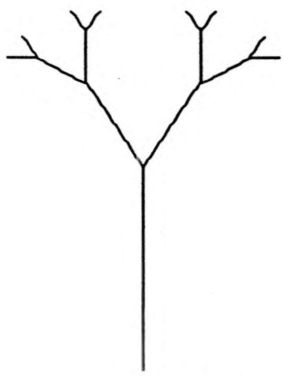


ARBRE1 80

Pour le deuxième cas :

arbre-gauche = arbre (L/2, P - 1)
arbre-droit = arbre (L/2, P - 1)

```
POUR ARBRE2 :L :P
SI :P = 0 [AV :L RE :L STOP]
AV :L
 TG 30
ARBRE2 :L / 2 :P - 1
 TD 60
ARBRE2 :L / 2 :P - 1
 TG 30
RE :L
FIN
```



ARBRE2 80 3

Remarque :

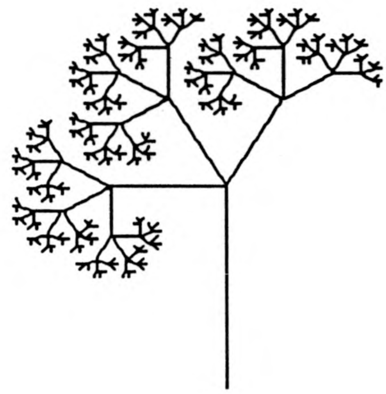
On notera la différence entre les dessins obtenus par ARBRE1 80 et ARBRE2 80 3, ce qui nous montre l'influence de la condition d'arrêt.

POUR ALLER PLUS LOIN

On veut maintenant construire un arbre à **n** branches.

La branche initiale va donner naissance à n arbres «plus petits» qui auront aussi **n** branches chacun. On écrira :

```
POUR ARBRE :L :N
SI :L < 5 [AV :L RE :L STOP]
 AV :L TG 90
REPETE :N [ARBRE :L / 2 :N TD 180 / :N]
  TG 90 RE :L
FIN
```



ARBRE 80 3

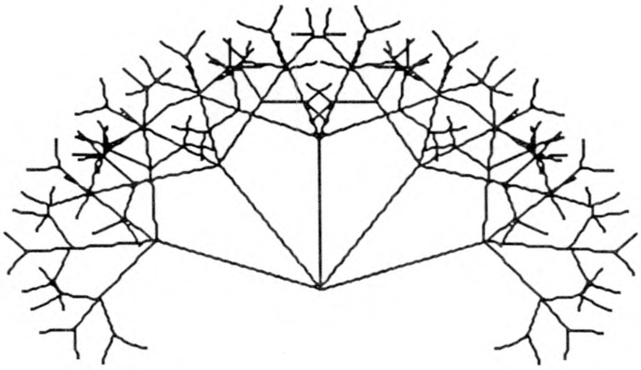
Remarque :

Nos lecteurs attentifs auront sans doute remarqué combien cet arbre est soumis à un grand vent d'Est! Quelle correction faut-il apporter aux angles pour qu'il retrouve une belle symétrie par rapport au tronc ?

L'ombelle

Nous voulons faire encore un dessin ayant N branches de longueur L, mais, cette fois-ci chaque branche donne naissance à N - 1 branches de longueur L / 2, etc.

(Les amateurs de botanique y reconnaîtront peut-être une ombelle !)



Ceci est une ombelle à cinq branches.

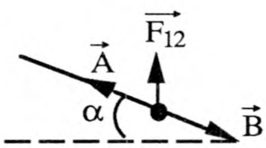
CHOIX DES FLECHES DEPART ET ARRIVEE

On choisit d'écrire une procédure transparente ayant deux paramètres N et L.

La flèche départ-arrivée $\overrightarrow{F_{12}}$ de la tortue sera choisie selon l'axe de symétrie de l'ombelle et dirigée vers les branches.

Deux branches consécutives font entre elles un angle de 180 / N degrés. Si on veut que les N branches soient, de plus, disposées symétriquement par rapport à la branche mère, il faut calculer un angle α tel que :

$\alpha = (180/2) / N = 90 / N$



(* α est l'angle avec l'horizontale de la «1ère branche» de l'ombelle)

DESCRIPTION DE L'ALGORITHME

Voici deux algorithmes pour décrire ce dessin, où vous retrouverez les deux premières analyses du peigne.

premier algorithme

Au bout de chaque branche on retrouve une ombelle de paramètres (N-1, L/2).
OMBELLE (N,L) est faite de N branches. Cette analyse est répétitive, on peut écrire:

POUR OMBELLE :N :L	* la tortue est selon $\overrightarrow{F_{12}}$ *
SI :N = 0 [STOP]	
TG 90 TD 90 / :N	* on amène la tortue selon $\vec{A}$ *
REPETE :N [AV :L OMBELLE :N - 1 :L / 2 RE :L TD 180 / :N]	* la tortue est selon $\vec{B}$ *
TG 90 / :N TG 90	* la tortue est remise selon $\overrightarrow{F_{12}}$ *
FIN	

deuxième algorithme

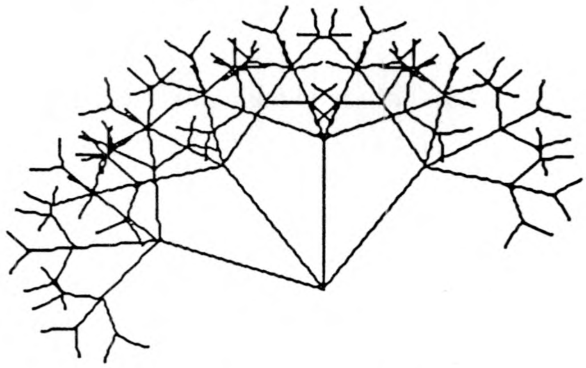
Pour mieux comprendre cette méthode, nous allons passer par une étape intermédiaire :

Nous voulons avoir la possibilité de ne dessiner qu'une partie de OMBELLE (N, L) c'est-à-dire de ne dessiner que ses I premières branches (la 1ère branche étant par exemple la branche la plus à gauche).

Nous allons écrire une procédure PARTIOMB avec 3 paramètres I, N, L.

Dans ce qui suit on aura $I \leq N$.
On choisira comme dans OMBELLE, $\overrightarrow{F_{12}}$ selon l'axe de symétrie de l'ombelle **complète.**

PARTIOMB 1 5 60



PARTIOMB 4 5 60

1ère analyse.

PARTIOMB (I, N, L) comporte I branches, il suffira donc

1. de remplacer le REPETE :N [......] du programme de OMBELLE par REPETE :I [......]

2. de ramener la tortue à sa position départ (sans se tromper, en tournant du bon angle !)

ce qui donne la procédure suivante :

```
POUR PARTIOMB :I :N :L
SI :N = 0 [STOP]
TG 90  TD 90 / :N
REPETE :I [AV :L PARTIOMB :N - 1 :N - 1 :L / 2  RE :L TD 180 / :N]
TG (180 * :I / :N)  TG 90 / :N  TD 90
FIN
```

Il est facile de voir que PARTIOMB (N, N, L) correspond bien à OMBELLE (N, L)

L'écriture de cette procédure diffère très peu de la procédure OMBELLE, puisqu'elle utilise aussi une répétition.

2ème analyse.

On peut éviter la structure répétitive précédente, en décomposant PARTIOMB (I, N, L) de la façon suivante :

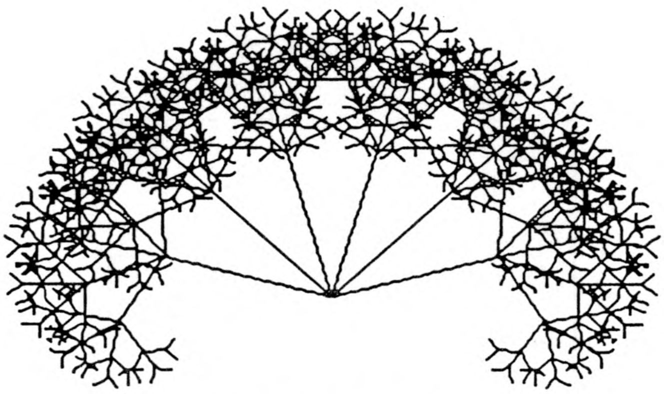
PARTIOMB (I, N, L) est formée de un trait suivi de

- 1 branche c'est-à-dire PARTIOMB (N-1, N-1, L/2)
- le reste c'est-à-dire PARTIOMB (I-1, N, L)

Il reste à écrire la procédure complètement, en s'efforçant de ne pas trop se perdre dans les paramètres et les orientations de la tortue!

```
POUR PARTIOMB :I :N :L
SI :I = 0 [STOP]
TG 90 TD 90 / :N
AV :L
PARTIOMB :N - 1 :N - 1 :L / 2
RE :L TD 90 TD 90 / :N
PARTIOMB :I - 1 :N :L
TG 180 / :N
FIN
```

