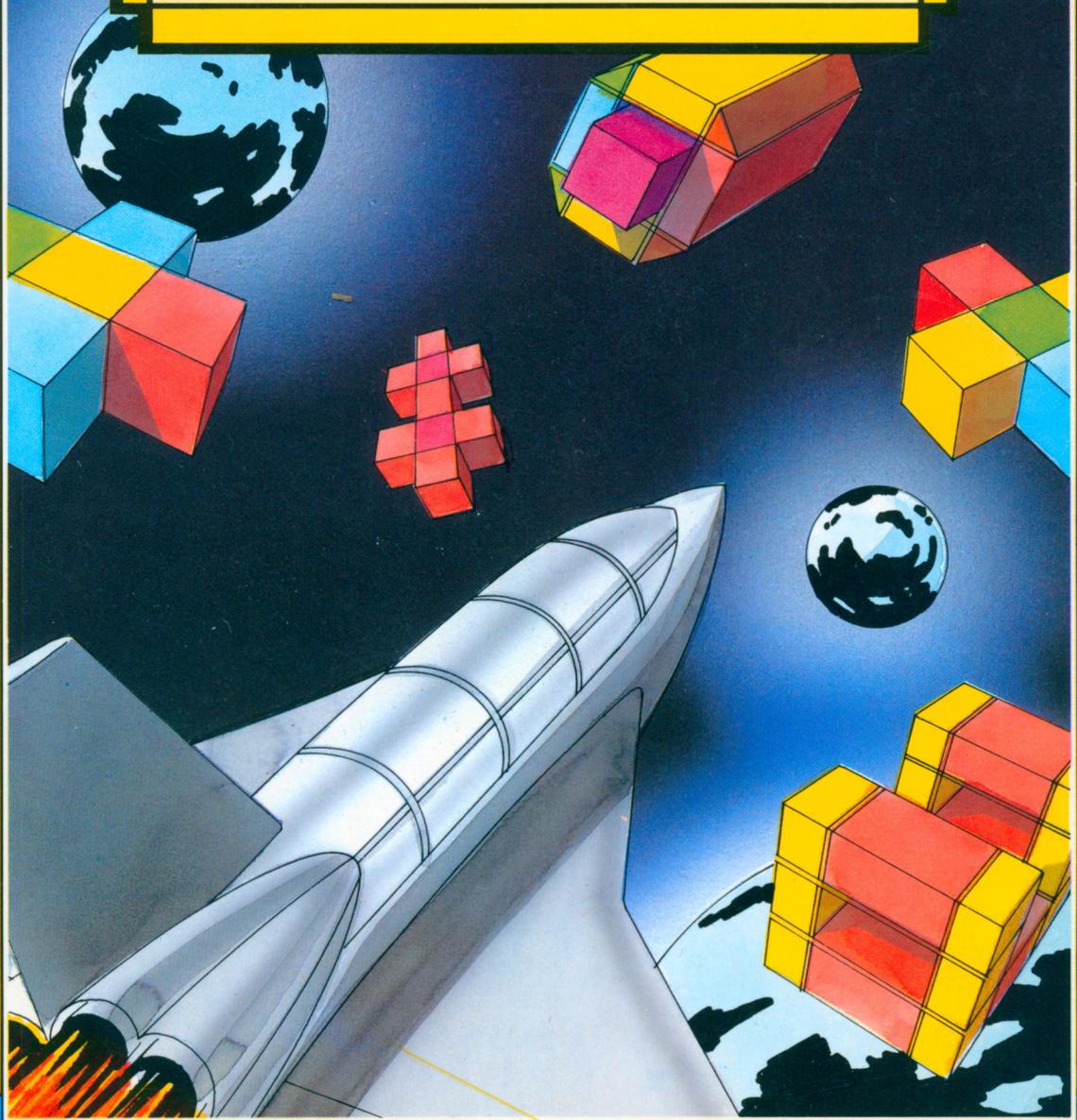


RENDEZ-VOUS **MICRO**

VOYAGE EN BASIC

Navette spatiale, sur M05 et AMSTRAD.



