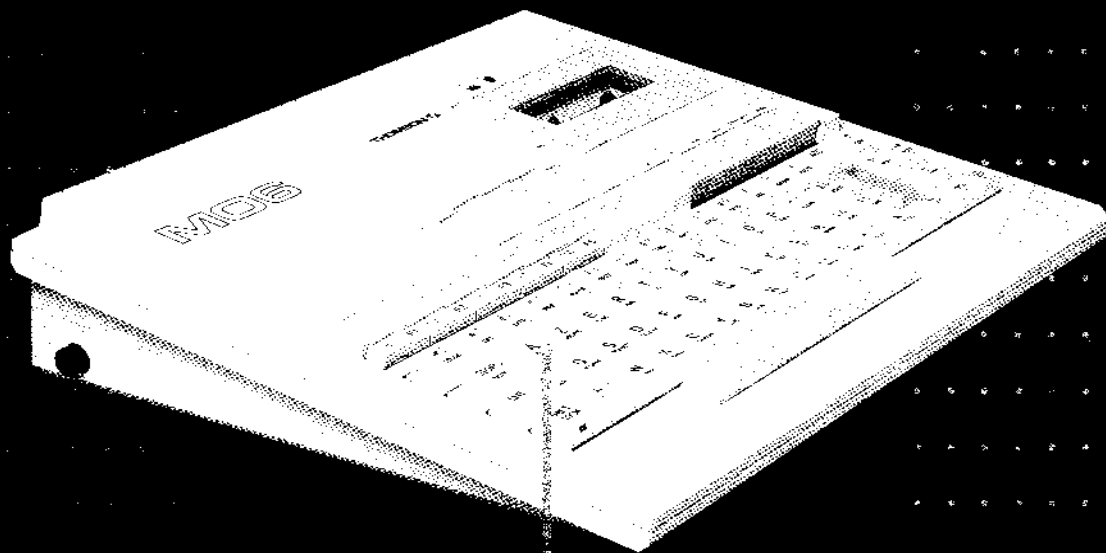


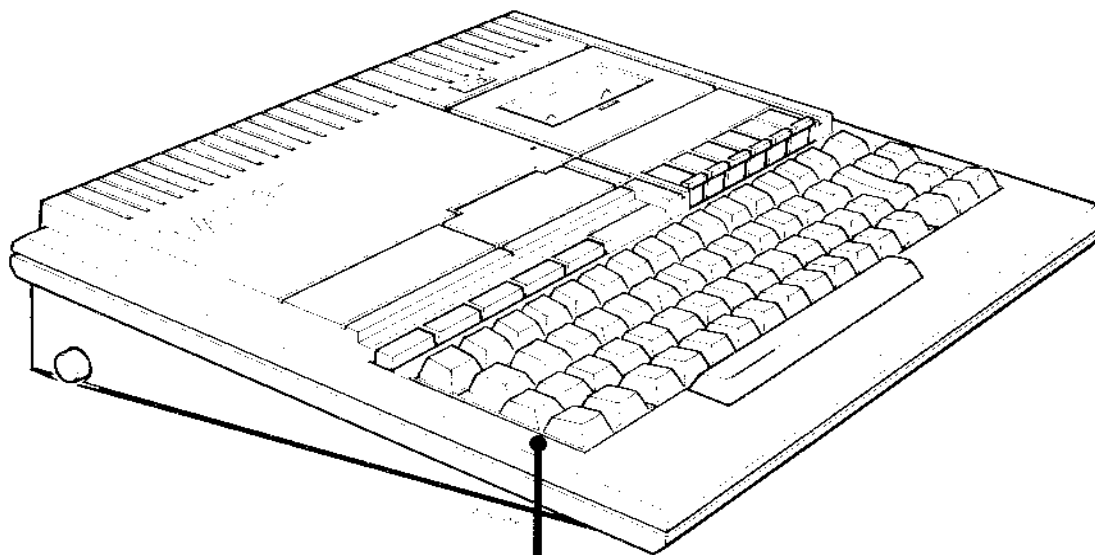
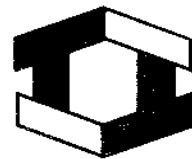
THOMSON



MO6

GUIDE

THOMSON



MO6

GUIDE

Dessins: Studio Boniface

© Thomson/SIMIV - 1986 - Edité par Cedic/Nathan

ISBN 2-7124-0182-4

Guide d'installation

Avant-propos

Ce guide vous présente le micro-ordinateur THOMSON MO6.

Il vous permettra, dans sa première partie, d'acquérir les automatismes de l'utilisation du MO6, et, dans sa deuxième partie, de vous initier à la programmation en langage BASIC.

Vous constaterez qu'il n'est nul besoin d'avoir des connaissances en informatique pour utiliser le MO6, ceci grâce à la présentation très claire des choix à effectuer dans des "menus" qui vous sont proposés spontanément.

Si vous disposez d'un crayon optique ou d'une souris, les choix s'effectuent par pointage sur la touche optique ou le pictogramme correspondant.

Vous apprécierez le confort d'utilisation de son clavier mécanique et l'agrément de sa palette de couleurs, vous permettant de choisir 16 couleurs parmi 4096 nuances possibles.

Vous n'aurez plus de fils encombrants qui traversent votre espace de travail, le MO6 possède son propre lecteur de cassettes.

Les amateurs de programmation découvriront un langage très puissant, le BASIC 128 MICROSOFT®, qui exploite la totalité de la capacité mémoire du MO6, et ouvre un large éventail de possibilités et de modes graphiques.

De plus, vous trouverez intégré au MO6 le BASIC MICROSOFT® 1.0, pour permettre de profiter de la vaste bibliothèque de logiciels développée pour le MO5.

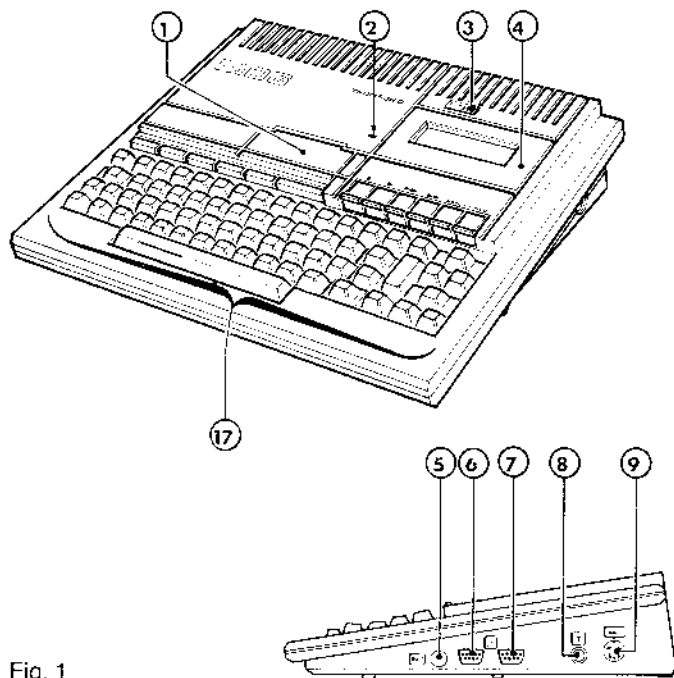
Guide d'installation

Présentation

Vous venez de sortir votre MO6 de son emballage et vous avez devant vous :

- l'unité centrale MO6,
- le cordon péritélévision.

Pour vous permettre de vous familiariser avec votre micro-ordinateur, voici une description de l'appareil :



- 1 — La trappe pour la cartouche
- 2 — Le voyant marche-arrêt
- 3 — Le compteur du lecteur de cassettes
- 4 — Le lecteur de cassettes
- 5 — Le bouton de réinitialisation
- 6 — Le connecteur de la première manette de jeu, numéro 0 en BASIC, ou de la souris
- 7 — Le connecteur de la deuxième manette de jeu, numéro 1 en BASIC
- 8 — La sortie antenne UHF PAL (modèle export)
- 9 — La prise pour le branchement du crayon optique
- 10 — Le bouton de mise en service
- 11 — Le cordon secteur
- 12 — Le connecteur d'extension
- 13 — La sortie pour imprimante
- 14 — La sortie SON
- 15 — La sortie péritélévision
- 16 — Le connecteur péritélévision
- 17 — Le clavier mécanique

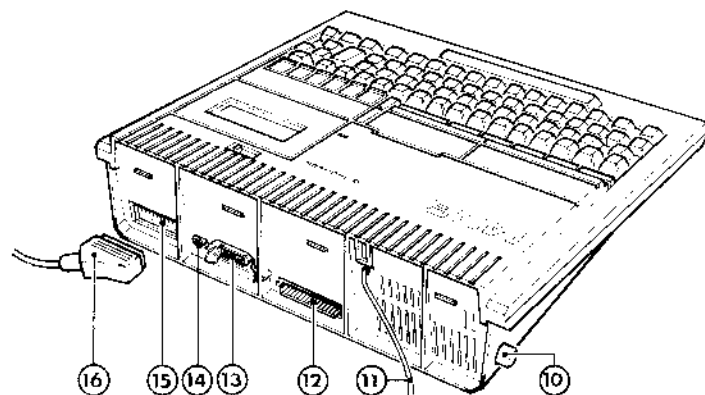


Fig. 1

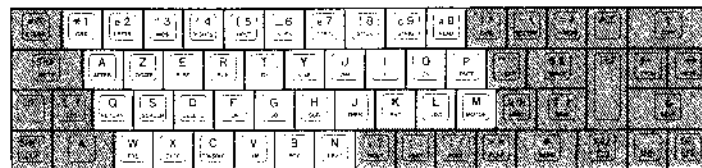
Guide d'installation

Le clavier

Le clavier se compose de quatre parties :

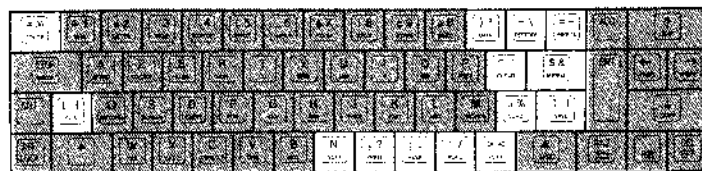
— Le clavier AZERTY avec minuscules accentuées, identique à celui d'une machine à écrire, avec la touche SHIFT-LOCK comportant un voyant rouge et permettant de sélectionner, soit le mode majuscules et chiffres lorsque le voyant rouge est allumé, soit le mode minuscules et minuscules accentuées.

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12



— Le complément du clavier alpha-numérique proprement dit, composé des touches de symbole et d'accentuation et qui n'est pas affecté par la touche SHIFT-LOCK. Les signes gravés en vert sont accessibles par la frappe simultanée de la touche ▲ et de la touche correspondante.

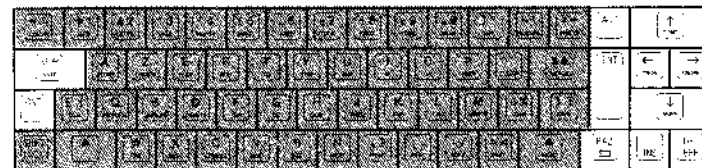
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12



— Les touches spéciales, telles les touches fléchées, la touche jaune ENT, les touches ACC, INS, EFF, RAZ, STOP, CNT, et les touches de fonctions F1 à F10, sont des touches de commande de certaines fonctions de votre MO6.

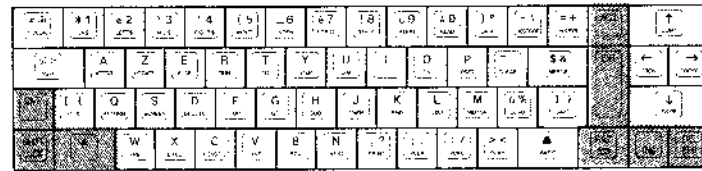
Les commandes gravées en vert sont accessibles par la frappe simultanée de la touche ▲ et de la touche correspondante.

F6 F7 F8 F9 F10 F11 F12



— Les touches comportant en bleu des instructions ou fonctions BASIC directement accessibles par l'appui simultané de la touche bleue BASIC et de la touche correspondante.

F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12



Guide d'installation

Installation et branchements

La première étape consiste à relier le MO6 à votre téléviseur ou moniteur à l'aide du câble péritélévision.

Pour cela, il est nécessaire que votre téléviseur soit éteint.

Vous raccordez n'importe quel côté du câble au téléviseur et l'autre au MO6 dans les prises correspondantes.

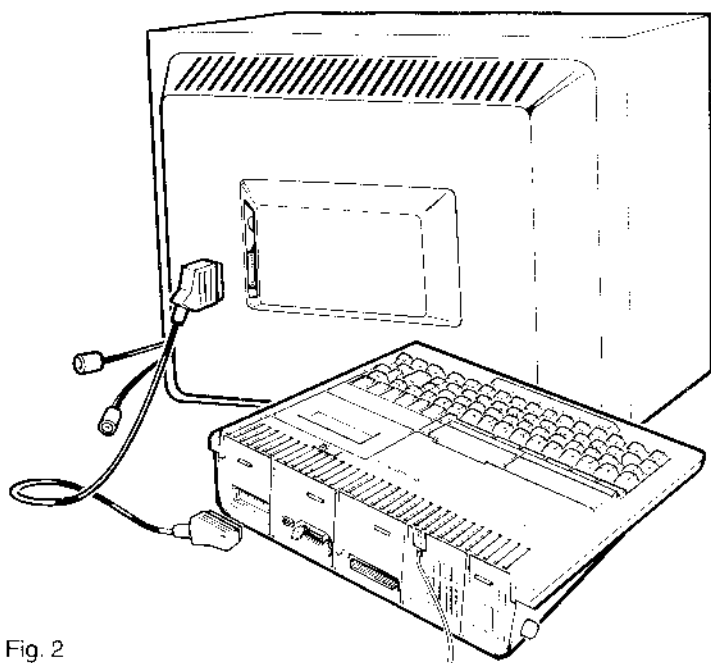


Fig. 2

Ensuite, vous raccordez la prise secteur du MO6 sur le 220 V, 50Hz.

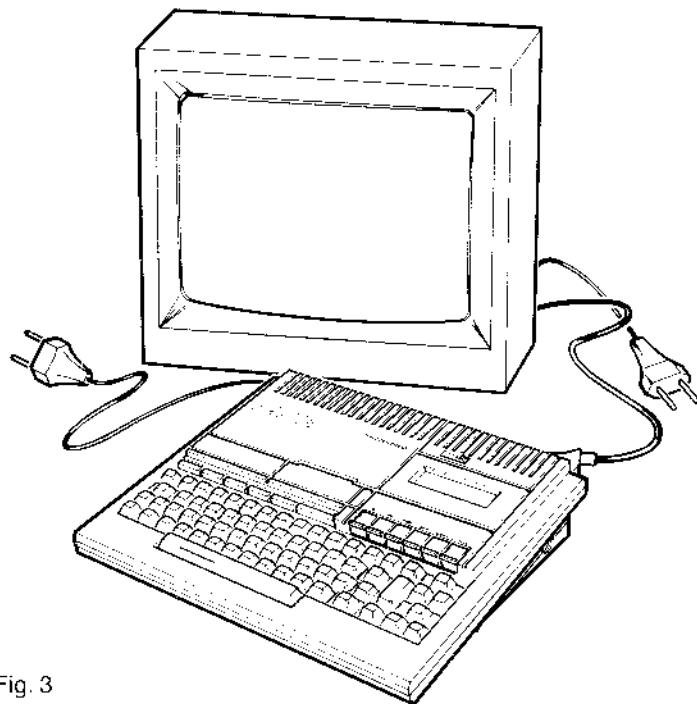


Fig. 3

Si vous possédez un téléviseur sans prise Péritélévision, vous devez utiliser un codeur modulateur SECAM, en vente chez votre revendeur, effectuer le branchement comme indiqué fig. n°4, et sélectionner sur une chaîne inutilisée le canal 36.