On peut vérifier que PARTIOMB :N :N :L produit bien OMBELLE :N :L



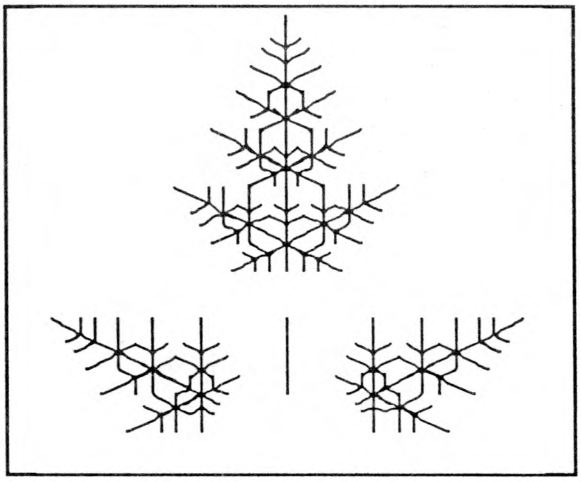
PARTIOMB 6 6 60

OMBELLE 6 60

Le sapin

Description.

Nous allons considérer qu'un sapin est constitué par
- deux branches latérales
- un «bout» de tronc
- un tête



Pour que cette description soit récursive on va dire que :
- les branches latérales sont des sapins «plus petits»
- la tête est aussi un sapin plus petit

A partir de là chacun peut dessiner le sapin qui lui convient...

ALGORITHME

Précisons cette description :
Nous appellerons sapin (N) un sapin de hauteur N.

- Les branches latérales sont symétriques par rapport à la verticale et font un angle de 60° avec celle-ci.

- Les branches latérales sont des **sapins** de hauteur N/2.

- Le bout de tronc a pour hauteur N/4.

- La tête est un **sapin** de hauteur 3N/4.

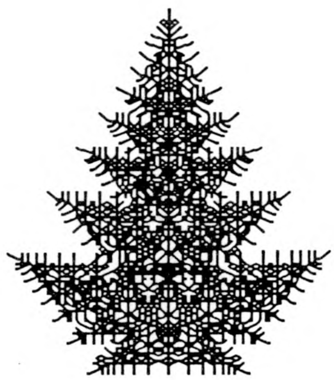
- On décide que lorsque N < 5 le sapin est constitué d'une seule tige de longueur N.

Traduisons maintenant ce qui précède en LOGO.

Comme dans les exemples précédents, nous écrirons une procédure transparente, en choisissant comme position départ et arrivée la flèche $\overrightarrow{F_{12}}$, c'est-à-dire le pied du sapin.

```
POUR SAPIN :N
   SI :N < 5 [AV :N RE :N STOP]        * condition d'arrêt*
   TG 60 SAPIN :N / 2                  * 1ère branche latérale *
   TD 120 SAPIN :N / 2                 * 2ème branche latérale *
   TG 60                               * tortue remise selon F12 *
   AV :N / 4                           * bout de tronc *
   SAPIN 3 * :N / 4                    * tête *
   RE :N / 4                           * tortue remise selon F12 *
FIN
```

On obtient des sapins de toutes tailles dont voici un exemple :

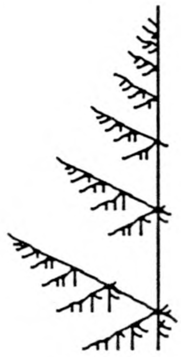


SAPIN 100

Remarque.

SAPIN est constitué de **3** appels récursifs. Pour mieux comprendre le fonctionnement de la récursivité, supprimons par exemple l'appel récursif correspondant à «la» branche latérale droite: il manquera bien sûr toute la moitié droite du sapin **mais aussi** toutes les moitiés droites des branches latérales gauches.