 FOUCHER

Ce livre est une traduction de l'ouvrage
de Gary Marshall publié en Grande-Bretagne
par Aladdin Books Ltd sous le titre
Beginning BASIC: Space journey
dans la collection « write your own program »

An Aladdin book designed and produced by
Aladdin Books Ltd
70 Old Compton Street
London W1

Traduction : INFOPUCE
78120 GREFFIERS

RENDEZ-VOUS MICRO

Gary Marshall

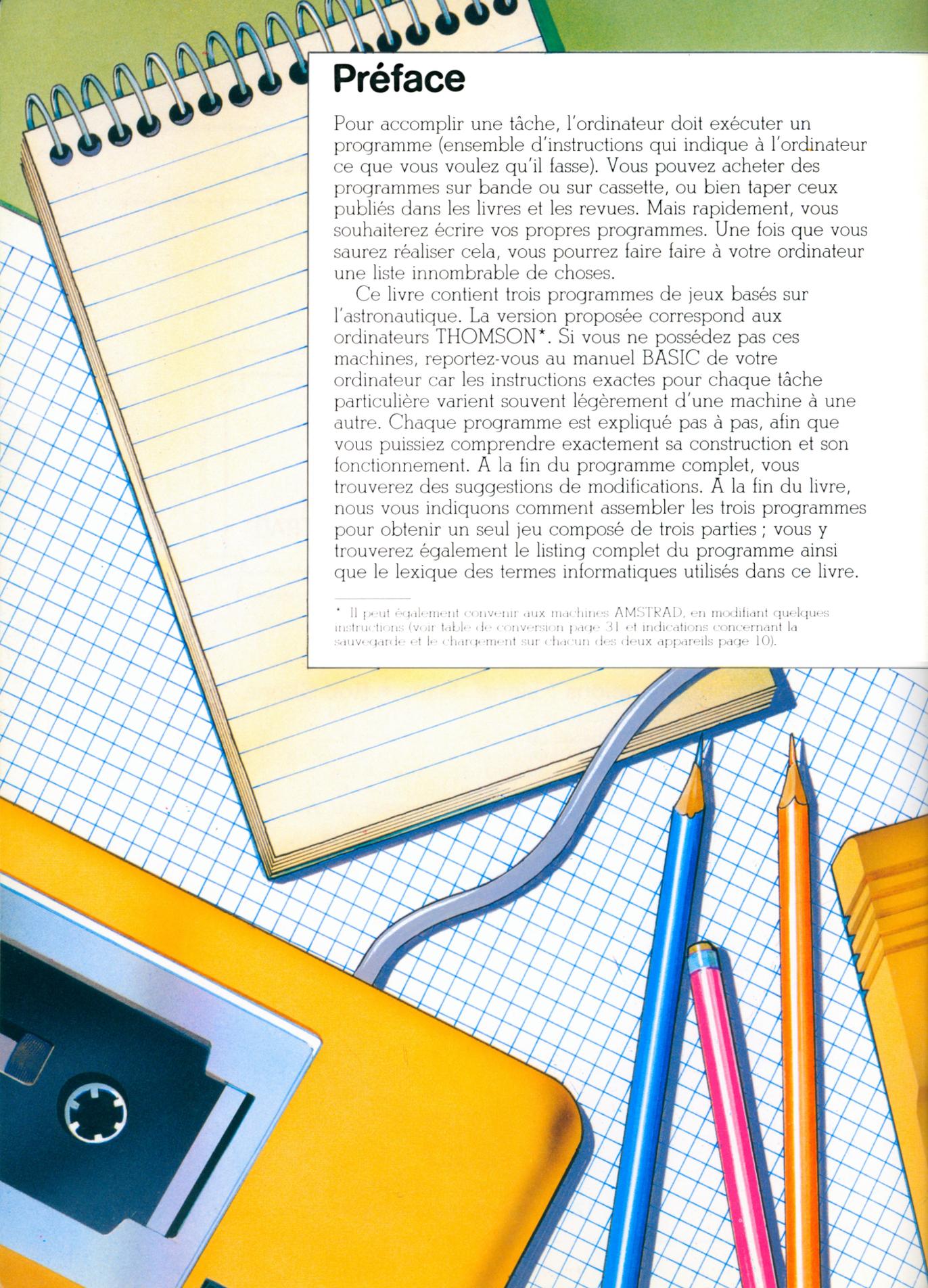
VOYAGE EN BASIC

Navette spatiale, sur M05 et AMSTRAD.