Guide d'installation

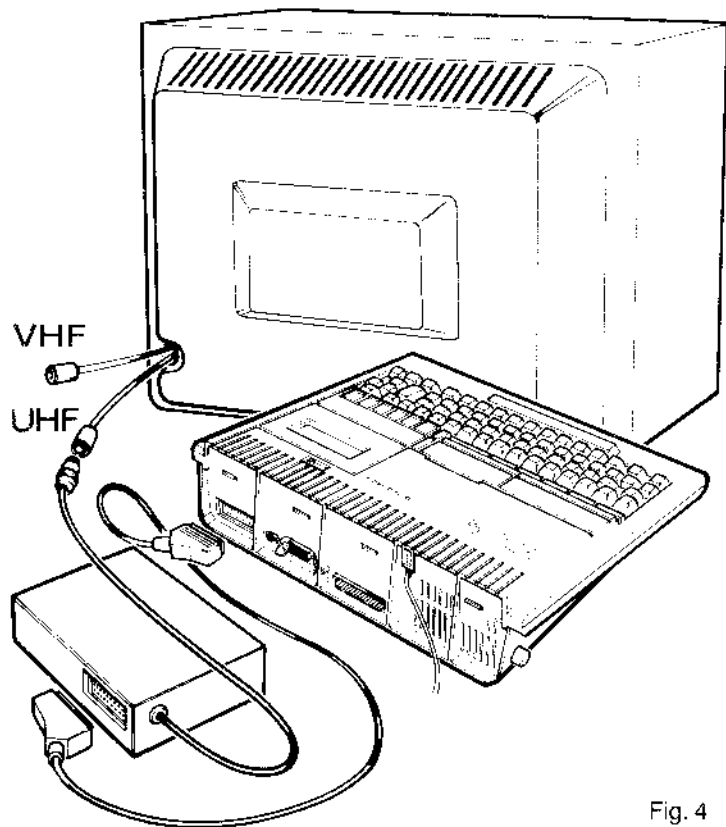


Fig. 4

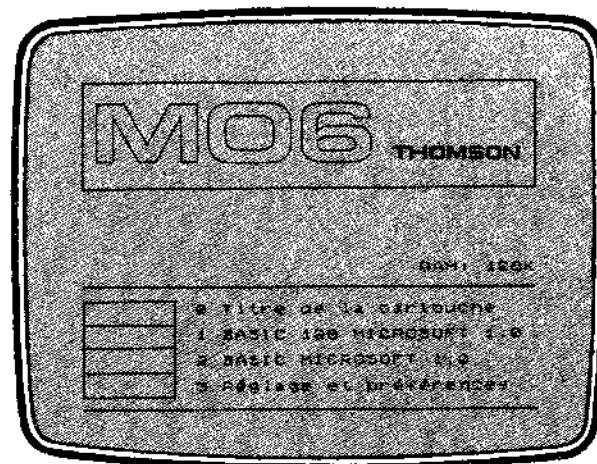
Si vous possédez des périphériques : crayon optique, lecteur de disquettes et imprimante, reliez-les aux connecteurs correspondants.

Si vous possédez une seule manette de jeu ou une souris, raccordez-la à la prise 6 de la figure numéro 1. Dans toute la suite, nous utiliserons le mot "pointer" pour désigner l'action de viser avec le crayon optique et d'appuyer sa pointe sur l'écran, ou de viser avec la souris et d'appuyer sur le bouton de gauche.

Mise en service

Vous allumez dans l'ordre votre téléviseur sur n'importe quelle chaîne, vos périphériques, puis votre MO6 en appuyant sur le bouton marche-arrêt. Un voyant vert s'allume à gauche du lecteur de cassettes.

Après quelques secondes, une image apparaît : c'est la page en-tête et son menu de sélection.



Guide d'installation

La page en-tête

La page en-tête vous indique :

- la taille de la mémoire dont vous disposez : 128 Ko,
- le titre de la cartouche de programme, si vous en avez inséré une, accessible par la touche 0 (zéro) du clavier, ou par pointage du rectangle jaune correspondant à l'aide d'un crayon optique ou d'une souris,
- le BASIC 128 MICROSOFT®, accessible par la touche 1 du clavier, ou par pointage du rectangle jaune correspondant,
- le BASIC 1.0 MICROSOFT® du MO5, accessible par la touche 2 du clavier, ou par pointage,
- l'écran de réglage et préférences, accessible par la touche 3 du clavier, ou par pointage.

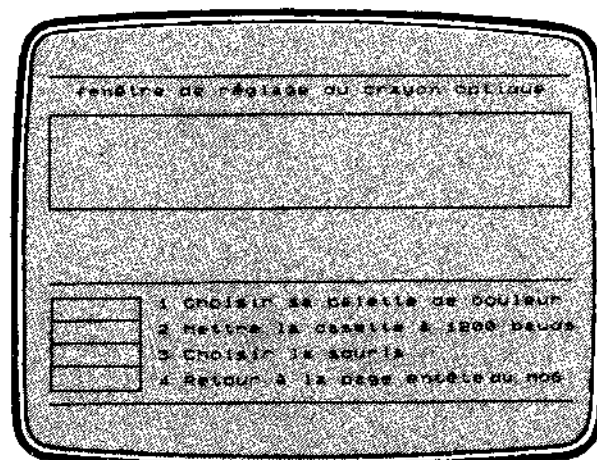
Si vous possédez un lecteur de disquette, le choix du BASIC 128 permet de charger en mémoire d'une façon automatique un programme sur disquette dont le nom est AUTO.BAT.

Le choix du BASIC 1.0 implique, pour pouvoir utiliser le lecteur de disquette, la présence à l'intérieur du lecteur de la disquette DOS.

De la même façon, le BASIC 1.0 pourra charger automatiquement un programme présent sur la disquette DOS, si son nom est : AUTO.BAT.

Réglages et préférences

Sélectionnez au clavier la touche 3 : réglages et préférences.



Un nouvel écran apparaît, comportant :

— dans sa partie supérieure, une fenêtre de réglage du crayon optique.

Si vous possédez un crayon optique, il suffit, pour procéder à son réglage, d'appuyer la pointe du crayon perpendiculairement à l'écran, une barre verticale apparaît.

Guide d'installation

Faites coïncider avec les touches ← et → cette barre avec la pointe du crayon, le réglage est terminé.

— dans sa partie inférieure, un menu :

- 1 — Choisir sa palette de couleurs
- 2 — Mettre la cassette à 1200 Bauds
- 3 — Choisir la souris
- 4 — Retour à la page en-tête

Si vous avez une souris, le menu apparaît comme suit :

- 1 — Choisir sa palette de couleurs
- 2 — Mettre la cassette à 1200 Bauds
- 3 — Choisir le crayon optique
- 4 — Retour à la page en-tête

Le MO6 vous donne en effet la possibilité de brancher à la fois une souris et un crayon optique, mais vous ne pouvez pas les utiliser simultanément .

Le MO6 a donc besoin de savoir avec quel périphérique vous voulez travailler.

A la mise en service, le MO6 détecte la présence d'une souris, et vous donne la possibilité dans le menu d'utiliser le crayon optique.

Si vous choisissez le crayon optique, le menu se modifiera pour vous donner la possibilité de choisir à nouveau la souris :

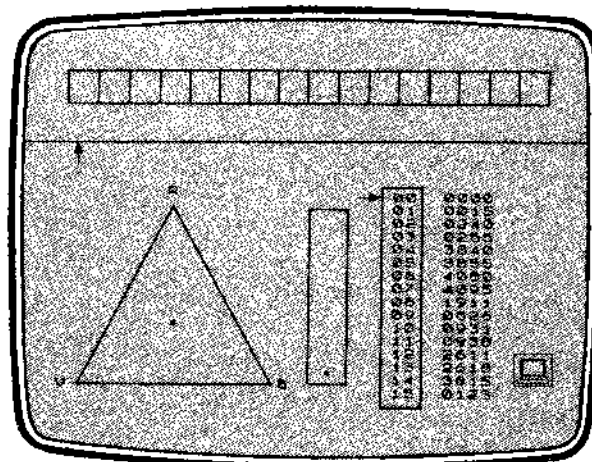
- 1 — Choisir sa palette de couleurs
- 2 — Mettre la cassette à 1200 Bauds
- 3 — Choisir la souris
- 4 — Retour à la page en-tête

Le dernier choix du menu vous permet de revenir à la page entête du MO6.

La palette de couleurs

Choisissez la palette de couleurs avec la touche 1.

Un nouvel écran apparaît :

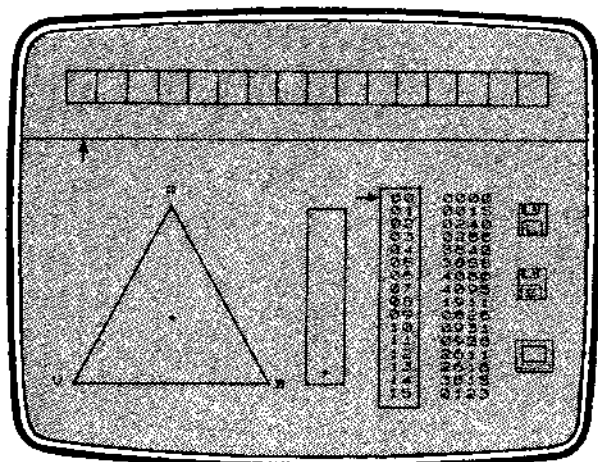


Il est constitué dans sa partie supérieure de la palette initiale du MO6, composée de 16 couleurs, dont la première, le noir, est désignée par une flèche et sert de couleur d'encadrement de la palette.

Au-dessous se trouvent un triangle de sélection de la couleur, un rectangle vertical de sélection de l'intensité de cette couleur, plus à droite les numéros des couleurs avec une flèche indiquant la couleur sélectionnée, et les nombres représentant ces couleurs parmi les 4096 nuances possibles, ensuite un pictogramme représentant un micro-ordinateur pour revenir à la page en-tête.

Guide d'installation

Si vous possédez un lecteur de disquettes ou un QDD, deux pictogrammes supplémentaires sont présents pour vous permettre de sauvegarder et de relire votre palette personnalisée.



Pour sélectionner une couleur, il suffit de déplacer la flèche désignant les couleurs en tapant le nombre de fois nécessaire sur la touche jaune ENT. Vous remarquerez que la flèche saute de la couleur 15 à la couleur 0.

Simultanément, vous pouvez observer que la couleur de contour de la palette prend la teinte de la couleur sélectionnée et qu'un point apparaît à divers emplacements sur le triangle de sélection de la couleur et sur le rectangle d'intensité.

Pour modifier une couleur,

- commencez par la sélectionner avec la touche **ENT**,
- puis déplacez le point correspondant dans le triangle de sélection à l'aide des quatre touches fléchées du clavier.

Les trois sommets du triangle correspondent aux trois couleurs fondamentales Rouge, Vert, Bleu, qui, par addition, donnent toutes les nuances de couleurs représentées chacune par un point du triangle.

Pour modifier l'intensité, vous utilisez les touches marquées

et .

- la touche intensifie la teinte,
- la touche assombrit la teinte jusqu'au noir.

Avec une souris ou un crayon optique, le réglage est encore plus simple :

- sélectionnez la couleur à modifier en pointant la souris ou le crayon optique sur son numéro,
- déplacez le point représentant la couleur dans le triangle en pointant la souris ou le crayon optique,
- choisissez l'intensité en déplaçant de la même façon le point représentatif dans le rectangle de réglage.

Si vous possédez un lecteur-enregistreur de disquettes, il vous est possible de conserver votre palette personnelle en l'enregistrant.

Pour cela, il suffit d'appuyer au clavier sur la touche **S** pour la sauvegarder et sur la touche **L** pour lire la palette enregistrée sur la disquette, ou avec un crayon optique ou une souris, de pointer les pictogrammes correspondants.

Le fichier contenant votre palette s'appelle PALETTE.CFG. Pour sauvegarder votre palette sur cassette, voir page 11.

Pour retourner à la page en-tête, appuyez sur la touche STOP, ou pointez le pictogramme .

Guide d'installation

Le lecteur de cassettes

Le lecteur incorporé au MO6 fonctionne avec une vitesse de transfert des informations double de celle du MO5 : soit 2400 Bauds (Un Baud = un Bit par seconde).

Le deuxième choix du menu : "mettre la cassette à 1200 Bauds", vous permet d'enregistrer des cassettes à une vitesse de transfert compatible avec un lecteur de MO5.

A la lecture, le lecteur reconnaît automatiquement à quelle vitesse la cassette a été enregistrée, donc vous n'avez pas besoin de modifier ce paramètre.

Le clavier du lecteur de cassettes se compose de six touches :

de gauche à droite :

- la touche enregistrement
- la touche lecture
- la touche défilement arrière rapide
- la touche défilement avant rapide
- la touche d'arrêt de défilement de la bande et d'éjection de la cassette
- la touche d'arrêt momentané

Pour mettre en place une cassette, il faut appuyer sur la touche "STOP-EJECT" pour ouvrir le volet porte-cassette. Ensuite, il suffit d'introduire la cassette avec la bande magnétique orientée vers vous, et de refermer le volet.

Il est préférable d'utiliser des cassettes neuves ou préalablement effacées.

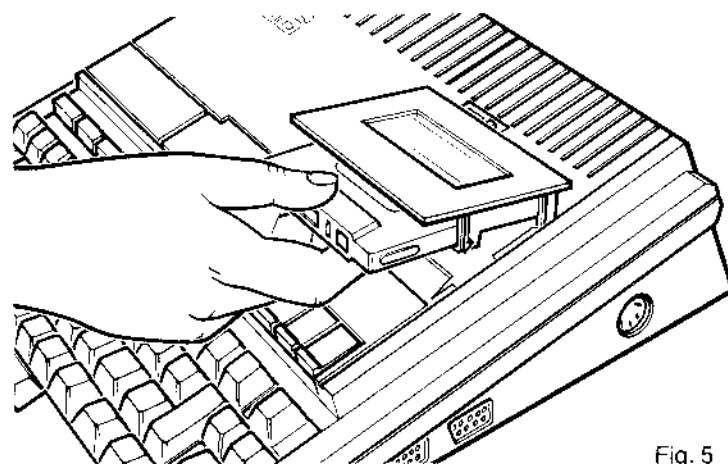


Fig. 5

Enregistrement

La cassette est utilisable des deux côtés.

Pour enregistrer au début d'une cassette, vous devez au préalable mettre le compteur à zéro en appuyant sur le bouton situé à sa droite, et faire défiler la bande un court instant en appuyant sur la touche de défilement avant rapide puis la touche STOP, de façon à éviter de commencer l'enregistrement sur la bande amorce.

Ensuite, appuyez simultanément sur la touche enregistrement et la touche lecture. Le voyant vert de mise en service passe au rouge. Le départ de la bande est commandé par le MO6.

A la fin de l'enregistrement, appuyez sur la touche "STOP-EJECT" pour déverrouiller les touches de lecture et d'enregistrement, et notez l'indication du compteur pour vous permettre d'enregistrer un autre programme à la suite.

Guide d'installation

Il est préférable de laisser un espace sur la bande entre deux enregistrements de façon à faciliter le repérage.

Vous pouvez protéger des enregistrements importants contre un effacement accidentel en supprimant la languette correspondant à la face à protéger.

Pour pouvoir enregistrer à nouveau sur une face protégée, il faut obturer l'ouverture de protection avec un ruban adhésif.

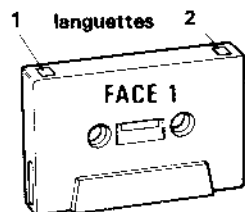


Fig. 6 Cassettes à bande à oxyde de fer

Lecture

La recherche d'un enregistrement est accélérée en utilisant le repérage du compteur réalisé lors de l'enregistrement.

Pour cela, vous pouvez faire défiler la bande à l'aide des touches de défilement avant ou arrière pour la positionner un peu avant le nombre repéré lors de l'enregistrement, appuyer sur la touche "STOP-EJECT" pour arrêter le défilement, puis appuyer sur la touche lecture. Le départ est commandé par le MO6.

Entretien

Pour maintenir toutes les qualités du lecteur de cassette, vous devrez de temps à autre procéder au nettoyage des têtes, du cabestan et du galet presseur à l'aide d'un coton-tige imbibé d'alcool à brûler.

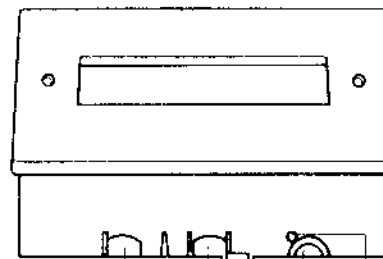


Fig. 7

Tête d'effacement
Tête de lecture et d'enregistrement
Cabestan
Galet presseur

Pour cela, enlevez la cassette, puis appuyez sur la touche lecture.

A la fin de l'opération de nettoyage, appuyez sur la touche "STOP-EJECT".

Guide d'installation

Pour utiliser une cartouche

Pour utiliser une cartouche de langage ou de programme dans votre MO6, insérez-la dans la trappe située au-dessus du clavier, l'étiquette de votre côté, et suivez les instructions accompagnant la cartouche.