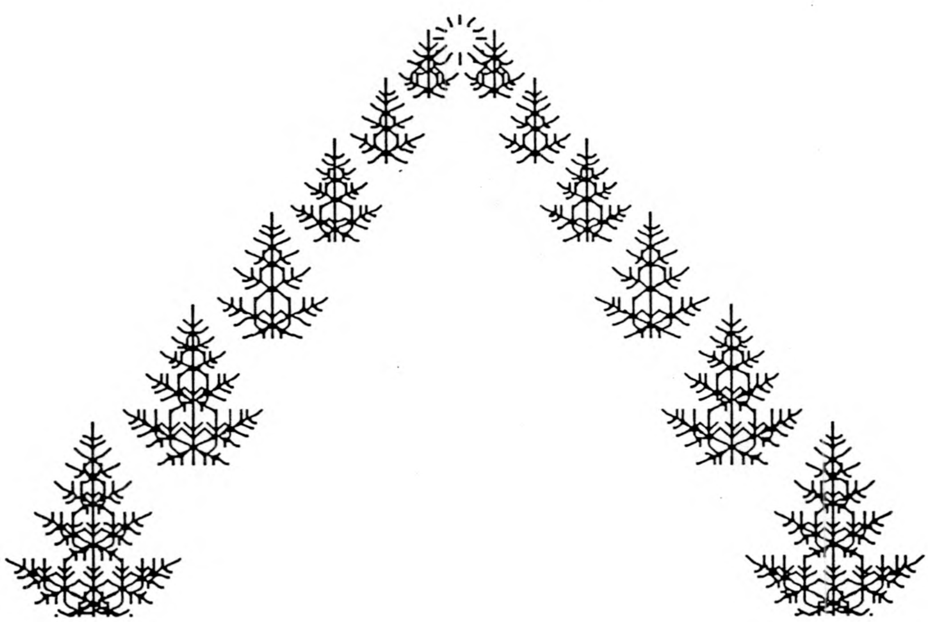
```
POUR SAP :N
  SI :N < 5 [AV :N RE :N STOP]
  TG 60 SAP :N / 2
  TD 60
  AV :N / 4
  SAP :N * 3/4
  RE :N / 4
FIN
```



SAP 120

Une allée

Pour terminer, pourquoi ne pas utiliser nos sapins pour border une allée récursivement, bien sûr!



Exemples non graphiques

Il existe de nombreux problèmes non graphiques où l'utilisation d'une procédure récursive est très utile, mais l'analyse du problème doit alors se faire sans le support de la représentation graphique, ce qui est peut -être plus difficile.

Nous allons donc faire référence, dans les exemples qui suivent, aux différentes analyse du peigne, pour tenter de faciliter le raisonnement.

I - CALCUL DES VALEURS D'UNE FONCTION

Exemple : la fonction puissance

On veut calculer x^n pour n entier positif .

On note : x^n = puiss (x,n)

Reprenons les différentes analyses du peigne (cf. page 104) à propos de ce calcul.

1ère analyse : puiss (x,n) $= x^n = \underbrace{x * x * x \ldots * x}_{n \text{ fois}}$

2ème analyse :

$x^n = x * x^{n-1}$

puiss (x,n) = x * puiss (x, n-1)

avec $x^0 = 1$

3ème analyse :

$x^n = x^{n-1} * x$

puiss (x,n) = puiss(x, n-1) * x

avec $x^0 = 1$

4ème analyse :

$x^n = x * x^{n-2} * x = x^2 * x^{n-2}$

puiss (x,n) = x^2 * puiss (x, n-2)

avec $x^1 = x$ et $x^0 = 1$

5ème analyse : si n = 2p ou n = 2p + 1

$x^n = x^p * x^{n-p}$

puiss (x,n) = puiss (x,p)* puiss (x,n-p)

avec $x^1 = x$ et $x^0 = 1$

6ème analyse : si n est paire $x^n = x^{n/2} * x^{n/2} = (x^{n/2})^2$

puiss (x,n) = puiss (x* x, n/2)

si n est impaire $x^n = x * x^{n-1}$

puiss (x,n) = x * puiss (x, n-1)

et $x^0 = 1$

cette analyse donne le programme le plus rapide:

En effet , pour x^{16} elle va calculer x^0, x^1, x^2, x^4, x^8 (5 appels récursifs) alors que la 2ème analyse calcule x^0, x^1 ... x^{15} (15 appels récursifs).

Le nombre des opérations est nettement diminué, ce qui raccourcit d'autant la durée d'exécution.

Nous n'écrirons que la procédure correspondant à cette 6ème analyse, les autres étant très simples.

Rappelons qu'une fonction ne produit pas une action mais est égale à une valeur : une fonction est une procédure qui rend une valeur (ici un nombre). Il existe en Logo un primitive qui permet de créer une procédure de type fonction, c'est la commande RENDS que nous utilisons dans ce qui suit.

```
POUR PUISS :X :N
SI :N = 0 [RENDS 1]
SI PAIRE? :N [RENDS PUISS :X * :X :N / 2] [RENDS :X * PUIS :X :N - 1]
FIN
```

avec

```
POUR PAIRE? :N
SI EGAL? RESTE :N 2 0 [RENDS VRAI] [RENDS FAUX]
FIN
```

II - TRAITEMENT DES LISTES

Dans le langage Logo, les données sont structurées en liste, c'est-à-dire en file ordonnée noté [a b c d ....] a, b, c, étant les objets de la liste.