Illustrations : Lionel Jeans • Rob Shone



LES ÉDITIONS FOUCHER
128 rue de Rivoli 75001 Paris

The background of the page is a colorful illustration of school supplies. In the top left, there is a spiral-bound notebook with a yellow cover and lined pages. Below it is a book with a blue cover and a white spine. In the bottom left, a portion of a yellow pencil sharpener is visible. On the right side, three pencils are shown: a blue one, a pink one, and an orange one. The entire scene is set against a light blue grid pattern.

Préface

Pour accomplir une tâche, l'ordinateur doit exécuter un programme (ensemble d'instructions qui indique à l'ordinateur ce que vous voulez qu'il fasse). Vous pouvez acheter des programmes sur bande ou sur cassette, ou bien taper ceux publiés dans les livres et les revues. Mais rapidement, vous souhaitez écrire vos propres programmes. Une fois que vous saurez réaliser cela, vous pourrez faire faire à votre ordinateur une liste innombrable de choses.

Ce livre contient trois programmes de jeux basés sur l'astronautique. La version proposée correspond aux ordinateurs THOMSON*. Si vous ne possédez pas ces machines, reportez-vous au manuel BASIC de votre ordinateur car les instructions exactes pour chaque tâche particulière varient souvent légèrement d'une machine à une autre. Chaque programme est expliqué pas à pas, afin que vous puissiez comprendre exactement sa construction et son fonctionnement. À la fin du programme complet, vous trouverez des suggestions de modifications. À la fin du livre, nous vous indiquons comment assembler les trois programmes pour obtenir un seul jeu composé de trois parties ; vous y trouverez également le listing complet du programme ainsi que le lexique des termes informatiques utilisés dans ce livre.

* Il peut également convenir aux machines AMSTRAD, en modifiant quelques instructions (voir table de conversion page 31 et indications concernant la sauvegarde et le chargement sur chacun des deux appareils page 10).

Sommaire

Introduction au BASIC	6
Faites connaissance avec votre ordinateur	7
Préparation d'un programme	8
Organigramme	9
Sauvegarde des programmes	10
Programme 1 LANCEMENT DE LA NAVETTE	11
Amélioration du programme	15
Programme 2 ASTEROÏDES	16
Mise en couleurs	22
Programme 3 ATERRISSAGE DE LA NAVETTE	23
Amélioration et chaînage des programmes	27
Listing du programme	27
Glossaire	29
Table de conversion	31
Index	31

```
20 LOCATE 10,4:PRINT "LANCEMENT DE LA NAVETTE"  
30 LOCATE 10,5:PRINT "*****"  
40 LOCATE 8,8:PRINT "NAVETTE PRETE AU DECOLLAGE"  
50 LOCATE 11,10:PRINT "VOUS DEVEZ LA LANCER!"  
60 LOCATE 8,13:PRINT "VOUS N'AVEZ QUE DIX ESSAIS"  
70 LOCATE 8,16:PRINT "TAPEZ N'AVEZ MAINTENANT VOTRE NOM"  
80 LOCATE 9,18:PRINT "ET APPUYEZ SUR 'ENTREE' :"  
90 LOCATE 15,20:INPUT " ",N$  
100 CLS  
110 PRINT "BIENVENUE A BORD CAPITAINE ";N$  
120 LOCATE 1,4:PRINT "POUR LANCER LA NAVETTE, VOUS  
DEVEZ"  
130 LOCATE 1,6:PRINT "DEVINEZ LE NIVEAU DE PUISSAN  
CE REQUIS"  
140 LOCATE 1,8:PRINT "(N'IMPOR  
1 ET 100)"  
150 LOCATE
```

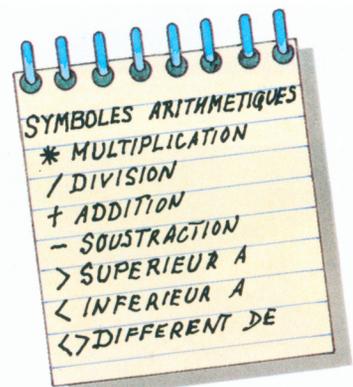
Introduction au BASIC

La plupart des programmes destinés aux ordinateurs domestiques sont écrits dans un langage informatique appelé BASIC. Comme tout autre langage, le BASIC comporte des règles à suivre impérativement pour que le programme fonctionne. Par exemple, l'instruction BASIC **PRINT** provoque l'affichage d'informations à l'écran, mais tout le texte doit être mis entre guillemets, comme on peut le voir dans la première ligne du programme ci-dessous. Tapez les autres lignes et observez ce qui se passe. Vous remarquerez que les lignes 2 et 3 produisent des résultats très différents : en ligne 2, "15*2" est considéré comme un message tandis qu'en ligne 3, il s'agit d'un calcul (absence de guillemets). Les symboles arithmétiques utilisés en BASIC sont répertoriés sur le bloc-notes ci-contre.

```
PRINT "BON ANNIVERSAIRE"  
PRINT "15*2"  
PRINT 15*2
```