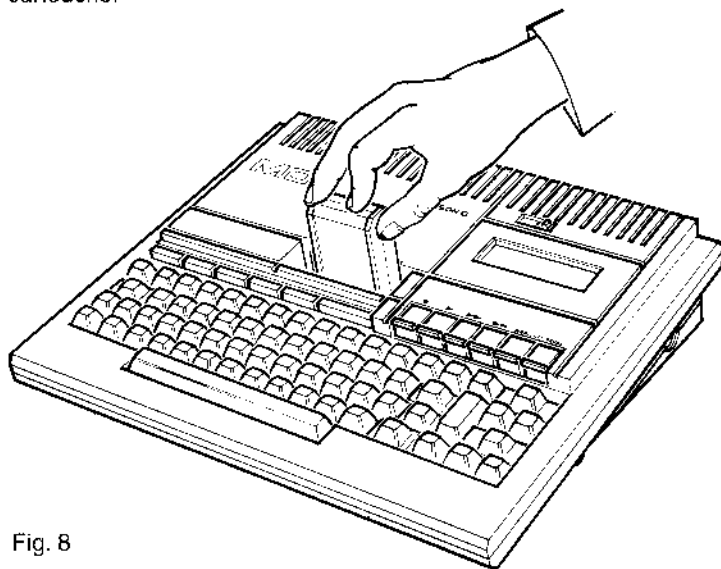


Fig. 8

Les périphériques du MO6

Sur votre MO6, vous pouvez raccorder les périphériques suivants :

- un crayon optique
- une souris
- une ou deux manettes de jeu
- une imprimante : 40 colonnes, modèle PR 90 055
80 colonnes, modèles PR 90 582 ou PR 90 600

Sur le connecteur d'extension, vous pouvez raccorder un des périphériques suivants :

- un QDD avec son contrôleur
- un lecteur-enregistreur de disquettes 3,5 pouces avec son contrôleur
- un lecteur-enregistreur de disquettes 5,25 pouces, modèle 320 Ko double densité ou modèle 80 Ko simple densité
- un MODEM modèle MD 90 333
- une interface IEEE
- une interface série RS 232 (uniquement le modèle RS 57 932)
- une interface d'incrustation vidéo

En revanche, vous ne pouvez pas connecter les périphériques suivants :

- la cartouche RAM d'extension mémoire 64 Ko pour utilisation en réseau (déjà intégrée)
- l'extension musique et jeux (déjà intégrée)
- l'extension sortie parallèle (déjà intégrée)
- l'extension digitalisation
- l'extension télématique

NOTA 1 : Lors de la sélection d'un des langages BASIC du MO6 avec un lecteur de disquettes 5,25 pouces double densité connecté, le contrôleur détecte automatiquement la densité de la disquette présente, et continuera à travailler dans cette densité. Si, par erreur, vous avez laissé une disquette enregistrée dans la densité que vous ne souhaitez pas utiliser, il faut mettre une disquette de la bonne densité dans le lecteur et appuyer sur la touche RESET.

NOTA 2 : Lors de l'utilisation du MO6 avec à la fois une souris active et une imprimante, l'appui sur une des touches de la souris stoppe l'impression d'un document. Cette impression reprend en relâchant la touche.

Caractéristiques principales

I — Particularités électriques

Alimentation secteur : 220 V, 50 Hz
Consommation : 24 W
Fusible : T 160 mA

II — Présentation

Coffret : matière moulée
Dimensions : L = 362, H = 87, p = 315
Masse : 3 Kg
Clavier : AZERTY, 69 touches mécaniques, majuscules et minuscules accentuées avec voyant de mode majuscule.

III — Particularités techniques

Microprocesseur : 6809 E - 1 MHz
Mémoire : morte : 64 Ko extensible comportant les logiciels intégrés :
 BASIC 128, BASIC 1.0 MICROSOFT
 vive : 128 Ko non extensible
Ecran : sortie RVB + son par prise péritélévision
 320 × 200 - 16 couleurs (40 colonnes)
 640 × 200 - 2 couleurs (80 colonnes)
 320 × 200 - 4 couleurs point par point (40 colonnes)
 160 × 200 - 16 couleurs point par point
 320 × 200 - 3 couleurs avec un niveau de transparence
 320 × 200 - 2 couleurs avec affichage alternatif de 2 pages
 160 × 200 - 5 couleurs avec 3 niveaux de transparence

Toutes les couleurs sont sélectionnables dans une palette proposant 4096 nuances.

Crayon optique : résolution 320 x 200 points

Lecteur-enregistreur de programme intégré 1200 ou 2400 bauds à reconnaissance automatique en lecture

Synthétiseur musical intégré de 4 voix sur 7 octaves

Sortie son : prise CINCH

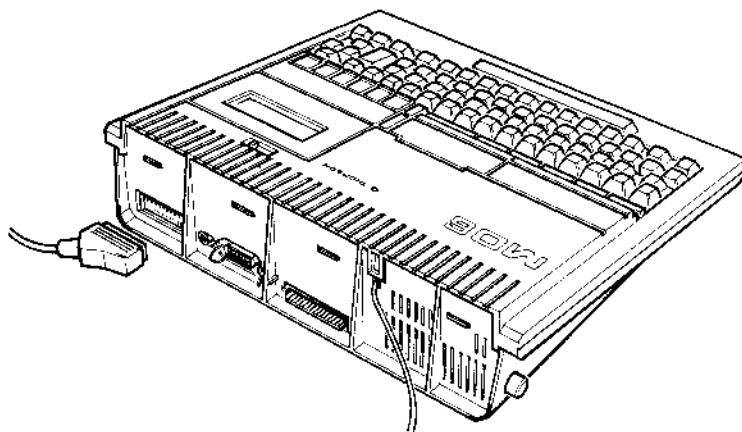
Connecteurs :

sur l'arrière : un connecteur d'extension polyvalent

 une prise MICRORIBBON 14 points, standard CENTRONICS pour imprimante

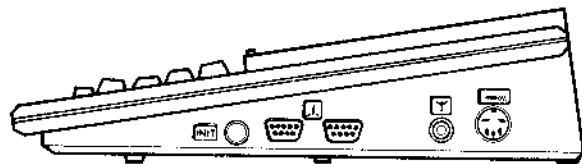
 une prise CINCH sortie son

 une sortie Péritélévision SCART

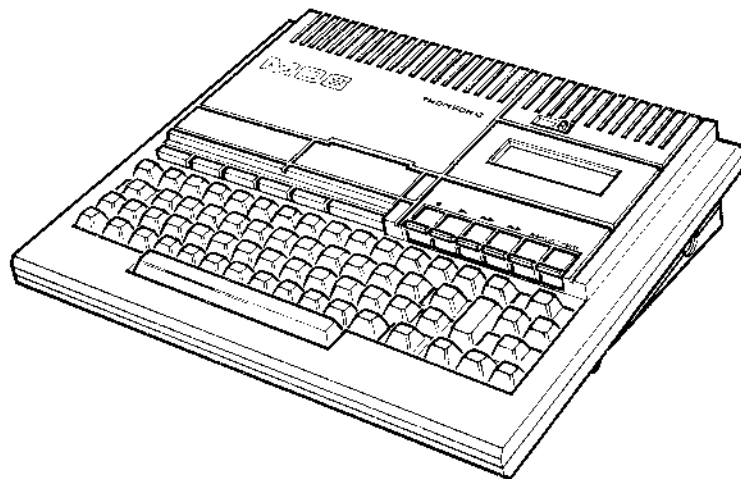


Guide d'installation

sur le côté droit : deux prises Sub-D 9 points pour souris, manettes de jeu
une prise CINCH sortie UHF-PAL (export)
une prise DIN pour crayon optique



sur le dessus : une trappe pour cartouche de programme ou de langage.



Guide du BASIC

Après cette première prise de contact, il est temps d'utiliser toutes les possibilités du BASIC MO6. En fait, votre MO6 possède deux langages BASIC :

- par la touche 1, vous accédez au BASIC 128 MICROSOFT®;
- par la touche 2, vous accédez au BASIC MICROSOFT® 1.0 ;

Le BASIC 1.0 est intégré pour vous permettre d'utiliser la bibliothèque de logiciels du MO5.

Le guide du BASIC vous permet de découvrir la puissance et la richesse du BASIC 128. Il se compose de deux parties :

- une initiation à la programmation proposée en 50 modules simples,
- un guide de référence précisant toutes les particularités de chacun des deux langages disponibles.

Apprendre la programmation peut prendre du temps, mais obtenir immédiatement des résultats encourageants sur l'écran ne demande que quelques minutes. Du module 1 au module 17, vous introduisez toutes les instructions qui vous permettent de vous initier

au fonctionnement du BASIC. Les résultats graphiques et musicaux sont prometteurs.

Pour guider le programmeur novice, les modules 21 et 22 donnent les principes généraux de la programmation. Après, la route est libre ! Chaque module suivant enrichit le vocabulaire connu et résout un petit problème : comment réaliser une saisie au clavier, comment dessiner ses propres motifs, etc.

Pour finir, les modules 49 et 50 réservent aux curieux bien des satisfactions. En effet, en quelques lignes de programme, il permettent de créer des effets vidéo tout à fait saisissants (49) et fournissent un réservoir inépuisable de Rock and Roll (50).

Pour l'instant, sélectionnez le BASIC 128 avec la touche 1 et... suivez le guide !

La touche ENTRÉE et les autres

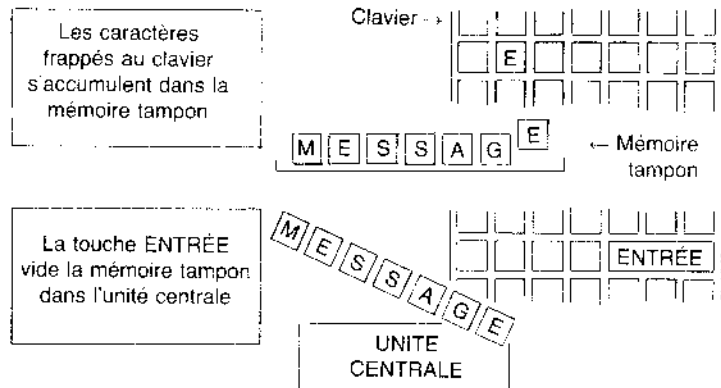
Avant toute chose, il est important de planter le décor. Pour l'utilisateur du MO6, le décor se compose du clavier et de l'écran. Il "commande" son MO6 à partir du clavier. Les réponses de la machine — quand elle en donne — sont visibles sur l'écran. La touche importante permettant de réaliser ce dialogue est la touche ENTRÉE.

Elle se remarque au premier coup d'œil et est très accessible. Vous allez comprendre pourquoi dès que vous aurez un peu utilisé le clavier: elle sert tout le temps.

La touche ENTRÉE

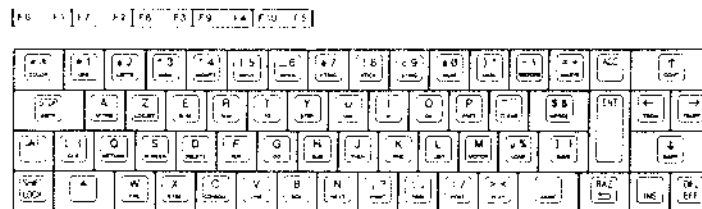
On dit aussi que c'est la touche de Retour chariot puisque sa fonction est semblable à celle d'un retour de chariot sur une machine à écrire classique. Avec cependant des différences importantes qui lui donnent de fait un rôle prépondérant.

Lorsque vous tapez sur le clavier un message à destination de l'ordinateur, les caractères sont empilés dans une mémoire intermédiaire, dite tampon, qui enregistre passivement votre message: lorsque vous écrivez une lettre, les mots s'accumulent sur le papier sans que votre correspondant en ait connaissance. Le geste important consiste à glisser votre message dans la boîte aux lettres. Avec un micro-ordinateur, l'envoi du message se fait simplement en appuyant sur la touche ENTRÉE: la suite des caractères frappés se vide alors dans la partie active de la machine qui analysera et exécutera éventuellement les instructions qu'elle contient.











La capacité de la mémoire tampon est de 255 caractères. Cela signifie que le dialogue avec la machine est limité à 255 caractères pour un seul envoi. Cela permet quand même de se comprendre.


Un coup d'œil au clavier



Ce clavier est du type AZERTY, c'est-à-dire qu'il est utilisé dans les pays de langue française. (AZERTY fait référence aux premières lettres que l'on peut lire sur les touches du clavier.)

Lorsque vous programmez en BASIC, prenez l'habitude d'enfoncer au préalable la touche SHIFT-LOCK (voir page 5). Le voyant s'allume. Prenons l'exemple de la touche  qui permet de produire trois écritures différentes.

appuyer sur 	produit l'écriture de 
appuyer sur  	produit l'écriture de 
appuyer sur BASIC 	produit l'écriture de 

Les touches autres qu'alphanumériques ne sont pas affectées par la touche SHIFT-LOCK. Pour celles-ci, vous obtiendrez le signe inscrit en vert en appuyant d'abord sur la touche  (voir page 5). Le mot BASIC s'obtient comme précédemment.

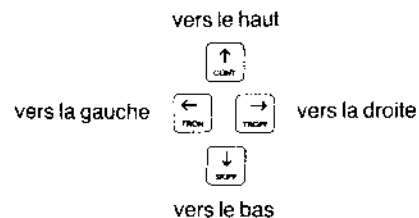
L'utilisation de ces touches "préprogrammées" permet d'éviter des erreurs de frappe et accélère l'écriture.

Le propriétaire du MO6 a donc une photographie instantanée des mots de BASIC qui sont déjà écrits sur son clavier. Ce sont les plus courants. Il est important de bien connaître son clavier pour ne pas chercher trop longtemps les mots qui s'y trouvent (ou qui ne s'y trouvent pas...).


Les touches de déplacement du curseur

Le curseur est un caractère spécial toujours présent à l'écran et (en général) clignotant. Il indique à l'utilisateur l'endroit précis où s'affichera le prochain caractère tapé.

Quatre touches fléchées permettent de déplacer le curseur et leur disposition rappelle leur action:



Il est important de bien se familiariser avec leur emploi.

Notons également, à l'autre extrémité du clavier, une touche qui agit sur le curseur: elle le replace tout en haut et à gauche. Elle est symbolisée par .

Remarque: il est important de bien différencier le zéro "0", et la lettre O.

Nous vous renvoyons à la partie installation pour l'accès aux majuscules/minuscules ainsi qu'aux accents et cédille.

La frappe de minuscules avec accent circonflexe ou tréma s'opère de la façon suivante :

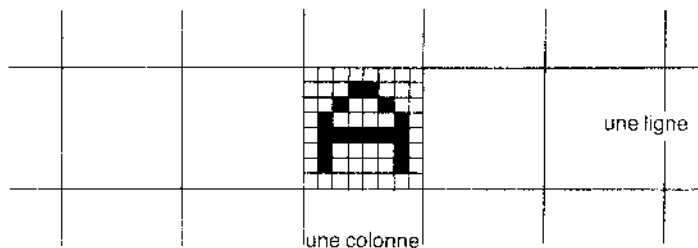
- le voyant de la touche SHIFT-LOCK doit être éteint,
- on frappe la touche ACC,
- on frappe l'accent circonflexe ou le tréma.
- on frappe la minuscule à accentuer.

Les autres minuscules accentuées sont accessibles directement.

L'écran et le clavier

Il faut voir l'écran du téléviseur comme une grille : des colonnes verticales, des lignes horizontales dont les intersections définissent des cases où viennent s'inscrire les caractères.

Chaque case elle-même peut être vue comme un autre quadrillage plus fin dans lequel certaines cases (les pixels) sont allumées ou éteintes afin de faire apparaître un caractère.



L'appui sur l'une des touches de déplacement du curseur permet donc de passer d'une case dans l'une de ses quatre voisines. Le nombre de lignes et le nombre de colonnes ne sont pas une caractéristique de l'écran mais du micro-ordinateur lui-même.

Le MO6 affiche sur 40 colonnes et 25 lignes.

Nettoyer l'écran

La touche RAZ efface complètement l'écran : RAZ signifie Remise à zéro. Le curseur se retrouve ainsi en haut et à gauche et le MO6 est prêt à recevoir de nouveaux ordres.

L'écran est propre : on peut commencer.

Affichage

Puisque l'écran est l'outil qui permet à l'ordinateur de vous communiquer les résultats de ses travaux, il est bien normal que vous commenciez par un ordre d'affichage. Mieux vaut commencer par les nombres puisque ce sont probablement eux que la machine manipule le plus facilement.

L'ordre d'affichage est PRINT. Dans toute cette partie, ce que vous devrez taper au clavier sera inscrit en couleur, les réponses de l'ordinateur seront inscrites en noir.

```
PRINT 1789 [ENT]
1789
OK
```

Ce premier essai n'est pas très spectaculaire, il faut en convenir. En fait, le spectacle était caché. On peut soulever légèrement le voile. La mémoire tampon a d'abord enregistré les dix caractères frappés, ceux qui ont précédé la touche ENTRÉE.

```

                blanc
P  R  I  N  T  [ ]  1  7  8  9
barre d'espace
```

L'appui sur la touche ENTRÉE a vidé la mémoire tampon dans l'interpréteur BASIC qui a aussitôt attaqué son travail d'analyse. Au cinquième caractère, il a identifié un mot clef du langage BASIC. Il a lancé la procédure attachée à ce mot : ignorer le blanc puis afficher à l'écran tout ce qui précède la touche ENTRÉE.