Le traitement des données se ramènera donc le plus souvent au traitement des listes.

Soit *liste(n)* une liste.

Comme pour notre *peigne*, on peut d'une façon intuitive, décrire cette liste de deux manières :

- Description répétitive

liste (n) est une suite de **n** objets

- Description récursive

si $n \neq 0$ *liste (n)* est formée d'**un** objet suivi d'une liste ***liste (n-1)***

si $n = 0$ *liste (n)* est une suite de 0 objets, appelée **liste vide**

En s'appuyant sur cette analyse, le traitement des listes se fera en général au travers des étapes suivantes:

- traiter la liste vide
- traiter un objet
- traiter la liste privée de cet objet

Premier exemple: Enumérer les objets d'un liste

```
POUR ENUMERE :L
 SI VIDE? :L [STOP]
 EC PREM :L
 ENUMERE SP :L
FIN
```

```
ENUMERE [FRED ALAIN LUC JEAN ]
FRED
ALAIN
 LUC
JEAN
```

Deuxième exemple:

Etant donner une liste d'objets munis d'une relation d'ordre total, quel est l'élément minimum (resp. maximum) de cette liste ?

Appelons L_n cette liste et Min (L_n) son minimum.

Dans cet exemple,nous supposerons $n \geq 1$ (le cas $n = 0$, n'est pas envisageable)

On aura

- si $n = 1$ le minimum de L_n est l'unique élément de la liste
- si $n > 1$ On définit Min (L_n) à partir de Min (L_{n-1}) en comparant le premier élément de la liste L_n au minimum de L_{n-1}.

Pour une écriture plus lisible, nous avons défini au préalable deux procédures:

plus petit (a,b) est une fonction qui est égale au plus petit des deux éléments *a* et *b* .

un?(l) est une fonction booléenne égale à VRAI quand la liste *l* n'a qu'un seul élément.

```
POUR PLUSPETIT :A :B
SI :A < :B [RENDS :A ] [RENDS :B]
FIN
```

```
POUR UN? :L
SI VIDE? :L [RENDS FAUX]
SI VIDE? SP :L [RENDS VRAI] [RENDS
FAUX]
FIN
```

La procédure de recherche du minimum s'écrit alors :

```
POUR MIN :L
SI UN? :L [ RENDS PREM :L]
RENDS PLUSPETIT PREM :L MIN SP :L
FIN
```

Troisième exemple : Ordonner les éléments d'une liste

C'est le problème du tri, très souvent rencontrer en informatique et qui a reçu des solutions nombreuses. Nous en donnerons deux* :

- Tri par recherche du minimum (resp. maximum)
- Tri par insertion

Dans ces deux cas nous ferons une analyse récursive du problème, en passant de la liste L_n à la liste L_{n-1}.

Tri par recherche du minimum

Nous appelerons TRIMIN (L) la fonction égale à la liste L ordonnée.

L'algorithme consiste à rechercher le minimum de L, qui sera le premier élément de la liste triée, puis à trier la liste privée de ce minimum.

Si la liste L ne contient qu'un élément, elle est déjà triée, c'est-à-dire que dans ce cas TRIMIN (L) = L

Pour écrire cette procédure, nous utiliserons la procédure MIN(L) précédente.

Nous avons besoin de définir également une procédure SAUF(X,L) qui rends la liste L privée de l'élément X. Ce qui s'écrit en Logo :

```
POUR SAUF :X :L
SI VIDE? :L [RENDS []]
SI EGAL? PREMIER :L :X [RENDS SP :L]
RENDS MP PREMIER :L SAUF :X SP :L
FIN
```

```
POUR TRIMIN :L
SI UN? :L [RENDS :L]
RENDS MP MIN :L TRIMIN SAUF MIN :L :L
FIN
```

---

* Les deux algorithmes de tri que nous allons décrire sont très peu performants ; nous les avons choisi uniquement pour leur description assez simple. Il en existe bien d'autres beaucoup plus rapides.

On remarquera que dans cette procédure, la valeur MIN :L est utilisée deux fois dans le même appel, ce qui provoque deux fois le calcul de cette valeur. On peut éviter ceci en écrivant la procédure intermédiaire suivante :

```
POUR TRIMIN1 :X :L
RENDS MP :X TRIMIN SAUF :X :L
FIN
```

```
POUR TRIMIN :L
RENDS TRIMIN1 MIN :L :L
FIN
```

Tri par insertion.

Nous appellerons TRI.INS (L) la fonction égale à la liste L ordonnée.