L'ordinateur exécute les instructions d'un programme dans un ordre logique. Aussi, un programme ne fonctionnerait pas si l'ordinateur effectuait chaque instruction dès que celle-ci est tapée. En attribuant à chaque instruction un numéro de ligne, nous demandons à l'ordinateur de l'enregistrer dans sa mémoire jusqu'à l'exécution du programme demandée par l'instruction **RUN**. Les numéros de ligne sont généralement attribués de 10 en 10, les trois premières lignes du programme portant souvent les numéros 10, 20 et 30. L'ordinateur exécute les instructions en respectant strictement leur ordre numérique. Des instructions supplémentaires peuvent être insérées ultérieurement en leur donnant des numéros de lignes intermédiaires, tels que 15, 25, etc. Exercez-vous à taper le petit programme ci-dessous.

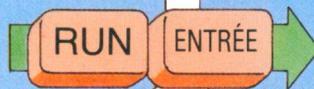
Notez que la ligne 10 ne produit rien à l'écran. L'instruction **REM** (signifiant *Remarque*) indique à l'ordinateur d'ignorer la fin de la ligne. La ligne 10 ne joue qu'un rôle de « note » écrite par le programmeur pour sa propre information.



```
10 REM VOEUX D'ANNIVERSAIRE  
20 PRINT "BON ANNIVERSAIRE"  
30 PRINT "BON ANNIVERSAIRE"  
40 PRINT "BON ANNIVERSAIRE CHER ORDINATEUR"  
50 PRINT "BON ANNIVERSAIRE"  
60 PRINT  
70 PRINT "VOUS AVEZ"  
80 PRINT "1"  
90 PRINT "AN"
```

RUN (ENTREE)

```
BON ANNIVERSAIRE  
BON ANNIVERSAIRE  
BON ANNIVERSAIRE CHER ORDINATEUR  
BON ANNIVERSAIRE  
VOUS AVEZ  
1  
AN
```



Faites connaissance avec votre ordinateur

Avant de commencer à travailler sur un programme, familiarisez-vous avec le clavier de votre ordinateur. Ce que vous tapez n'a pas d'importance : vous ne pouvez pas endommager votre machine. Le plus souvent, elle affichera comme réponse "Syntax error".

Apprenez tout particulièrement à jouer avec les touches d'édition pour vous permettre de corriger toutes les erreurs que vous pourriez faire. Le mode d'emploi qui accompagne votre ordinateur vous aidera à les utiliser. Ne vous inquiétez pas si vous ne comprenez pas certaines fonctions. Vous découvrirez leur rôle lorsque vous en aurez besoin. Voici une liste des touches ou des fonctions les plus importantes utilisées sur les ordinateurs THOMSON.

N.B. – Les touches indiquées en bleu doivent être tapées.

**DEL
INS**

La touche **DEL** efface le caractère placé sous le curseur. Les touches fléchées déplacent le curseur. La touche **INS** permet d'insérer des caractères dans une ligne.

DELETE

L'instruction **DELETE** permet d'effacer un groupe de lignes. Par exemple **DELETE 10-50** efface toutes les lignes depuis la ligne 10 incluse jusqu'à la ligne 50 incluse. Pour effacer une seule ligne, tapez son numéro, immédiatement suivi par **ENTREE**.

LIST

Si vous tapez **LIST** puis **ENTREE**, vous verrez défiler le programme à l'écran.

ENTREE

Une action sur cette touche indique à l'ordinateur que vous avez terminé une phrase (ou *ligne de programme*) ; vous lui demandez de la ranger dans sa mémoire en vue d'une exécution ultérieure ou de l'exécuter immédiatement (dans ce cas, la ligne ne commence pas par un numéro).

STOP

Permet d'arrêter provisoirement le défilement du listing à l'écran (lancé par l'instruction **LIST**). Le fait d'appuyer sur une touche quelconque relance le défilement, alors que l'appui simultané des touches **CNT** et **C** arrête définitivement le listing.