Passons immédiatement à un usage plus spectaculaire de l'ordre PRINT.

```
PRINT 17+35 [ENT]
52
OK
```

Voilà qui est mieux ! Notre MO6 a l'air de se comporter comme une vulgaire calculatrice, ce qui n'est déjà pas si mal. On peut tout de même se demander pourquoi il est nécessaire de commander l'affichage alors que ce n'est pas indispensable avec une calculatrice. Vous le verrez plus tard mais vous pouvez le comprendre tout de suite. Si je commande un calcul assez long, il est probable que seul le résultat final m'intéresse ; l'affichage des résultats intermédiaires peut être sans intérêt, voire désagréable. Il faut donc penser à l'ordinateur de la façon suivante : je peux lui demander tous les

calculs que je veux mais si je veux voir les résultats, je dois lui demander de les afficher. Grande différence avec une calculatrice. Gros avantage en fait, car l'utilisateur maîtrise complètement ce qu'il va voir.

En fait, vous vous en doutez sûrement, même en ce qui concerne les calculs, l'ordinateur fait beaucoup mieux qu'une calculatrice...

Voici par exemple l'exécution d'un calcul compliqué dans lequel on utilise:

- Le point . pour séparer dans les nombres décimaux la partie entière de la partie décimale conformément à l'usage anglo-saxon
- L'addition + et la soustraction -
- La multiplication * (pour ne pas confondre avec la lettre X)
- La division / qu'on appelle aussi "slash"
- Les parenthèses (et) pour lesquelles il faut s'assurer qu'elles sont ouvertes et fermées dans le bon ordre.

Il est important de savoir également que les règles de priorité des quatre opérations suivent celles couramment utilisées en mathématiques: la multiplication et la division à égalité sont prioritaires sur l'addition et la soustraction à égalité. Les parenthèses imposent des priorités absolues.

```
PRINT (5-3.4-2*5.6)/7+2 [ENTER]
2.82857
OK
```

Les avantages les plus importants par rapport à une calculatrice sont:

- La permanence du calcul et de son résultat
- La possibilité grâce à l'éditeur de ligne de modifier et de corriger.

L'éditeur

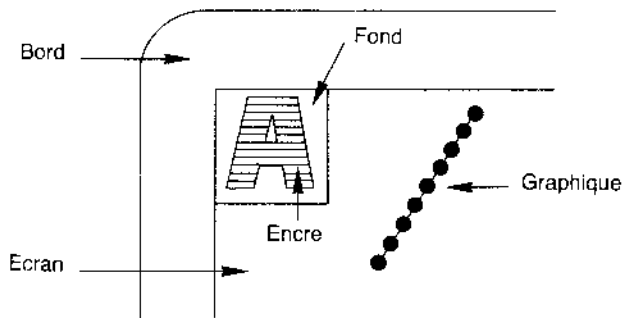
L'éditeur est un programme interne destiné à faciliter l'affichage et la correction. Il est également très important de bien maîtriser l'éditeur. Quand une faute a été repérée, les touches de déplacement du curseur vous permettent de vous positionner sur la zone suspecte pour y effectuer la correction; cette correction peut être de trois types:

- Le remplacement d'un caractère: il suffit de frapper le nouveau caractère à la place de l'ancien sans autre précaution; le résultat est immédiat à l'écran.
- La suppression: pour supprimer un caractère, il ne faut surtout pas frapper la barre d'espacement car il est nécessaire que dans le même temps, le caractère disparaisse de l'écran et que le reste de la ligne se recolle. Vous disposez de la touche EFF (comme effacement) qui réalise d'un coup ces deux opérations.
- L'insertion: pour ajouter un ou plusieurs caractères, il faut appuyer sur la touche INS (comme insertion) le caractère devant lequel va se faire l'insertion s'affiche en vidéo inverse. Les nouveaux caractères que vous ajoutez s'écrivent en repoussant le reste de la ligne: vous êtes en mode insertion. Pour sortir du mode insertion, il suffit d'appuyer à nouveau sur la touche INS: le caractère devant lequel l'insertion s'est réalisée revient aux couleurs standard.

D'autre part, les quatre flèches de déplacement du curseur vous permettent de vous déplacer partout sur l'écran et d'y inscrire n'importe quoi, n'importe où. On dit que l'éditeur du MO6 est un éditeur "plein écran".

Les couleurs de l'écran

Lors d'une insertion, nous avons remarqué que le MO6 utilisait la vidéo inverse. C'est l'une des possibilités offertes par le MO6 pour gérer les couleurs. De façon générale, l'utilisateur peut faire intervenir des codes couleur à cinq niveaux :



Votre MO6 possède seize couleurs sélectionnables parmi 4 096 nuances. Au départ, l'affichage se fait en bleu foncé sur cyan (bleu pâle). Nous allons changer les caractéristiques de l'affichage. L'ordre de modification des couleurs de l'écran doit être donné à la machine sous la forme :

SCREEN 7.0.4

Tapez cette instruction puis demandez à l'ordinateur de "l'exécuter" avec la touche ENTRÉE. Nous comptons sur vous pour bien avoir compris le rôle de la touche ENTRÉE et nous ne la ferons plus figurer en fin de commande.

Si votre message est bien transmis, et bien reçu..., l'instruction s'exécute et les caractères deviennent blancs sur fond noir, avec un pourtour bleu.

Ensuite l'ordinateur répond OK et le curseur revient au début de la ligne suivante, en attente d'une nouvelle instruction.

L'instruction SCREEN (= écran) est précisée par trois nombres :

- le premier pour la couleur des caractères
- le deuxième pour la couleur du fond
- le troisième pour la couleur du cadre qui entoure le rectangle utile.

Un peu de technique sur les couleurs

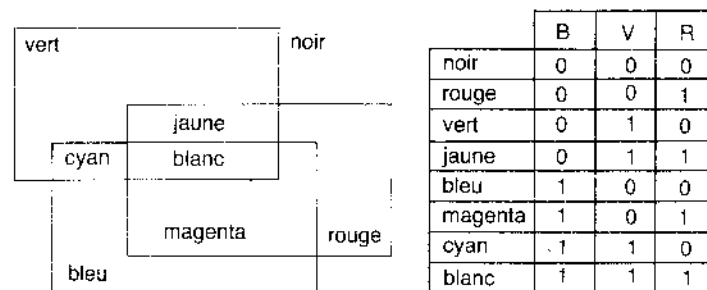
L'écran d'un téléviseur couleur est constellé de points portant chacun trois couleurs: rouge, vert et bleu (RVB).

Sur l'écran, contrairement à ce qui se passe sur le papier, l'œil fait une addition des couleurs. A partir des trois couleurs fondamentales, on obtient cinq autres couleurs de synthèse, soit au total huit couleurs (2^3) de base.

Les couleurs sont codées dans la machine à l'aide de "bits". Le bit est la plus petite unité de stockage d'information utilisée par votre micro-ordinateur. Il vaut 1 ou 0 et ne peut prendre que ces deux valeurs.

Exemple pour le codage des huit couleurs de base

Chacune de ces huit couleurs peut être codée sur trois bits: chaque bit est à 0 ou à 1 suivant que l'une des trois couleurs fondamentales est présente ou pas. Le schéma et le tableau suivant résument la synthèse et le codage de ces huit couleurs.



Pour augmenter le nombre de nuances, on joue sur la luminosité de chacune des trois couleurs fondamentales.

On code ainsi un quatrième bit pour avoir, soit la couleur foncée, soit la couleur pâle.

Note : une exception est faite pour le "blanc pâle" qui est orange!!!

Tableau des couleurs

0 noir	6 cyan (bleu pâle)	12 bleu clair
1 rouge	7 blanc	13 magenta clair
2 vert	8 gris	14 cyan clair
3 jaune	9 rouge clair	15 orange
4 bleu	10 vert clair	
5 violet (magenta)	11 jaune clair	

Cette numérotation est "standard". L'option 1 du menu principal (choisir sa palette de couleurs) vous permet de changer les couleurs et leurs codes. Dans tout le manuel, nous utiliserons la palette standard.

La syntaxe de l'instruction SCREEN

Une instruction peut souvent s'écrire sous des formes plus ou moins simplifiées.

Ainsi dans l'instruction SCREEN, vous n'êtes pas obligé de faire suivre le mot clé par trois nombres.

Exemples : pour changer seulement la couleur des caractères en rouge, il suffit de taper :

```
SCREEN 1
```

et pour changer la couleur des caractères et du fond, on fait :

```
SCREEN 1,5
```

Il existe même un quatrième paramètre (ou "argument") utilisable :

```
SCREEN ...,1
```

Cette commande inverse les couleurs des caractères et du fond ("vidéo inverse"). Si vous exécutez une deuxième fois :

```
SCREEN ...,1
```

il se produit une nouvelle inversion qui ramène l'affichage aux couleurs initiales.

Ces règles constituent la syntaxe de l'instruction SCREEN. Par la suite, nous ne donnerons que la forme la plus courante des instructions, ou celle dont nous aurons besoin, et c'est dans la partie Référence que vous trouverez toutes les possibilités de chaque instruction.

Quand le MO6 refuse d'avancer!

A force de taper des PRINT ou des SCREEN, vous allez vous retrouver avec une réponse du type: "Syntax Error". Cette réponse signifie que le MO6 — ou plus particulièrement le programme interne chargé de comprendre ce que vous entrez au clavier — n'est pas capable de reconnaître un mot attendu. Ce programme interne s'appelle l'interpréteur BASIC.

Il est responsable de la traduction des instructions que vous tapez de façon que le micro-ordinateur puisse vous comprendre. En effet, le MO6 et vous-même ne parlez pas le même langage. Vous faites un effort en parlant "BASIC" ; l'interpréteur se charge du reste. Quand il fait afficher "Syntax Error", c'est souvent qu'il y a eu une erreur de frappe (par exemple PRIT à la place de PRINT) et qu'il ne comprend plus la syntaxe de votre instruction. L'utilisation de l'éditeur pleine page vous permettra de rectifier l'erreur repérée. Une seconde erreur peut intervenir avec l'instruction SCREEN. Si vous donnez un nombre supérieur à 15 pour une couleur ou si vous oubliez la virgule entre deux paramètres (paramètres de la couleur des caractères et de celle du fond par exemple) vous obtiendrez le message suivant:

```
Illegal Function Call
```

Cela signifie que l'interpréteur a bien compris que vous voulez changer les couleurs de l'affichage (donc l'instruction SCREEN) mais que les paramètres que vous proposez ne correspondent pas à des couleurs autorisées. En même temps que vous travaillez votre BASIC, sans vous en rendre compte, vous progressez en anglais.

Note : La liste des messages d'erreur se trouve en annexe.

Afficher n'importe quoi, n'importe où...

... mais pas n'importe comment !

Essayons un peu :

```
PRINT BABA
0
OK
```

Drôle de surprise ! Au lieu d'imprimer le BABA attendu, c'est 0 qui nous est servi. Vous vous convaincrez facilement que PRINT BIBI donnerait exactement le même résultat et que finalement, le mot ne fait rien à l'affaire.

L'explication de cette curiosité se trouve au module 6. Les impatients s'y précipiteront. Pour les autres, voici la bonne façon d'afficher le mot BABA : il suffit de le placer entre deux guillemets.

```
PRINT "BABA"
BABA
OK
```

Remarque : C'est le même guillemet qui ouvre et qui ferme. Si le mot PRINT est suivi d'un guillemet, l'ordinateur affiche tout ce qui est entré au clavier entre les deux guillemets.

Il faut observer qu'à ce compte-là, il est impossible d'afficher des guillemets. Mais impossible ne fait pas partie du vocabulaire BASIC : vous trouverez une solution à ce délicat problème à la fin du module 10.

Entre les guillemets, l'utilisateur est libre de placer les caractères qu'il désire, y compris des blancs, des nombres : l'ensemble de ces caractères s'appelle une chaîne de caractères.

La longueur des chaînes de caractères est limitée à 255 caractères, plus de six lignes sur un écran de quarante colonnes. Pour afficher de longs textes, il est donc nécessaire de faire intervenir plusieurs chaînes. Si la chaîne de caractères dépasse quarante caractères, elle sera affichée sur plusieurs lignes. Rappelez-vous bien à cette occasion qu'il ne faut pas confondre une ligne d'écran avec une ligne d'instruction. La fin d'une ligne d'instruction est marquée par la touche ENTRÉE et n'est pas visible concrètement à l'écran.

Si une chaîne occupe plus d'une ligne, elle sera découpée en fonction de la dimension de l'écran sans tenir compte des mots.

En voici un exemple :

```
PRINT " NOUS SOMMES DEUX DROLES AJX LARG
ES EPAL LES"
NOUS SOMMES DEUX DROLES AUX LARGES EPAU
LES
OK
```

Subtilités : Ne pas confondre une chaîne vide (deux guillemets successifs) avec une chaîne contenant un ou plusieurs blancs obtenus avec la barre d'espace. Pour bien distinguer sur les listings, il faut simplement tenir compte du fait que tous les caractères à l'écran ou sur imprimante occupent la même longueur. Les instructions :

```
PRINT "
PRINT" "
PRINT" "
```

aboutissent au même résultat (rien), mais l'ordinateur, lui, ne s'y trompe pas. Il n'acceptera pas, par exemple, une chaîne qui contiendrait plus de 255 blancs.

Méditez quelques instants sur ces deux expériences.

```
PRINT "12+3"
12+3
OK
```

```
PRINT 12 + 3
15
OK
```

On gagne du temps en remplaçant le mot PRINT par un simple point d'interrogation ? . L'interpréteur BASIC comprend bien votre souci d'économiser vos forces et traduira de lui-même le ? en PRINT.

Vous pouvez aussi, et cela revient au même, presser ensemble la touche BASIC et la touche ? et afficher le mot PRINT d'un seul coup de doigts.

Pour l'instant, nous n'avons proposé d'écrire qu'une seule instruction. Votre MO6 ne fait pas plusieurs choses à la fois mais permet d'enchaîner les travaux si rapidement que c'est tout comme.

Enchaîner les travaux grâce aux séparateurs

Les deux-points

Plusieurs instructions peuvent se suivre dans une même ligne d'instruction à condition de les séparer par deux points. Puisque nous ne disposons pour l'instant que de la seule instruction PRINT, nous ne pouvons enchaîner que des ordres d'affichage.

```
PRINT 1914 : PRINT 1918
```

```
1914  
1918
```

```
OK
```

On observe que le deuxième ordre PRINT fait automatiquement passer à la ligne. C'est la même chose avec les chaînes de caractères.

```
PRINT "ALI" : PRINT "BABA"
```

```
ALI  
BABA  
OK
```

L'ordre PRINT seul permet de sauter une ligne.

```
PRINT "FOCH" : PRINT : PRINT "CLEMENCEAU"
```

```
FOCH
```

```
CLEMENCEAU
```

```
OK
```

Le point-virgule

Pour réaliser plusieurs affichages sur la même ligne d'écran, plus précisément, à la suite les uns des autres, on fait suivre le nombre, ou la chaîne de caractères à afficher d'un point-virgule.

```
PRINT 1914 ; 1918
```

```
1914 1918
```

```
OK
```

Le blanc qui sépare les deux nombres est réservé au signe, ici le signe + qui n'est pas imprimé.

L'exemple précédent pourrait être réalisé d'une façon différente.

```
PRINT 1914 : : PRINT 1918
```

```
1914 1918
```

```
OK
```

Si donc, l'utilisateur décide d'écrire deux instructions d'affichage distinctes, il doit obligatoirement les séparer par deux points. Toute omission ou toute inversion provoquera inmanquablement un message d'erreur.

Remarque : On a vu que l'écran est composé de 25 lignes de 40 colonnes. Elles sont repérées par un nombre de 0 à 24 de haut en bas pour les lignes, et de 0 à 39 de gauche à droite pour les colonnes. Il est possible de demander l'affichage à un endroit particulier de l'écran en faisant précéder l'ordre PRINT d'un ordre LOCATE.

Exemple :

```
LOCATE 10,15 : PRINT "BONJOUR"
```

affiche BONJOUR au milieu de l'écran.

Le B de Bonjour est en colonne 10 et ligne 15.

La virgule

La virgule, dans les ordres d'impression, s'utilise de la même façon que le point-virgule mais elle réalise des affichages sur la même ligne, dans des zones de treize caractères. S'il s'agit de nombres, le premier est affiché à partir de la première colonne, le second à partir de la quatorzième colonne, le troisième à partir de la vingt-septième, etc. Ce signe est utilisé pour réaliser des pages écran bien disposées même lorsque la taille des nombres et des chaînes de caractères est variable.

```
PRINT 1,2,3
1    2    3
OK
PRINT 10,100,1000
10   100  1000
OK
PRINT "ALI","BABA","VOLEUR"
ALI  BABA  VOLEUR
OK
```

Faisons le point sur la ponctuation

- Sépare deux instructions dans une même ligne d'instructions.
- Placé après un nombre ou une chaîne de caractères à imprimer, le point-virgule force le prochain affichage sur la même ligne à la suite du précédent.
- Placée entre deux nombres ou chaînes de caractères, la virgule provoque un affichage en ligne toutes les quatorze colonnes à partir de la première.
- ? Abréviation de PRINT.
- Tient lieu de virgule pour les nombres décimaux.