L'algorithme consiste à **insérer** le premier élément de la liste L dans la liste L-1 qui est elle-même triée.

Pour écrire cette procédure, nous avons besoin de définir au préalable une procédure INSERE (X,L) qui rends la liste L dans laquelle on a inséré "à la bonne place" l'élément X.

```
POUR INSERE :X :L
 SI VIDE? :L [RENDS MP :A :L]
 SI :A < PREMIER :L [RENDS MP :A :L]
 RENDS MP PREMIER :L INSERE :A SP :L
FIN
```

La procédure de tri par insertion s'écrit alors :

```
POUR TRI.INS :L
SI UN? :L [RENDS :L]
RENDS INSERE PREMIER :L TRI.INS SP :L
FIN
```

Quatrième exemple : Permutations des listes

Permutations circulaires

Etant donné une liste, on veut écrire toutes ses permutations circulaires.

Exemple:	Soit la liste	[a b c]
on veut obtenir à l'écran		a b c
		b c a
		c a b

Nous définirons d'abord une fonction CIRC :L qui rend une liste dans laquelle le premier élément est devenu le dernier.

```
POUR CIRC :L
RENDS MD PREMIER :L SP :L
FIN
```

On voit que CIR :L fournit **une** permutation circulaire de la liste L. Pour en obtenir une autre, il suffit de faire agir de nouveau la fonction CIRC.

On aura toutes les permutations quand tous les éléments de L auront été déplacés une fois. Il faudra donc utiliser un deuxième paramètre pour arrêter l'exécution de la procédure. On peut écrire, par exemple :

```
POUR PERM1.CIRCULAIRES :L :N
SI :N = 0 [ STOP ]
EC :L
PERM1.CIRCULAIRES CIRC :L :N - 1
FIN
```

Pour initialiser le compteur N il faudra utiliser la primitive COMPTE :L qui rend le nombre d'objets d'une liste. On écrira donc :

```
POUR PERM.CIRCULAIRES :L
PERM1.CIRCULAIRES :L COMPTE :L
FIN
```

On peut éviter l'utilisation de la fonction COMPTE, en prenant comme second paramètre une liste de référence de même longueur que L, à laquelle on enlève un élément à chaque permutation. Ce qui donne les procédures :

```
POUR PERM1.CIRCULAIRES :L :REF
SI VIDE? :REF [ STOP ]
EC :L
PERM1.CIRCULAIRES CIRC :L SP :REF
FIN

POUR PERM.CIRCULAIRES :L
PERM1.CIRCULAIRES :L :L
FIN
```

Permutations d'une liste

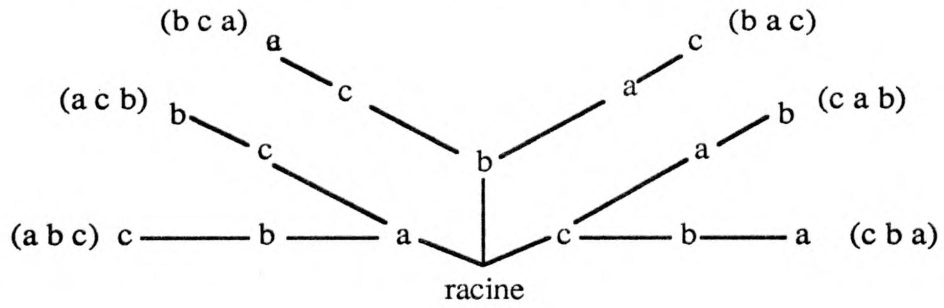
Etant donné une liste, on veut écrire toutes ses permutations.

Exemple: Soit la liste [a b c]

on veut obtenir à l'écran

a b c
a c b
b c a
b a c
c a b
c b a

On peut représenter ces permutations sous la forme d'un arbre.



On peut voir que cet arbre a la même structure que la figure OMBELLE (cf.p), mais ici les sous arbres ne sont pas identiques (à cause des lettres)

On ne peut donc pas utiliser comme dans OMBELLE, REPETE :N["appel récursif"]

En effet, les sous arbres ne sont pas identiques (les lettres changent); cependant on va pouvoir écrire une procédure PARTIPERM analogue à PARTIOMB (2ème analyse) comportant deux appels récursifs :

Description de l'algorithme :

- Chaque branche a comme nom une lettre.
- De la racine partent 3 branches. Chacun de ces départs correspond **à une** permutation circulaire de la liste initiale. Il y a donc autant de branches que d'éléments dans la liste initiale.
- On va parcourir cet arbre de la racine aux différentes extrémités.
- A chaque extrémité, on va écrire la liste des noms des branches parcourues pour aller de la racine à cette extrémité.

Parcours de l'arbre :

Il faut bien voir qu'il y a deux mouvements distincts pour se déplacer dans l'arbre.

- avancer le long d'une branche pour arriver à un sous-arbre :
  on accumule alors dans une variable DEB le nom de cette branche.
- changer de branche sans avancer, en pivotant sur place :
  on change alors la liste à permuter en CIRC :L, la variable DEB ne changeant pas.

a	b c
a	c b

—1er appel récursif

la variable DEB est augmentée de l'élément **a .**

la liste à permuter devient [b c]

b c a
b a c
c a b
c b a

– 2ème appel récursif
la variable DEB ne change pas.
la liste à permuter devient CIRC [a b c]

Choix des paramètres.

La procédure nécessite deux paramètres annexes :

-une liste DEB, vide au départ, qui sera écrite à chaque extrémité

-un paramètre pour le test d'arrêt : comme dans les permutations circulaires, on peut choisir comme paramètre le nombre d'éléments de la liste à permuter (obtenu par COMPTE :L) ou bien une liste de référence, identique au départ à la liste à permuter. Ce qui s'écrit :

```
POUR  PARTIPERM  :L :D :N
 SI VIDE? :L [EC :D ]                                  *une extrémité de l'arbre
 SI :N = 0 [STOP ]                                     *test d'arrêt
 PARTIPERM (SP :L)(MD PREM :L :D)(COMPTE SP :L)        *avancer d'une branche
 PARTIPERM (CIRC :L) (:D) (:N - 1)                     *changer de branche
FIN
```

```
POUR PERMUTATIONS :L
PARTIPERM :L [ ] COMPTE :L
FIN
```

Ou bien, en utilisant une liste de référence :

```
POUR PERMUTATIONS1 :L :D : REF
 SI VIDE? :L [EC :D ]                                  *une extrémité de l'arbre
 SI :VIDE? :REF  [STOP ]                               *test d'arrêt
 PERMUTATIONS1 (SP :L)  (MD PREM :L :D)  ( SP :L)      *avancer d'une branche
 PERMUTATIONS1 (CIRC :L) (:D) (SP :REF)                *changer de branche
FIN
```

```
POUR PERMUTATIONS :L
PERMUTATIONS1 :L [ ] :L
FIN
```

Remarque :

Les procédures de permutations décritent ci-dessus ne sont pas des fonctions; elles affichent des objets à l'écran qui ne sont pas utilisables ni mémorisés. Il faudrait les modifier légèrement pour qu'elles produisent des listes de listes.

Annexe 1 : Programmes des différentes chaînes !

Tout d'abord faisons le programme **des têtes.**

On écrit

```
POUR TETE1 :N
TD 60
TRIEQU :N                avec TRIEQU définit page 114 qui trace un triangle équilatéral plein
TG 60
FIN
```

```
POUR TETE2 :R
TD 90 AV :R
REPETE 36 [AV :R RE :R TD 10]
RE :R TG 90
FIN
```

Puis écrivons les programmes **du cadre.**

On écrit

```
POUR CADRE1 :L                  POUR CADRE2 :L
AV :L TD 90 AV :L TD 90         AV :L TD 90 AV :L TD 90
AV :L / 2 TETE1 :L / 2          AV :L / 2 TETE2 :L / 2
AV :L / 2 TD 90 AV :L           AV :L / 2 TD 90 AV :L
TD 90                           TD 90
FIN                             FIN
```

1) Lorsque notre téléviseur A est sur la 1ère chaîne et que B diffuse la 1ère chaîne, on écrit

```
POUR TELE1 :L                   * image vue sur A
SI :N < 10 [STOP]
CADRE1 :L                       * le cadre de la 1ère chaîne
TELE1 :L / 2                    * image vue sur B
FIN
```

2) Lorsque notre téléviseur A est sur la 2ème chaîne et que C et D diffusent la 2ème chaîne. On écrit

```
POUR TELE2 :L                   * image vue sur A
SI :N < 10 [STOP]
CADRE2 :L                       * le cadre de la 2ème chaîne
AV :L / 2
TELE2 :L / 2                    * image vue sur D
RE :L / 2
TELE2 :L / 2                    * image vue sur C
FIN
```

3) Variation sur le même thème.