Ecrire gros ou petit

Quatre tailles sont disponibles pour l'affichage des caractères. Elles sont décidées grâce à l'instruction ATTRB.

	Hauteur	Largeur	Exemple
ATTRB 0,0	normale	normale	
ATTRB 1,0	normale	double	
ATTRB 0,1	double	normale	
ATTRB 1,1	double	double	

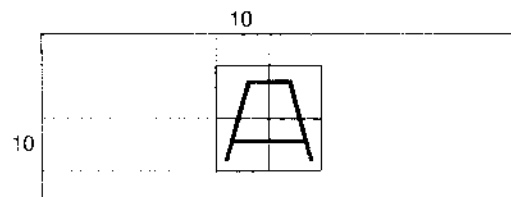
L'absence d'instruction sous-entend ATTRB 0,0.

Lorsqu'une instruction ATTRB est donnée, elle reste valable jusqu'à nouvel ordre.

L'éditeur BASIC revient toujours au mode ATTRB 0,0.

Remarque importante

Lors d'une impression en double dimension, l'instruction LOCATE fixe la position du caractère en bas à gauche.



```
ATTRB 1,1 : LOCATE 10,10 : PRINT "A"
```

On s'apercevra donc que l'impression en double hauteur d'un caractère sur la ligne 0 ne laisse apparaître que la partie inférieure du caractère.

Changer la couleur de l'affichage: COLOR

Pour mettre un ou plusieurs mots en relief sur une page à l'écran, il est possible de changer la couleur des caractères de ces mots. Par exemple, avec l'instruction COLOR 1,4, l'affichage s'effectue en rouge (1) sur fond bleu (4).

L'instruction COLOR a une syntaxe proche de celle de SCREEN puisque le premier chiffre désigne la couleur des caractères et le deuxième celle du fond. Mais elle n'agit que sur les affichages ultérieurs alors que SCREEN agit sur tout l'écran.

Dans vos multiples essais, vous avez pu constater les possibilités offertes par le micro-ordinateur. Dans les deux remarques suivantes, nous indiquons quelques particularités qui peuvent être utiles:

Remarque importante

Une instruction comme:

```
LOCATE 35,10 : PRINT "ALI-BABA"
```

pose un problème que le MO6 résout à sa façon. En démarrant l'impression de la chaîne ALI-BABA (huit caractères) en colonne 35, il est impossible de ne pas couper la chaîne, ce qui peut être désagréable pour tout le monde. Le MO6 prend donc le parti de ne pas écrire la chaîne à l'endroit demandé, mais au début de la ligne suivante. Avant chaque exécution d'une instruction PRINT, le MO6 évalue donc si l'impression est possible sans saut de ligne. Dans le cas contraire, il commence l'impression au début de la ligne suivante.

Ce comportement présente à la fois des avantages et des inconvénients. Si vous souhaitez forcer l'affichage à l'endroit que vous avez choisi, remplacez PRINT par PRINT USING "&"; et reportez-vous à la partie Références (page 180).

Pour les perfectionnistes

L'instruction LOCATE accepte un troisième argument qui décide de la présence éventuelle du curseur.

Dans certains écrans graphiques, lorsqu'une belle image est affichée, il peut être désagréable de voir le curseur clignoter n'importe où d'une façon disgracieuse.

```
LOCATE X,Y,0
```

permettra au curseur de ne pas apparaître,

```
LOCATE X,Y,1
```

le fera revenir.

Un problème surprenant:

```
PRINT BABA +5
```

```
5
```

```
OK
```

```
PRINT BABA -3
```

```
-3
```

```
OK
```

```
PRINT BABA/2
```

```
0
```

```
OK
```

Mais qui est baba? Avec un petit peu d'observation, vous devriez deviner ce qui vient de se produire sous vos yeux. Sinon, la réponse est contenue dans le module suivant... à condition que vous fassiez un peu d'Histoire.

Vous avez sans doute été surpris par l'inculture flagrante des micro-ordinateurs. Hormis les tables de multiplication et quelques babioles de ce genre, ils ne savent rien. Par exemple, la date de la bataille de Marignan.

```
PRINT MARIGNAN
0
OK
```

On constate donc que l'élève ne sait pas sa leçon puisque la bataille de Marignan ne se déroula pas en l'an 0 mais en l'an 1515. Il est donc nécessaire à l'utilisateur de jouer les professeurs pour remonter légèrement le niveau culturel de cette pauvre machine. Voici comment s'y prendre.

```
MARIGNAN= 1515
OK
PRINT MARIGNAN
1515
OK
```

Pour comprendre : l'instruction `MARIGNAN = 1515` est ce qu'on appelle une instruction *d'affectation*. Elle commande à la machine de ranger, dans une ou plusieurs cases mémoire, le nombre 1515. Elle indique aussi que l'utilisateur pourra faire appel au nombre rangé en l'appelant `MARIGNAN`.

C'est ce qui est fait dans la deuxième instruction. L'utilisateur demande d'imprimer le contenu des cases mémoire `MARIGNAN`. La machine va le chercher là où elle l'a mis et le restitue parfaitement: 1515. A partir de là, l'utilisateur pourra demander autant de fois qu'il voudra, le contenu de `MARIGNAN`, il lui sera servi avec une fidélité absolue.

Variables

Dans l'exemple précédent, `MARIGNAN` est appelée une variable. Ce nom est fort mal choisi puisque le contenu de la variable `MARIGNAN` n'a aucune raison de varier! Seule une nouvelle instruction d'affectation peut en être la cause.

Voici un exemple qui utilise deux variables `A` et `B`, et offre l'occasion de réutiliser les séparateurs (voir module 4).

```
A=25.7 : B=140 : PRINT A ; B
25.7 140
OK
```

Avant toute instruction d'affectation, le contenu des variables est initialisé à 0. Voici donc pourquoi, lorsqu'on l'allume, le MO6 est persuadé que la bataille de Marignan eut lieu en l'an 0!

Transfert

Le contenu d'une variable `A` peut être versé dans une variable `B`. On obtient ainsi une duplication.

```
A=75 : B=0 : PRINT A ; B
75 0
OK
B=A : PRINT A ; B
75 75
OK
```

Au départ, `A` vaut 75 et `B` vaut 0.

L'instruction `B = A` met `A` dans `B` (on peut penser à $B \leftarrow A$).

Il ne faut pas oublier que cette instruction écrase l'ancien contenu de `B` dont il ne reste plus aucune trace.

Permutation

Soient deux variables. L'une s'appelle X et contient le nombre 10, l'autre s'appelle Y et contient le nombre 25.

On voudrait permuter les contenus de ces variables, c'est-à-dire mettre le contenu de X dans Y et inversement. On y parvient aisément par :

```
X=25 : Y=10
```

Imaginons maintenant que vous ne connaissiez pas les contenus de X et de Y. Imaginons encore que l'on vous interdise d'en prendre connaissance en tapant PRINT X ou PRINT Y.

Cette petite énigme réclame un peu de réflexion.

Avant de donner une solution, nous ne pouvons résister au plaisir de dire deux mots sur une mauvaise piste dans laquelle on a immédiatement envie de se lancer.

Puisqu'il faut permuter X et Y, on peut se dire : je vais mettre X dans Y puis je mettrai Y dans X. C'est une assez bonne idée mais dès l'instant où vous aurez mis X dans Y, vous aurez perdu le contenu de Y. En mettant ensuite Y dans X, c'est en fait X que vous remettez dans X.

De cette mauvaise idée naît une bonne idée : avant de passer X dans Y, il faut transférer le contenu de Y dans une variable accessoire U.

D'où les instructions suivantes :

```
X=25 : Y=10 : PRINT X ; Y
25 10
OK
U=Y : Y=X : X=U : PRINT X ; Y
10 25
OK
```

Gardons cette bonne idée dans un coin de notre mémoire mais remarquons que l'instruction SWAP réalise le même travail :

```
A=10 : B=20 : PRINT A ; B
10 20
```

```
OK
SWAP A,B : PRINT A ; B
20 10
OK
```

Pour les curieux : Pour repérer une variable (ici Marignan), le MO6 utilise une adresse, c'est-à-dire un nombre. La mémoire vive est ainsi découpée en parcelles ayant chacune sa référence. Une case mémoire peut contenir un chiffre, un caractère, le contenu d'une zone de l'écran, une partie de l'instruction PRINT, etc. Le nombre 1515 est rangé quelque part dans la mémoire vive mais où ? Une simple instruction BASIC permet de le savoir : VARPTR.

```
PRINT VARPTR(MARIGNAN)
```

Le nombre que vous obtiendrez dépend de ce que vous aurez fait avant cette instruction. Il vous indiquera l'adresse de la première case mémoire où est rangé le contenu de Marignan. On pourrait imaginer de poursuivre plus loin notre investigation en explorant le contenu de la case mémoire indiquée. Le lecteur doit prendre ici connaissance des quelques secrets contenus au module 36. Le contenu d'une variable numérique est en général rangé dans des cases mémoire consécutives et est soumis à un codage mathématique-informatique subtil.

Rappelons ici que le BASIC laisse justement le soin au micro-ordinateur d'organiser le rangement pour éviter tout souci à l'utilisateur.

Restriction sur les noms des variables : Ils ne doivent pas commencer par un mot clé du BASIC (SCREEN, COLOR,...). La liste de ces mots est donnée en annexe 5.

Ainsi,

```
VALEUR=PRIX*REDUCTION
```

vous retourne le message "Syntax Error" car VALEUR commence par VAL qui est un des mots clés du BASIC.

Chaînes et autres variables

Les qualités de calculateur du MO6 ne sont plus à démontrer. Revenons sur le traitement des chaînes de caractères (voir module 4). Il est possible d'utiliser l'affectation pour elles aussi. La seule précaution impérative que doit prendre l'utilisateur est de faire suivre le nom de la chaîne qu'il désire ranger en mémoire, du signe \$ qui indiquera à l'interpréteur qu'il s'agit de stocker une chaîne de caractères et non pas un nombre. Soit donc à stocker la chaîne "MARGINAN" dans la mémoire sous le nom B\$.

```
B$="MARGINAN":PRINT B$:PRINT B$
MARGINAN
MARGINAN
OK
```

Tout ce qui a été dit précédemment sur les règles de l'affectation vaut également pour les variables chaînes de caractères. C'est surtout la présence du \$ en fin de nom de variable et des guillemets qui fait toute la différence. On comprendra facilement qu'il ne faut pas confondre les variables chaînes et les variables nombres. Le micro-ordinateur n'admet aucune confusion.

Les instructions du type D = "MARGINAN" et B\$ = 1515 provoquent automatiquement un message d'erreur.

Pour la même raison, tout transfert est impossible entre une variable chaîne et une variable nombre: A\$ = B ou B = A\$ provoquent des messages d'erreur.

Par contre, il est possible de réaliser des permutations entre chaînes de caractères de la même façon qu'entre les nombres.

```
A$="MON":B$="DE":PRINT A$:B$
MONDE
OK
SWAP A$,B$:PRINT A$:B$
DEMON
OK
```

Dans ce qui précède, une distinction importante apparaît entre les variables numériques et les variables chaînes de caractères (on dit aussi alphanumériques). Ces variables sont de types différents.

Il existe, pour les variables numériques, des sous-catégories qu'il peut être utile de connaître. Ces types correspondent à des techniques de stockage en mémoire particulières.

Variables entières

Elles peuvent être utilisées pour stocker des nombres entiers compris entre -32768 et 32767. Pour indiquer qu'une variable est de type entier, on peut procéder de deux façons différentes:

1. Faire suivre le nom de la variable du signe %.

Ex.: A % = 1854

2. Faire précéder l'affectation de la déclaration DEFINT qui dispense alors du signe %. Cette méthode permet de réaliser des arrondis automatiques.

```
DEFINT A:A=9.86:PRINT A
10
OK
```

Variables réelles en simple précision

C'est la règle pour les variables dont le type n'est pas précisé. Comme M. Jourdain faisait de la prose sans le savoir, depuis que vous manipulez des variables numériques, ce sont des variables réelles en simple précision.

Ces variables prennent en considération un maximum de sept chiffres significatifs. Leur nom peut être suivi d'un!

Variables réelles en double précision

Ce type de variable se manipule comme le précédent mais peut retenir jusqu'à quinze chiffres significatifs. Les noms de variables doivent être suivis d'un # ou leur type doit être déclaré préventivement par l'instruction DEFDBL. Ce type de variable peut être utilisé pour améliorer la précision de certains calculs comme les divisions. On en jugera sur l'exemple suivant.

```
DEFDBL X
B=10/7 : X=10#/7# : PRINT B : PRINT X
1.42857
1.428571428571429
OK
```

On observera que dans le cas du calcul en double précision, les constantes 10 et 7 doivent être suivies du signe # pour que l'opération s'effectue en double précision.

Mélange

Un même ordre d'affichage peut mêler des variables numériques et des chaînes de caractères.

```
A=48 : B=1024
PRINT A ; "X" : B ; "=" : A*B
48 X 1024=49152
OK
```

Opérations numériques

Commençons par l'exemple le plus simple.

```
A=15+16
OK
```

Derrière cette instruction simple se cache l'enchaînement suivant :

1. Effectuer l'opération $15 + 16$
2. Chercher un emplacement mémoire libre
3. Y ranger le résultat de l'opération
4. Lui donner le nom A

Pour s'assurer de la bonne réalisation de cette manœuvre :

```
PRINT A
31
OK
```

L'intérêt de ranger les résultats dans des mémoires grâce à des

variables vient du fait que ces résultats pourront être rappelés, donc réutilisés dans des calculs futurs. On peut demander encore davantage.

```
X=15 : Y=16 : Z=X+Y
OK
```

Résumons :

1. Ranger le nombre quinze dans la variable X.
2. Ranger le nombre seize dans la variable Y.
3. Prendre le contenu de X et celui de Y.
4. Ajouter ces deux nombres.
5. Ranger le résultat dans la variable Z.

Une écriture peu mathématique

La transformation du contenu d'une variable peut être faite à partir du contenu de cette variable elle-même. On dit alors qu'on a créé un compteur.

```
A=10 : PRINT A
10
OK
A=A+1 : PRINT A
11
OK
```

La deuxième ligne d'instruction a permis d'augmenter d'une unité le contenu de la variable A : prendre le contenu de A, lui ajouter 1, ranger le résultat dans A.

On dit que le compteur A a été incrémenté. On peut réaliser de la même façon une décrémentation.

```
A=10 : PRINT A
10
OK
A=A-1 : PRINT A
9
OK
```


Concaténation: +

L'opération de concaténation consiste à mettre bout à bout deux chaînes de façon à n'en former qu'une seule.

```
A$="ABCD" : B$="EFG"
PRINT A$+B$
ABCDEFG
OK
```

Le résultat d'une concaténation peut être rangé en mémoire. L'exemple suivant fabrique deux slogans S1\$ et S2\$. Le lecteur y apprendra donc que les noms de variables chaînes de caractères, tout comme les noms de variables numériques, peuvent comporter des lettres. Il prendra soin d'inclure des blancs dans les chaînes à concaténer pour séparer les mots.

```
A$="VIVE " : B$="A BAS " : M$="MACHINI"
S1$=A$+M$ : S2$=B$+M$
PRINT S1$ : PRINT S2$
VIVE MACHINI
A BAS MACHINI
OK
```

Partie gauche: LEFT\$

La fonction LEFT\$ donne la partie gauche d'une chaîne dont il faut préciser le nom ou le contenu, et le nombre de caractères à retenir.

```
PRINT LEFT$("ABCDEF",4)
ABCD
OK
```

On peut utiliser les variables en prenant soin que la partie gauche demandée ne contienne pas plus de caractères que la chaîne elle-même.

```
A$="BABA"
L1$=LEFT$(A$,1) : L2$=LEFT$(A$,2)
L3$=LEFT$(A$,3) : L4$=LEFT$(A$,4)
PRINT L1$ : PRINT L2$ : PRINT L3$ : PRINT L4$
```

```
B
BA
BAB
BABA
OK
```

Partie droite: RIGHT\$

La fonction RIGHT\$ donne la partie droite d'une chaîne et les modalités de son emploi sont les mêmes que pour LEFT\$.

```
PRINT RIGHT$("ABCDEF",4)
CDEF
OK
```

Morceau: MID\$

Pour extraire un morceau d'une chaîne, il faut préciser :

- De quelle chaîne il s'agit.
- Le rang du premier caractère retenu.
- Le nombre de caractères retenus.

Ainsi: MID\$(A\$, I, J) retiendra, dans la chaîne A\$, J caractères à partir du I^e.

```
PRINT MID$("ABCDEF",3,2)
CD
OK
PRINT MID$("ABCDEF",2,3)
BCD
OK
```

Longueur d'une chaîne: LEN

La fonction LEN renvoie la longueur d'une chaîne et peut être rangée dans une variable numérique: c'est toujours un nombre entier compris entre 0 et 255.

```
A$="ABCDE" : L=LEN(A$) : PRINT L
5
OK
```

Repérage d'une sous-chaîne dans une chaîne:

INSTR

Soient X\$ et Y\$, deux chaînes. La fonction INSTR est utilisée pour fournir la position de Y\$ dans X\$ (ou le contraire).

INSTR (X\$, Y\$) renvoie le rang dans X\$ à partir duquel Y\$ a été repéré.

```
PRINT INSTR ("BASIC", "S")
3
OK
```

On conclut que le S apparaît pour la première fois en troisième position dans "BASIC". Si la chaîne Y\$ n'est pas présente dans X\$, la fonction INSTR renvoie le nombre 0.

La fonction INSTR est très utile dans les programmes interactifs nécessitant une analyse de réponse. Après une question posée par le programme, on range la réponse dans une variable chaîne de caractères dans laquelle on peut tester la présence ou l'absence de mots clefs.

Changements de types

Les variables rangées en mémoire sont de deux types, différents et incompatibles: numérique ou chaîne. Deux fonctions permettent de passer d'un type à l'autre c'est-à-dire de transformer une variable numérique en une variable chaîne et inversement. Evidemment ces transformations ne sont pas magiques et ne peuvent en aucun cas transformer spontanément 1515 en Marignan!

Passage du type numérique au type chaîne par

STR\$

Si on met dans la variable X le nombre 123 par X=123, on transformera ce nombre en une chaîne de caractères par STR\$(X).

```
X=123 : A$=STR$(X)
PRINT A$
123
OK
```

Apparemment, c'est-à-dire sur le listing et sur l'écran, aucune différence n'est perceptible entre X et A\$. Mais qu'on ne s'y trompe pas! X est un nombre et A\$ une chaîne. A\$ ne peut être traité comme un nombre, on ne pourra le multiplier ni le diviser. En revanche, toutes les opérations de chaînes peuvent être utilisées avec A\$.

```
A=123 : B=45
X$=STR$(A) : Y$=STR$(B)
PRINT A+B : PRINT X$+Y$
168
123 45
OK
```

On notera ici la différence essentielle dans l'emploi de l'opération + : addition pour les variables numériques, concaténation pour les chaînes de caractères. Le blanc qui sépare 123 et 45 marque la place du signe + qui n'est pas écrit.

Passage du type chaîne au type numérique par

VAL

La fonction VAL ne transforme pas les lettres en nombres mais si un nombre, c'est-à-dire une suite de chiffres, est présent en début de chaîne, ce nombre sera extrait et il sera possible de le ranger dans une variable numérique.

Ainsi VAL ("1515 MARIGNAN") est égal au nombre 1515. Si la chaîne contient plusieurs nombres séparés par des caractères, c'est le premier de ces nombres qui sera retenu.

```
VAL("14 JUILLET 1789") est égal à 14
```

Compléments sur les chaînes

Montrons quelques exemples utilisant la "chirurgie" sur les chaînes.

Longueur d'un nombre

Combinée avec STR\$, la fonction LEN permet de connaître la longueur d'un nombre, c'est-à-dire le nombre de ses chiffres.

```
X=1515 : A$=STR$(X)
L=LEN(A$) - 1 : PRINT L
4
OK
```

On a enlevé 1 à LEN(A\$) en tenant compte de la place du signe.

Le vide et les blancs

Dans le module 4, nous avons souligné la différence entre les chaînes vides et les chaînes pleines de blancs. La fonction LEN permet de mesurer cette différence avec précision.

```
A1$="" : A2$=" " : A3$="   "
PRINT LEN(A1$) : LEN(A2$) : LEN(A3$)
0 1 4
OK
```

La chaîne A1\$ ne contient aucun caractère, la chaîne A2\$ en contient un et la chaîne A3\$ en contient quatre.

En combinant les fonctions VAL et STR\$, on peut ainsi pratiquer une "chirurgie" sur les chiffres composant un nombre.

Exemple: pour extraire le premier chiffre d'un nombre:

```
X=7259 : A$=STR$(X)
B$=LEFT$(A$,2) : PC=VAL(B$)
PRINT PC
7
OK
```

Commentaires

1. L'avantage de cette méthode est qu'elle traite un nombre quelconque X, même si l'utilisateur n'en connaît ni la longueur, ni la valeur.

2. Dans B\$, nous avons retenu les deux premiers caractères de A\$ en tenant compte du caractère blanc mis pour le signe dans A\$.

3. Si on a l'esprit de synthèse et un souci de concision, les instructions calculant X peuvent être résumées en une seule:

```
PRINT VAL(LEFT$(STR$(X),2))
```

On notera la cohérence dans l'ouverture et la fermeture des parenthèses.

Nous terminons cette partie consacrée aux chaînes de caractères et aux variables numériques en attirant votre attention sur un certain nombre de difficultés. Il ne faut jamais oublier que si vous fournissez de mauvaises données au MO6, celui-ci peut aussi bien vous signaler une erreur que faire n'importe quoi. Nous vous conseillons donc de vérifier la cohérence des résultats obtenus.

Précautions d'emploi

1. VAL(75) et VAL(X) provoquent une erreur de type (Type Mismatch).

Pour les mêmes raisons STR\$("12") et STR\$(A\$) sont des erreurs de type.

Lorsque le premier caractère d'une chaîne n'est ni un blanc, ni un chiffre, ni un point, la valeur VAL de cette chaîne est 0. Ainsi VAL("A5") est égal à 0.

Attention également: sur le MO6, on aura VAL("2+3") = 2. Alors que l'on pouvait espérer que le BASIC aurait le bon goût d'effectuer l'opération et de fournir comme résultat VAL("2+3") = 5; il n'en est rien!

En revanche, il existe une fonction EVAL (associée à CRUNCH\$) qui effectue l'opération. Reportez-vous à la partie "Références" mais notez bien que cette instruction ne fonctionne pas en mode direct.

Cet exemple doit attirer votre attention sur le fait qu'il ne faut jamais se fier aux évidences et dire que le micro-ordinateur ne fait pas ce qu'on lui demande de faire. Sa logique n'est pas toujours la vôtre et le plus logique n'est pas toujours celui qui le dit!

2. Même si la machine accepte `PRINT LEFT$(A$,0)` et répond "correctement" à `PRINT RIGHT$("123",5)` en donnant "123", il est préférable de ne pas lui donner d'arguments fantaisistes: sa courtoisie a des limites.

Dernière remarque sur les affectations

Reprenons les exemples concernant les variables numériques: A partir d'une ou plusieurs variables numériques, on peut réaliser toutes sortes de manipulations opératoires: des calculs et des transferts. Le micro-ordinateur ne s'y perd jamais. Le programmeur devra tenter d'en faire autant...

Voici quelques exemples:

```
A=14 : B=12 : C=16
MOY=(A+B+C)/3 : PRINT MOY
14
OK
```

```
A=3.5 : B=A+A : C=B+B
PRINT A,B,C
3.5      7      14
OK
```

Pouvez-vous expliquer la différence entre ces deux exemples?

```
A=1 : B=2 : C=A+B : B=C : A=B
PRINT A,B,C
3      3      3
OK
```

```
A=1 : B=2 : C=A+B : A=B : B=C
PRINT A,B,C
2      3      3
OK
```

On voit donc que l'ordre dans lequel sont écrites les instructions d'affectation joue un rôle très important.

La réflexion et l'habitude vous aideront à surmonter les inévitables erreurs des premières tentatives.

ASCII se prononce "à ski" mais n'a rien à voir avec les sports d'hiver puisque c'est un mot formé des initiales de *American Standard Code Information Interchange*.

Le code ASCII donne une représentation numérique des caractères : lettres, majuscules et minuscules, chiffres, signes d'opérations et de ponctuations, caractères de contrôle. On en trouvera une description complète dans l'annexe 4.

On retrouve ici la notion de codage d'une information que l'on avait introduite avec les couleurs de l'écran.

Un morceau élémentaire d'information, équivalent à OUI ou NON, s'appelle un bit. Les lettres de l'alphabet, les chiffres, les symboles d'une opération, etc., sont traduits par une série de bits. Le MO6, et plus particulièrement le microprocesseur qui en est le cerveau, sait interpréter une suite de "0" et de "1". Il sait également les traiter, les stocker, les restituer quand on le lui demande.

Quand vous tapez le caractère A (en majuscule), le MO6 traite la suite de bits suivante :

1	0	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Plus précisément, le microprocesseur traite des paquets de huit bits appelés octets; ainsi le codage complet du caractère A sera :

0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Le code ASCII permet de représenter tous les caractères du clavier à l'aide d'un octet. Pour les mathématiciens connaissant le principe de numération en base 2, on peut remarquer que l'octet proposé pour coder la lettre A représente la décomposition du nombre 65 en base 2.

C'est ainsi que le code ASCII donne une valeur numérique à chaque caractère disponible sur le clavier du MO6 et même d'autres.

Deux fonctions BASIC très courantes permettent de connaître le code ASCII d'un caractère et réciproquement, de retrouver un caractère dont on connaît le code ASCII. Ces deux fonctions, tout comme VAL et STR\$, ont pour effet de transformer une variable numérique en une variable chaîne de caractères et inversement.

ASC

Le code ASCII du caractère A est 65. La fonction ASC le confirme.

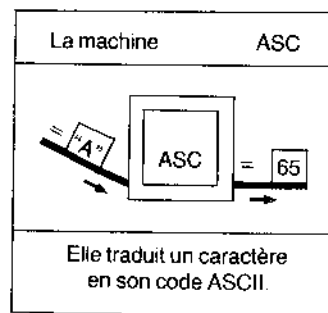
```
PRINT ASC("A")
65
OK
```

Plus surprenant :

```
PRINT ASC("7")
55
OK
```

En effet, 55 est le code ASCII du caractère 7.

On peut illustrer cette transformation par la machine ASC :



La fonction ASC peut aussi s'appliquer à des chaînes de plus d'un caractère. Dans ce cas, elle donne le code du premier caractère de la chaîne.

```
PRINT ASC("MARIGNAN")
77
OK
```

Etonnant!

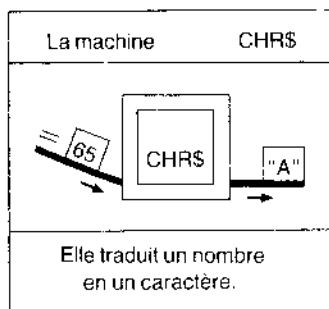
```
PRINT ASC(" ")
32
OK
```

Le premier caractère de la chaîne est un espace. Son code ASCII est 32. Si vous cherchez le code ASCII d'une chaîne vide, vous aurez droit à un message d'erreur: le vide n'est pas codé.

CHR\$

C'est la fonction réciproque de ASC. Elle donne le caractère correspondant au code fourni.

```
PRINT CHR$(65)
A
OK
```



On peut rire!

```
PRINT CHR$(49) : CHR$(43) : CHR$(49) : CHR$(51) : CHR$(51)
1+1=3
OK
```

Le MO6 vient d'écrire quelque chose d'assez stupide mais le pauvre malheureux ne s'en est pas rendu compte. CHR\$ renvoie toujours une chaîne de caractères qui peut être traitée comme toutes les chaînes de caractères.

Exemple:

```
A$ = CHR$(66) + CHR$(65) : A$ = A$ + A$
PRINT A$
BABA
OK
```

Précautions d'emploi

CHR\$ agit sur un nombre et ASC sur une chaîne de caractères. De ce fait: ASC(29), ASC(X), CHR\$("A"), CHR\$(M\$) provoquent automatiquement des erreurs de type (Type Mismatch).

Récréations

Deux récréations un peu sottes qui n'étonneront pas le lecteur attentif.

```
PRINT ASC(CHR$(65))
65
OK

PRINT CHR$(ASC("A"))
A
OK
```

Caractères de contrôle

Les caractères dont le code ASCII est entre 0 et 31 sont des caractères de contrôle. Ils n'affichent rien mais agissent directement sur l'ordinateur.

Exemple:

```
PRINT CHR$(7) provoque un bip
PRINT CHR$(12) efface l'écran
PRINT CHR$(30) envoie le curseur en haut et à gauche
```

Guillemets

Le code ASCII est formel: pour afficher les guillemets, il faut taper

```
PRINT CHR$(34)
```

C'est la seule solution mais elle est bonne!

L'un des principaux attraits du MO6 est la faculté qu'il donne à l'utilisateur de pouvoir créer des images.

Pour utiliser cet aspect graphique, il faut bien sûr connaître les fonctions graphiques du BASIC, mais aussi se rappeler (ou apprendre) les principes élémentaires de la représentation cartésienne des points.

Rappelons-les rapidement.

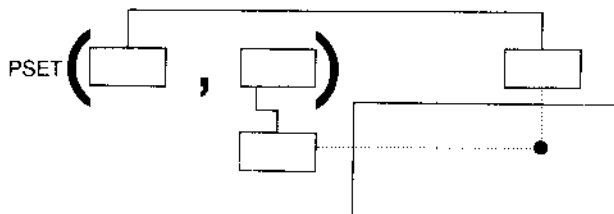
Chaque point (pixel) de l'écran est repéré par deux nombres qu'on appelle ses coordonnées. La coutume veut que la première coordonnée soit le repérage horizontal (abscisse) et la deuxième, le repérage vertical (ordonnée). Chaque coordonnée est un nombre correspondant à un numéro de graduation.

Pour le graphique, on parle d'écran haute résolution : plus le nombre de colonnes et de lignes est grand, meilleure est la précision et la richesse du graphique. Le MO6 offre 3 modes graphiques différents :

- 320 colonnes et 200 lignes
- 640 colonnes et 200 lignes
- 160 colonnes et 200 lignes

Allumer un point

On allume un point grâce à l'instruction : PSET(X,Y) où X est un numéro de colonne et Y, un numéro de ligne.



Pour modifier cette couleur, il suffit de faire suivre cette instruction du code de la couleur désirée.

Ainsi : PSET(200,100),1 ← point rouge
PSET(200,100),15 ← point orange

Tirer une ligne droite

Pour tracer une ligne droite, on utilise l'instruction LINE suivie des coordonnées de deux points qui seront les extrémités de la ligne.

Ainsi : LINE(50,50) – (250,150)

trace une ligne droite allant du point de coordonnées (50,50) à celui dont les coordonnées sont (250,150).

On notera avec profit que l'instruction LINE (250,150) – (50,50) aboutirait exactement au même résultat.

Exactement comme pour l'instruction PSET, la couleur peut être précisée à la fin de l'instruction.

Ainsi : LINE(0,0) – (319,199),4 trace une diagonale bleue.

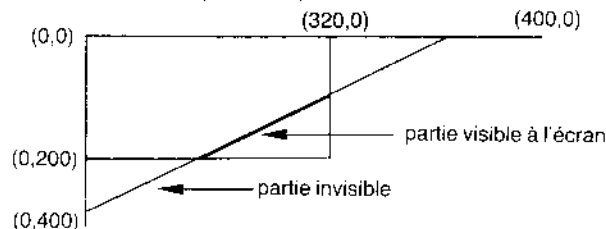
Il est permis d'omettre le point de départ de la ligne. Dans ce cas, c'est le dernier point marqué qui en fait office.

Ainsi : LINE(10,20) – (50,100) : LINE – (150,50)

tracera une ligne du point (10,20) au point (50,100) puis une autre du point (50,100) au point (150,50).

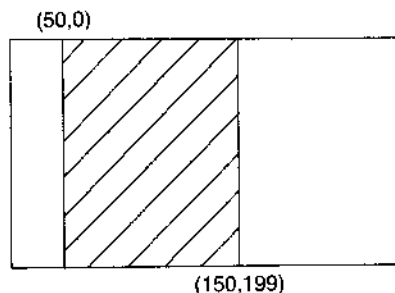
Remarque : l'instruction LINE (comme toutes les instructions graphiques) accepte des numéros de colonnes et de lignes entre –32768 et +32767. Ainsi, on peut dessiner en dehors de l'écran visible.

Exemple : LINE(400,0) – (0,400) donnera



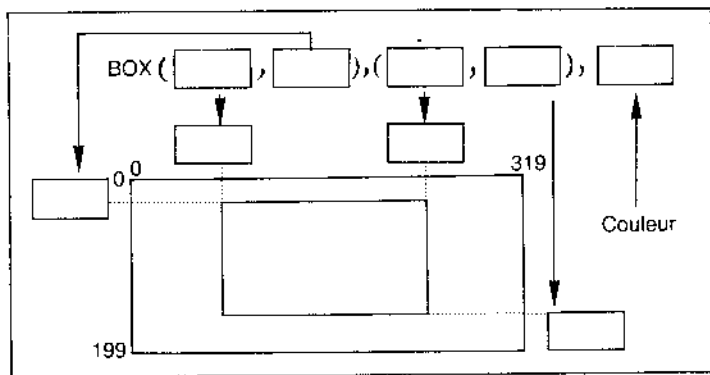
Puisqu'il est possible de dessiner en dehors de l'écran visible, c'est un peu comme si on voyait une partie d'un grand écran (de –32768

à +32767 de chaque côté) à travers une petite fenêtre de la taille de l'écran (de 0 à 319 en colonnes et de 0 à 199 en lignes).
 L'instruction WINDOW (fenêtre) permet de réduire encore la taille de la fenêtre et de la localiser où l'on veut dans la partie visible.
 L'instruction WINDOW (50,0)–(150,199) définit un rectangle-fenêtre dans lequel les graphiques viendront se dessiner.