Lorsqu'on regarde les deux procédures «cadre» on voit qu'elles se ressemblent : on peut donc écrire une **seule** procédure en rajoutant le paramètre A qui indiquera le numéro de la chaîne que diffuse le téléviseur A. On écrit

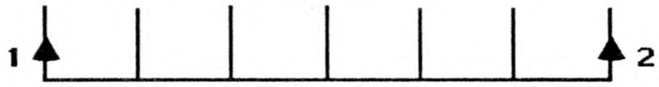
```
POUR CADRE :A :L
AV :L TD 90 AV :L TD 90 AV :L / 2
SI EGAL? :A 1 [TETE1 :L / 2] [SI EGAL? :A 2 [TETE2 :L / 2] [ ]]
AV :L / 2 TD 90 AV :L TD 90
FIN
```

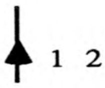
On peut aussi écrire une **seule** procédure TELE qui pourra décrire tous les cas possibles en rajoutant 4 paramètres A, B, C, D. Le paramètre A (resp B, C, D) indiquera le numéro de la chaîne que diffuse le téléviseur A (resp B, C, D).

On écrit

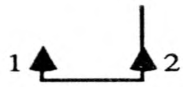
```
POUR TELE :A :B :C :D :L
SI :L < 10 [STOP]
CADRE :A :L
SI EGAL? :A 1 [TELE :B :B :C :D :L / 2]
SI EGAL? :A 2 [AV :L / 2 TELE :D :B :C :D :L / 2 RE :N / 2 TELE :C :B :C :D :L / 2]
FIN
```

Annexe 2 : Le peigne dans tous ses états

Le peigne à l'envers 

I 


```
POUR I
AV 10 RE 10
FIN
```

J 


```
POUR J
TD 90 AV 10 TG 90
I
FIN
```

L 


```
POUR L
I
TD 90 AV 10 TG 90
FIN
```

U 


```
POUR U
L
I
FIN
```

1ère analyse.

On écrit

```
POUR PEIGNE :N
REPETE :N [L]
I
FIN
```

2ème analyse.

```
POUR PEIGNE :N
SI :N = 0 [I STOP]
L
PEIGNE :N - 1
FIN
```

3ème analyse.

On écrit

```
POUR PEIGNE :N
SI :N = 0 [I STOP]
PEIGNE :N - 1
J
FIN
```

Annexe 3 : Une allée

A première vue, on pourrait analyser cette allée en la décomposant en un bord droit, un bord gauche et un soleil entre les deux. Il faudrait alors écrire une procédure différente pour chacun des bords et grâce à notre grande familiarité avec le «peigne», nous pouvons faire beaucoup mieux en utilisant la 4ème analyse.

En effet, cette allée est composée d'un sapin, d'une allée plus petite et d'un sapin.

Si on précise qu'une allée très petite (n < 15) se réduit à un soleil couchant, cela donne la procédure :

```
POUR ALLEE :N
SI :N < 15 [LC TD 90 AV 5 SOLEIL :N AV 5 TG 90 STOP]
 SAPIN :N
LC TD 40 AV :N TG 40 BC
ALLEE :N * 0.8
LC TG 40 RE :N TD 40 BC
 SAPIN :N
FIN
```

```
POUR SAPIN :N
SI :N < 5 [AV :N RE :N STOP]
TG 60 SAPIN :N / 2
TD 120 SAPIN :N / 2
TG 60 AV :N / 3 SAPIN 2 * :N / 3
RE :N / 3
FIN
```

```
POUR SOLEIL :L
 REPETE 15 [LC AV :L * 1 / 2 BC AV :L * 1 / 2 LC RE :L TD 30]
 TG 90
FIN
```

BIBLIOGRAPHIE

ARTIGUE M. et ROBINET J. (1982). Conceptions du cercle chez les enfants de 8 à 10 ans. *Rapport de Recherche - ed. IREM - Paris-Sud.*

INRP (1987). Apprentissage à la résolution de problèmes. *CRDP - Grenoble.*

MENDELSOHN P. (1984). Situation de programmation et fonctionnement opératoire chez l'enfant. *Société Française de Psychologie, Colloque «Les apports de l'intelligence artificielle et de l'automatique à la psychologie». Grenoble.*

MYX A. (1985). Logo, votre langage de programmation. *Cedic - Nathan.*

PAPERT S. (1981). Jaillissement de l'esprit : ordinateurs et apprentissage. *Paris, Cedic - Nathan.*

PIAGET J. (1975). L'équilibration des structures cognitives. *Paris, PUF.*

PIAGET J. et INHELDER B. (1947). La représentation de l'espace chez l'enfant. *PUF, Paris.*

REGGINI H.C. (1983). LOGO, des ailes pour l'esprit. *Paris, Cedic - Nathan.*

Revue Grand ℕ - *CRDP - Grenoble.*