Pour revenir à l'état initial, il suffit de faire :
 WINDOW(0,0)–(319,199)

Dessiner des boîtes



Pour obtenir un rectangle, on utilise l'instruction BOX (anglais pour boîte) suivie des coordonnées de deux points désignant deux sommets opposés. Les côtés du rectangle auront leurs côtés parallèles aux bords de l'écran.

En faisant quelques essais, on se convaincra facilement qu'étant donné deux points quelconques sur l'écran il n'existe qu'un seul rectangle admettant ces deux points comme sommets opposés. L'ordre dans lequel on donne ces deux points est donc sans importance.

Comme pour PSET et LINE, l'instruction BOX peut être suivie d'un code couleur.

Boîtes pleines

Le rectangle sera entièrement coloré si l'on remplace BOX par BOXF. Le F étant mis pour *Fill*, en anglais : remplir.

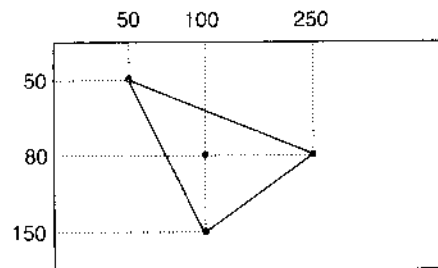
Peindre

Il est possible de remplir l'intérieur d'une courbe fermée à l'aide de l'instruction PAINT (X,Y) où X et Y sont les coordonnées d'un point "intérieur" de la courbe.

Exemple :

Pour colorier l'intérieur d'un triangle.

```
LINE(50,50)--(250,80),4 : LINE--(100,150),4 :  
LINE--(50,50),4 : PAINT(100,80),4
```



Tracer un cercle...

Le BASIC du MO6 dispose d'une instruction CIRCLE qui permet de dessiner un cercle dont on fournit les coordonnées du centre et le rayon.

Exemple :

```
CIRCLE(200,150),30
```

dessine le cercle de centre (200, 150) et de rayon 30.

Sa couleur peut être différente de celle des caractères si on précise l'argument suivant :

```
CIRCLE(200,150),30,2
```

Le rayon ainsi que les coordonnées du centre et la couleur peuvent être définis par des variables.

une ellipse...

La même instruction CIRCLE permet de tracer des ellipses à condition de préciser deux paramètres supplémentaires.

Si RH et RV désignent les "rayons" horizontaux et verticaux, l'ellipse sera obtenue par :

```
CIRCLE(C,L)RH,RV
```

Attention à la syntaxe : pas de virgule avant RH ; c'est cette absence de virgule qui précise que l'instruction doit dessiner une ellipse et non un cercle.

Exemple :

```
CIRCLE(160,100)70,50
```

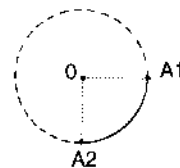
trace une ellipse centrée en 160,100 dont l'axe horizontal fait 70 et l'axe vertical fait 50.

On peut évidemment préciser la couleur grâce à l'argument mis à la suite.

On remarquera que le tracé de l'ellipse n'utilise pas les foyers de l'ellipse. Un bon sujet de réflexion consisterait à les utiliser...

...et un arc de cercle.

Mais mieux encore, vous pouvez dessiner n'importe quelle portion de cercle en n'importe quel point de l'écran.



Une portion de cercle, par exemple A1 A2, est limitée par deux rayons OA1 et OA2. La position de ces deux rayons sera précisée par les angles qu'ils font avec l'horizontale : 0° pour OA1, 90° pour OA2.

Pour l'ordinateur, les angles doivent être exprimés en radians.

Sachant que $\pi = 3,14$ radians correspondent à 180° , il ne reste plus qu'à faire une règle de 3.

Ainsi $90^\circ = \pi/2 \text{ rd} = 1,57 \text{ rd}$

Si le cercle est placé au centre de l'écran, l'arc de cercle A1 A2 sera obtenu par :

```
CIRCLE (160,100),50:0,1.57
```

Les deux angles limites sont précisés après le rayon et séparés de celui-ci par un point-virgule.

Remarquez que les angles sont comptés dans le sens des aiguilles d'une montre et à partir du point A1. L'axe des y étant orienté vers le bas, on retrouve ici le sens trigonométrique auquel les mathématiciens sont bien habitués.

Choisir son motif : PATTERN

Une boîte ou un cercle peuvent également se remplir avec un motif :

Exemple :

```
PATTERN "/"
```

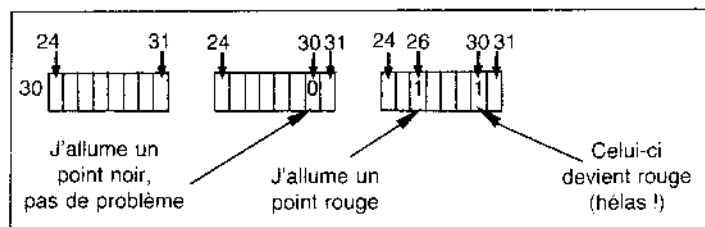
```
BOXF(40,50)-(200,150)
```

remplira le rectangle désigné avec des barres obliques.

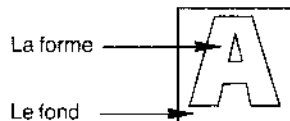
Remarque : tous les caractères peuvent servir de motif de remplissage.

Rigidité des couleurs

Supposons que l'écran ait sa couleur d'origine (cyan : 6).
L'instruction PSET (30,30), 0 allume un point noir en (30,30).
L'instruction PSET (26,30), 1 allume un point rouge à gauche du précédent. En y regardant de plus près, on s'aperçoit que le point noir précédent est lui aussi passé au rouge ! Mauvaise surprise ! Il faut imaginer un morcellement de l'écran en petites barres horizontales de 8 points (encore un octet). La contrainte des couleurs est la suivante : chaque groupe de 8 points ne peut recevoir que 2 couleurs, c'est le dernier point allumé qui contamine les autres.



Peut-être pensez-vous que le noir et le rouge, au fond ça ne fait que deux couleurs et que le MO6 aurait dû les accepter. C'est sans compter sur la couleur du fond d'écran (cyan) ; avec elle, il y a bien trois couleurs présentes sur cet octet et c'est donc une de trop. On n'y pense pas assez souvent : pour écrire, il faut deux couleurs, une pour l'encre, une pour le papier.



Trois des modes proposés résolvent le problème de la rigidité. Ils sont accessibles par l'instruction CONSOLE (voir Références). L'un offre deux couleurs mais double le nombre de pixels, le second ne permet que quatre couleurs sans rigidité, le troisième offre seize couleurs sans rigidité en limitant le nombre de colonnes à 160.

Passage d'une ligne à travers une boîte

Exemple 1 :

```
SCREEN 4,11,11  
BOXF(48,50)-(199,150),0  
LINE(0,0)-(319,199),1
```

La perturbation observée lors du passage de la ligne jaune dans la boîte noire tient à la rigidité des couleurs. Tout point jaune allumé dans la boîte colore en jaune tous les points noirs du segment auquel il appartient.

Pour remédier à ce fâcheux inconvénient, il suffit d'allumer les points de la boîte dans une couleur de fond qui tolérera le passage des points jaunes en couleur de forme.

Exemple 2 :

```
SCREEN 4,11,11  
BOXF(48,50)-(199,150),-1  
LINE(0,0)-(319,199),1
```

un code couleur négatif pour une couleur de fond.

A retenir : colorier en couleur de "fond" se fait en donnant un code négatif aux couleurs : $-(C+1)$ pour la couleur de numéro C.

On obtient donc le codage suivant :

Noir : -1
Rouge : -2
Vert : -3
etc.

Au secours, une tortue !

La tortue est un personnage célèbre... du langage LOGO. Le BASIC

du MO6 vous permet d'utiliser également une tortue : on peut

- la faire apparaître en tout point de l'écran,
- lui demander d'avancer, de reculer,
- la faire changer de taille ou disparaître.

On peut même lui donner une forme étrange et en faire un objet animé, rapide et surprenant. Son intérêt ? C'est de laisser une trace derrière elle et de fabriquer ainsi des graphiques singulièrement compliqués.

Faire apparaître une tortue :

TURTLE, SHOW, FWD

Au départ, une tortue est simplement représentée par un triangle isocèle. Appelons la première tortue en mode direct avec l'instruction TURTLE :

TURTLE 0

Elle est prête à se déplacer, à dessiner... mais invisible. On la visualise grâce à l'instruction

SHOW 1

Le triangle apparaît au centre de l'écran, position de départ standard pour les tortues. Si vous changez SHOW 1 en SHOW 0, la tortue redevient invisible.

L'instruction FWD (mise pour *forward*, en avant) suivie d'un nombre fait avancer la tortue de cette quantité. Si le nombre précisé est négatif, alors la tortue reculera. Le déplacement se fait vers la droite, direction du nez de la tortue ; et il est plutôt rapide !

Dessiner avec la tortue : TRACE, HEAD

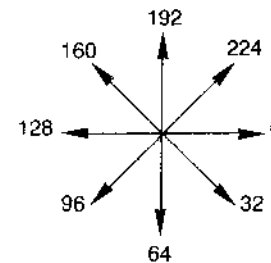
En se déplaçant, la tortue peut baisser le crayon afin de matérialiser sa trajectoire. L'ordre correspondant est TRACE 1 (elle lève son

crayon avec TRACE 0). Voici une commande qui appelle la tortue, la montre, lui fait baisser le crayon et enfin la fait avancer de 50 points :

TURTLE 0 : SHOW 1 : TRACE 1 : FWD 50

La tortue apparaît au centre de l'écran. Remarquez que le crayon est situé au centre de la tortue. En remplaçant TRACE 1 par TRACE 0, la tortue avance sans dessiner.

La direction du mouvement est modifiable par l'instruction HEAD (tête) suivie d'un nombre précisant de combien il faut tourner : un tour complet (360°) correspond à 256, un demi-tour (180°) à 128, un quart de tour (90°) à 64, un huitième de tour (45°) à 32 et ainsi de suite.



Ainsi HEAD 64 signifie : prendre la direction à 90° vers la droite et HEAD -64, prendre la direction à 90° vers la gauche.

Dans l'exemple précédent, on remarque que la tortue a fait son apparition au milieu de l'écran. Il est possible de déplacer le point de départ de la tortue en précisant les coordonnées du point de départ désiré dans l'instruction TURTLE après le numéro de la tortue :

TURTLE 0,160,20

Faire tourner la tortue sur elle-même : ROT

Lorsque la tortue se déplace, elle garde la tête dirigée vers la droite, quelle que soit sa direction. L'instruction HEAD change sa direction mais pas son orientation propre. C'est l'instruction ROT qui la fait tourner autour de son centre. Cette instruction est suivie d'un nombre qui précise de combien doit tourner la tortue par rapport à son orientation précédente.

Ainsi, pour que la tortue suive l'orientation de son trajet, il suffit de faire suivre chaque instruction HEAD par une instruction ROT de même valeur.

Un effet de ZOOM

La taille de chaque tortue peut varier continûment sur une échelle de 0 à 255 :

- 16 correspond à l'échelle 1, c'est-à-dire la taille standard
- 32 correspond à l'échelle 2
- 48 correspond à l'échelle 3
- ...
- 255 correspond à l'échelle 16.

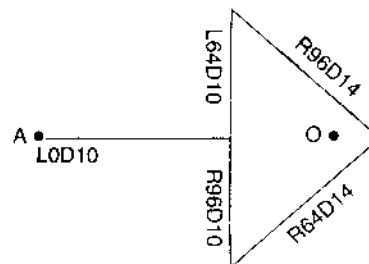
L'instruction ZOOM suivie d'un nombre fixe la taille de la tortue.

Une nouvelle tortue

Créons une deuxième tortue de forme différente. Elle sera définie dans l'instruction TURTLE par une chaîne de caractères précisant la suite des segments nécessaires pour dessiner cette tortue. Pour la distinguer de la première, on lui affecte un numéro différent. Ce numéro est indiqué immédiatement après TURTLE.

Exemple :

```
TURTLE 2 "L10D10L64D10R96D14R64D14R96D10" .SHOW 1,1
```



La chaîne définissant une tortue est formée d'une suite de doublets. Chaque doublet comprend deux éléments : le premier indique de combien il faut tourner, le second de combien il faut avancer. Les deux éléments sont obligatoires.

L0D10 signifie :

- tourne à gauche (*Left*) de 0
- avance en traçant (*Down*) de 10.

R96D14 signifie :

- tourne à droite (*Right*) de 96
- avance en traçant (*Down*) de 14.

Si nous désirons faire tourner cette nouvelle tortue, la rotation s'effectuera autour du point de départ A du tracé. Avec le triangle de la tortue standard, la rotation se faisait autour du centre du triangle. Pour faire tourner notre tortue sur elle-même autour d'un autre point O plus central, on ajoute le doublet L128U15 qui signifie :

- change l'orientation de départ de 0 en 128
- avance sans tracer (*Up*) de 15.

```
TURTLE 2, "L128U15L128D10L64D10R96R64D14R96D10"
```

Cette fois, la tortue tourne autour du point O. Ce point de départ du tracé d'une tortue est aussi la position de son "crayon".

Module 14

De la musique

PLAY

L'instruction PLAY permet de faire jouer des notes de musique. La suite des notes jouées doit être donnée derrière PLAY sous la forme d'une chaîne de caractères. Cette chaîne peut être une constante (les notes placées entre guillemets) ou une variable préalablement définie. Les notes doivent être écrites sans séparation. Elles peuvent être choisies parmi :

DO RE MI FA SO LA SI

On remarquera que toutes ces notes sont écrites avec deux lettres, même Sol qui a été abrégé. Par exemple, pour jouer le début de la gamme, on pourra écrire :

PLAY "DOREMI"

ou bien :

A\$="DOREMI"; PLAY A\$

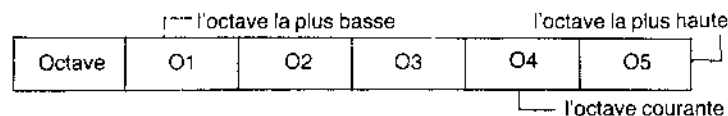
Silence

Pour jouer un silence, c'est-à-dire une note que l'on n'entend pas, on utilise la lettre P. Par exemple, pour jouer les notes Do, Ré, Mi séparées par un silence, écrire :

PLAY "DOPREPMIP"

Octaves

Les notes peuvent être jouées sur cinq octaves différentes, c'est-à-dire à cinq hauteurs différentes. L'octave la plus basse est codée 1 et la plus haute est codée 5. Lorsqu'on ne précise pas, l'octave est 4. Pour choisir l'octave, on intercale dans la chaîne la lettre O suivie du numéro.



Par exemple, pour jouer la note La aux cinq octaves permises, on écrit :

PLAY "O1LAO2LAO3LAO4LAO5LA"

Longueur

Les notes peuvent être jouées avec des longueurs différentes. La longueur la plus petite est codée 1 et la plus longue est codée 96. Lorsqu'on ne précise pas, la longueur est 24. Cette longueur correspond à ce que les musiciens appellent une noire. La longueur double (48) correspond donc à une blanche. Pour choisir la longueur, on intercale dans la chaîne la lettre L suivie du nombre correspondant à la longueur choisie.

Longueur	L96	L48	L24	L12
Note	ronde	blanche	noire	croche

Longueur courante

Attaque

La note peut être jouée pendant toute sa longueur (son continu) ou seulement au début (attaque). L'attaque est codée par un nombre entre 0 et 255. Lorsqu'on ne précise pas, l'attaque est 0 et correspond à un son continu. Pour choisir l'attaque, on intercale dans la chaîne la lettre A suivie du nombre correspondant à l'attaque choisie.

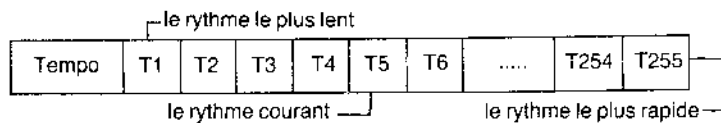
Attaque	A0	A1	A2	A3	A4	A254	A255
---------	----	----	----	----	----	-------	------	------

son continu attaque courante

Remarque : il est inutile de choisir des attaques au-delà de 100 car la note jouée n'est alors pas audible.

Tempo

Le tempo permet de régler la vitesse d'exécution d'une suite de notes. Le tempo est fixé par un nombre entre 1 (le plus rapide) et 255 (le plus lent). Lorsqu'on ne précise pas, le tempo est 5. Pour choisir le tempo, on intercale dans la chaîne la lettre T suivie du nombre correspondant au rythme choisi.



Dièse et bémol

Pour obtenir toutes les notes d'une gamme, on doit utiliser le caractère # pour dièse et le caractère b pour bémol juste après le nom de la note. Nous n'utilisons pas le bémol : une note suivie d'un bémol sera représentée par la précédente suivie d'un dièse.

Claire fontaine

Un détail important : avant d'annoncer quelle note doit être jouée, il faut commencer par donner sa longueur. Mettez-vous quelques secondes à la place de la machine et vous comprendrez qu'il est nécessaire de connaître la longueur avant la note. Voici quelques exemples de traductions :

Solfège				
Français	Noire - Sol	Blanche - La	Croche - Mi	Ronde - Mi
BASIC	L24SO	L48LA	L12MI	L96MI

Voici une partition simplifiée du refrain et la traduction BASIC de chacune de ses notes.

Lorsqu'une octave, une longueur, une attaque ou un tempo ont été précisés, ils le restent jusqu'à la prochaine indication contraire. Jouez !

```
M1$ = "L96DOL48DOMI"
M2$ = "MIL24REDGL48MIDO"
M3$ = "L96MIL48MIL24REDG"
M4$ = "L48MIREL96DO"
PLAY M1$ + M2$ + M3$ + M4$
```

La première chaîne définie (M1\$) contient les trois notes de la première mesure. La première note est une blanche (L96), les deux suivantes sont des noires (L48, c'est-à-dire la moitié d'une blanche). Vous observerez que la longueur de la troisième note (MI) n'est pas précisée. C'est la longueur précédemment définie (48) qui restera valable pour elle, comme pour la première note de la deuxième mesure (variable M2\$).

Module 15

Premières boucles

Nous avons pris soin jusqu'à présent, afin de préserver l'enthousiasme du lecteur, de ne pas lui demander de frapper des séries d'instructions trop importantes. Les résultats obtenus sur l'écran n'ont pas été spectaculaires, du moins par leur volume. Il est temps de montrer que, grâce au langage BASIC, on peut afficher beaucoup de caractères à peu de frais. Grâce aux boucles.

```
FOR K=1 TO 5 PRINT "BABA" NEXT K
```

```
BABA  
BABA  
BABA  
BABA  
BABA  
OK
```

Les nouveautés de cette ligne d'instructions sont : FOR, TO et NEXT. Ces trois mots permettent de construire des boucles, c'est-à-dire de répéter. Dans une répétition, les facteurs importants sont :

- ce qu'il faut répéter,
- combien de fois le répéter.

Avant de donner les règles précises qui doivent présider à la constitution des boucles, nous ajoutons quelques exemples au précédent en invitant le lecteur à fabriquer les siens, à les essayer, à les modifier.

```
FOR K=1 TO 100:PRINT "A":NEXT K
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA  
AAAAAAAAAAAAAAAAAAAAAAAAAAAA  
OK
```

Le résultat n'est pas enthousiasmant mais nous sommes sur la bonne voie.

```
FOR N=0 TO 50:PRINT N:NEXT N
```

```
0 1 2 3 4 5 6 7 8 9 10 11  
12 13 14 15 16 17 18 19 20 21  
22 23 24 25 26 27 28 29 30 31  
32 33 34 35 36 37 38 39 40 41  
42 43 44 45 46 47 48 49 50  
OK
```

```
FOR U = -10 TO 10:PRINT U*2::NEXT U
```

```
-20 -18 -16 -14 -12 -10 -8 -6 -4 -2 0  
2 4 6 8 10 12 14 16 18 20
```

```
D = 25 F = 50
```

```
FOR J = D TO F:PRINT J::NEXT J
```

```
25 26 27 28 29 30 31 32 33 34  
35 36 37 38 39 40 41 42 43 44  
45 46 47 48 49 50  
OK
```

```
FOR M = 1 TO 5:PRINT M;"KM":NEXT M
```

```
1 KM  
2 KM  
3 KM  
4 KM  
5 KM  
OK
```

Essayons d'afficher des bandes horizontales de couleur. Une bande sera obtenue en imprimant une série d'"espaces" dont la couleur de fond est modifiée par une instruction COLOR. Si nous voulons voir toutes les couleurs à l'écran, nous savons d'avance que cette même action sera répétée seize fois.

Par exemple, pour une bande rouge :

```
COLOR 1:PRINT " ";
```

L'instruction COLOR ne modifie ici que le fond et laisse inchangée la couleur des caractères; l'instruction PRINT affiche ensuite douze "blancs" de couleur rouge.

Exemple pour plusieurs couleurs :

```
FOR C = 0 TO 15 : COLOR,C : PRINT " " : NEXT C
```

Avec la tortue

Regardez un dessin se faire à l'écran à l'aide de la tortue : l'aspect répétitif des boucles fait que nous choisirons de dessiner des figures géométriques régulières, par exemple un octogone. Il sera formé de segments de même longueur, de directions successives décalées de 45°.

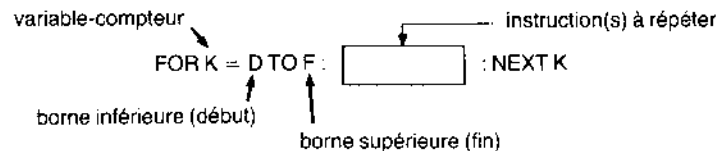
```
TURTLE 0 : TRACE 1 : FOR N = 1 TO 8 : FWD 30 : HEAD 32 : NEXT N
```

En remplaçant HEAD 32 par HEAD 16 et la valeur maximale de N par 16, on obtient un polygone à 16 côtés, mais une partie de la figure est en dehors de l'écran. Il faut donc déplacer le point de départ de la tortue en précisant les coordonnées dans l'instruction TURTLE. Par exemple :

```
TURTLE 0,150,20
```

Mettons les points sur les i

Une boucle commence par le mot FOR suivi d'un nom de variable numérique et se termine par NEXT suivi du même nom de variable. Entre ces deux instructions, il faut écrire la ou les instructions à répéter. La borne inférieure et la borne supérieure sont deux valeurs numériques encadrant TO.



Pour exécuter cette boucle, la machine procède comme suit :

1 — Comparaison entre le contenu de K et de F. Si K est inférieur à F, alors aller en 2, sinon arrêter tout.

2 — idem 1.

3 — idem 2.

4 — Incrémentation de K d'une unité ($K = K + 1$). Aller en 1.

Changement de pas

Dans les boucles précédentes, le pas est de 1 : la variable du compteur est régulièrement augmentée de 1. Il est possible d'augmenter ce pas en construisant des boucles comptant de 2 en 2, de 10 en 10 ou de 0,4 en 0,4. Puisqu'en anglais le pas se dit STEP, nous aurons les instructions suivantes :

```
FOR K = 0 TO 10 STEP 2 : PRINT K : NEXT K  
0 2 4 6 8 10  
OK
```

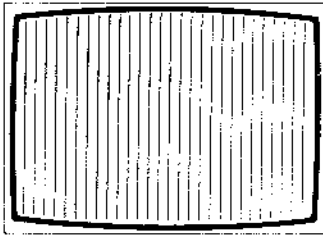
```
FOR I = 4 TO 5 STEP 0.2 : PRINT I : NEXT I  
4 4.2 4.4 4.6 4.8 5  
OK
```

```
FOR J = 5 TO 10 STEP 20 : PRINT J : NEXT J  
5  
OK
```

```
FOR S = 10 TO 0 STEP -1 : PRINT S : NEXT S  
10 9 8 7 6 5 4 3 2 1 0  
OK
```

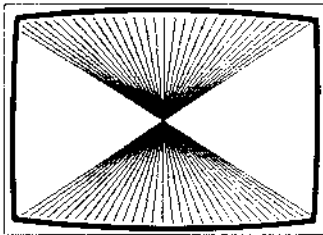

Des lignes plein l'écran

```
FOR C = 0 TO 320 STEP 10 : LINE(C,0) -- (C,199) : NEXT C
```



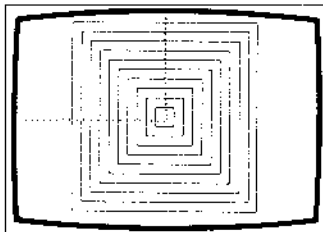
En changeant l'instruction à répéter, on obtient des motifs différents.

```
LINE(C,0) -- (319-C,199)
```



```
BOX(160-C,100-C) - (160+C,100+C)
```

160



100

Remarque importante : aussi bien pour le pas que pour les bornes, on peut utiliser des noms de variables numériques.

```
A = 1 : B = 2
FOR K = B -- A TO B + A STEP A : PRINT K : NEXT K
1
2
3
OK
```

Epluchage d'un mot

```
MOT$ = "BASIC" : L = LEN(MOT$)
FOR K = 1 TO L : PRINT MID$(MOT$,K,1) : NEXT K
```

```
B
A
S
I
C
OK
```

Remarque : MID\$(MOT\$,K,1) est en fait la K-ième lettre de MOT\$.

Pyramide

```
MOT$ = "BASIC" : L = LEN(MOT$)
FOR K = 1 TO L : PRINT LEFT$(MOT$,K) : NEXT K
```

```
B
BA
BAS
BASI
BASIC
OK
```

Perdre du temps

```
FOR K = 1 TO 1000 : NEXT K
OK
```

Curieuse boucle en vérité puisqu'on n'y fait rien sinon compter de 1 à 1000. L'utilisateur ne verra donc rien mais il constatera que la

machine y passe un certain temps qui dépend évidemment de la différence entre les bornes.

Ce qui est déconseillé

Un pas nul.

Une incohérence entre les bornes et le pas. Si D est la borne inférieure, F la borne supérieure et P le pas, on doit avoir :

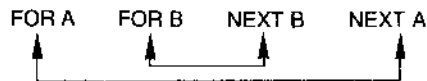
- si $D < F$ alors $P > 0$
- si $D > F$ alors $P < 0$

Dans le cas contraire, la boucle n'est pas effectuée et on passe l'instruction suivant NEXT.

On ne doit pas changer la variable-compteur dans les instructions à répéter. Si par exemple cette variable est K, toute instruction du type $K=K+1$ est à proscrire.

Boucles gigognes

Il est possible d'imbriquer deux boucles l'une dans l'autre à la manière des tables gigognes. La série d'instructions comporte alors deux FOR, deux NEXT et deux variables-compteur.



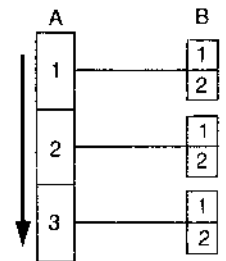
Ce schéma résume assez bien le seul point sur lequel il est nécessaire d'insister : le programmeur doit absolument respecter l'imbrication des boucles. L'erreur la plus fréquente consiste à permuter l'ordre de NEXT B et de NEXT A.

Dans le mode direct avec lequel nous travaillons, les boucles gigognes imposent des lignes d'instructions assez longues. On ne doit pas s'en effrayer. Il faut cependant ne pas oublier de taper sur ENTREE seulement après la dernière instruction même si l'ensemble occupe plusieurs lignes de l'écran.

```
FOR A=1 TO 3: FOR B=1 TO 2: PRINT "M" :
NEXT B: NEXT A
MMMMMM
OK
```

On obtient ainsi six fois la lettre M en ligne. La boucle B est effectuée trois fois dans la boucle A. C'est-à-dire qu'on imprime trois fois deux fois la lettre M.

Le déroulement des compteurs A et B peut être représenté par un arbre :



Tables de multiplication

```
FOR A=1 TO 9: FOR B=1 TO 9: PRINT A*B :
NEXT B: PRINT : NEXT A
1 2 3 4 5 6 7 8 9
2 4 6 8 10 12 14 16 18
3 6 9 12 15 18 21 24 27
4 8 12 16 20 24 28 32 36
5 10 15 20 25 30 35 40 45
6 12 18 24 30 36 42 48 54
7 14 21 28 35 42 49 56 65
8 16 24 32 40 48 56 64 72
9 18 27 36 45 54 63 72 81
OK
```

Remarque : il faut commencer par regarder la boucle B. Lorsqu'elle est terminée, l'instruction PRINT qui suit NEXT B fait aller à la ligne. Puis on reprend la boucle B. Cette série d'instructions imprime neuf fois (boucle A) une série de neuf nombres (boucle B).

L'alignement n'est pas parfait dans ce tableau à cause de la présence de nombres de 1 ou 2 chiffres. Il faut s'en accommoder pour l'instant.

Pour les jeux d'abord, mais aussi pour les applications plus sérieuses, il est important que le MO6 puisse faire certaines manipulations au hasard. Voilà pourtant un rôle pour lequel un micro-ordinateur semble mal disposé. Nous sommes bien habitués maintenant à ce que la machine ne laisse rien au hasard. Il faut donc s'attendre à ce que le hasard produit soit d'un type particulier, une sorte de faux hasard, une simulation.

C'est bien de cela dont il va s'agir : un hasard calculé. Mais si bien calculé qu'il ressemblera comme un frère à un vrai hasard, imprévisible. La fonction RND (de l'anglais *random*) donne un nombre décimal aléatoire entre 0 et 1.

```
PRINT RND : PRINT RND
.594972
.512679
OK
```

Ces nombres sont calculés par BASIC et se répartissent uniformément entre 0 et 1.

Applications

RND donne un nombre décimal entre 0 et 1. Grâce à quelques manipulations arithmétiques simples, on peut tirer des nombres de types variés dans des intervalles variés.

Décimaux

Entre 0 et X : $RND \cdot X$
Entre A et B : $A + RND \cdot (B - A)$

Entiers

Entre 0 et N : $INT(RND \cdot (N + 1))$
Entre P et Q : $P + INT(RND \cdot (Q - P + 1))$

Remarque : la fonction INT introduite ici donne le plus grand entier inférieur au nombre mis entre parenthèses.

```
PRINT INT(256/7)
36
OK
```

car $256/7 = 36,571428\dots$

Exemples

```
X = INT(RND * 2)
PRINT "PILE : 1, FACE : 0" : PRINT X
PILE : 1, FACE : 0
1
OK
```

C'est donc pile !

```
X = 1 + INT(RND * 6)
PRINT "FACE DU DE : " : X
FACE DU DE : 4
OK
```

```
X = 1 + INT(RND * 49)
PRINT "OTO GAGNANT : " : X
OTO GAGNANT : 22
OK
```

Pensez-vous vraiment que votre MO6 soit capable de vous faire gagner le gros lot ?

Ciel étoilé

```
SCREEN 3,0 G : FOR K = 1 TO 200 : X = INT(RND * 320) :
Y = INT(RND * 200) : PSET(X,Y) : NEXT K
```

Remarque : cette commande allume 200 points dont les coordonnées sont choisies au hasard dans un rectangle de 320 par 200 ; c'est donc qu'ils remplissent l'écran.

Promenade d'une particule

```
CLIQUE = 1000000:RANDOM(1,100)
X=0:Y=0:Z=0:ANGLE=RND(1)
FOR I=0 TO 1000000
  X=X+CLIQUE*Z*ABS(COS(ANGLE))
  Y=Y+CLIQUE*Z*ABS(SIN(ANGLE))
  Z=Z*ABS(COS(ANGLE))
NEXT I
```

Attention, ne validez la ligne qu'après NEXT I.

A l'exécution, on voit la trace d'une particule qui s'agite dans tous les sens.



Malheureusement, elle a tendance à s'échapper de l'écran, à y revenir, puis à en repartir.

On peut également utiliser les instructions de déplacement d'une tortue (voir module 13). Seul l'angle est choisi au hasard entre 0 et 256. Chaque déplacement élémentaire a pour longueur 20.

```
CLIQUE = 1000000:RANDOM(1,100)
X=0:Y=0:Z=0:ANGLE=RND(256)
FOR I=0 TO 1000000
  X=X+CLIQUE*Z*ABS(COS(ANGLE))
  Y=Y+CLIQUE*Z*ABS(SIN(ANGLE))
  Z=Z*ABS(COS(ANGLE))
NEXT I
```

Une mauvaise surprise

Petite expérience : faites imprimer quelques nombres tirés au hasard par la fonction RND et notez-les, puis éteignez votre MO6.

Vous aurez la mauvaise surprise en recommençant la même manœuvre de constater que les nombres qui vous sont servis sont strictement les mêmes. Drôle de hasard !

En fait, comme nous l'avons dit plus haut, ces nombres aléatoires sont calculés à partir d'une même formule qui donne toujours les mêmes résultats. Mais comme chaque RND est réutilisé dans la formule pour le calcul du suivant, la suite produite est toujours la même. Pour obtenir des suites différentes, il suffit d'initialiser la suite à partir de nombres différents.

Nous découvrirons au chapitre 29 comment on peut simuler du "vrai hasard" avec votre MO6.