RAZ

Cette touche efface l'écran et ne modifie pas le programme. Pour provoquer le même effet dans le programme, il faut taper l'instruction « **CLS** ».

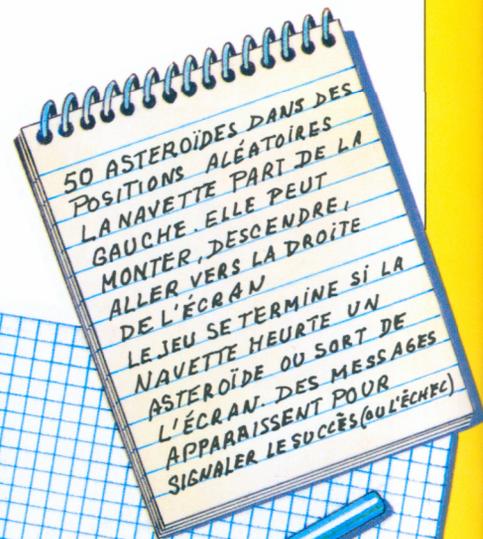
NEW

Cette instruction efface toutes les lignes d'un coup. La mémoire de l'ordinateur est donc vidée. **ATTENTION** : tout ce que vous aviez tapé est perdu. Sauvegardez sur cassette ou sur disquette votre programme avant d'employer **NEW** !

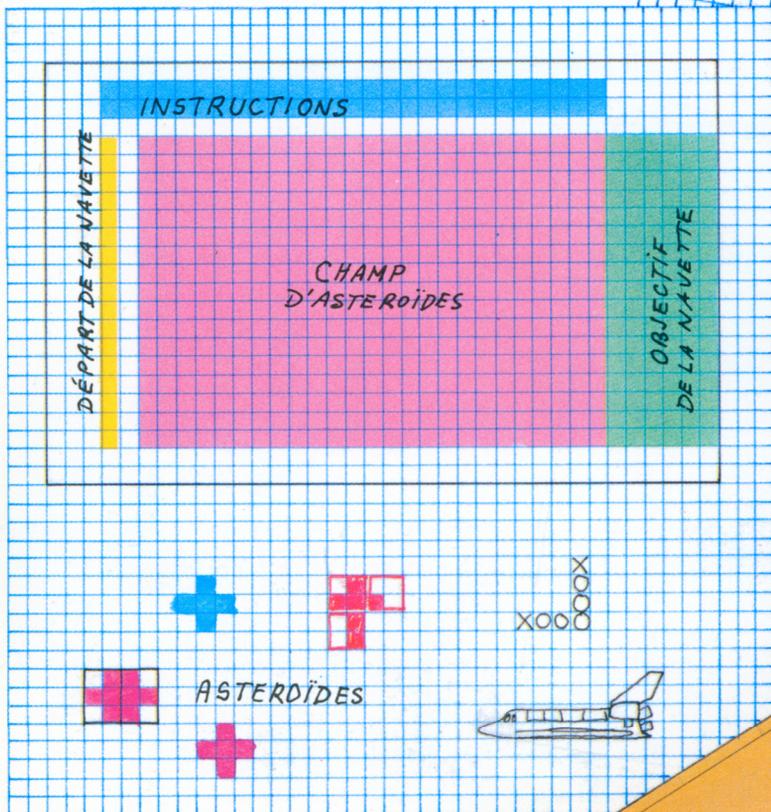
Préparation d'un programme

Avant de commencer à écrire un programme en BASIC, vous devez déterminer avec exactitude ce que vous voulez faire réaliser par l'ordinateur. La première étape consiste à écrire les grandes lignes du programme. Voici ce que cela donne pour le premier programme de ce livre : « Ce jeu consiste à deviner un nombre tiré au hasard. Si la solution proposée est juste, l'ordinateur affiche à l'écran une navette au décollage. Pas plus de dix essais. Les instructions du jeu doivent figurer clairement à l'écran. »

Vous pouvez également définir l'aspect de l'écran de telle façon que les instructions puissent être clairement présentées. Pour cela, utilisez une feuille de papier quadrillé, car l'écran de l'ordinateur est aussi composé de cases repérées par leurs coordonnées verticales et horizontales (le nombre de ces cases varie en fonction de la machine). Le projet de base correspondant au second jeu de ce livre est indiqué sur le bloc-note ci-contre : vous devez piloter une navette à travers un champ d'astéroïdes. Un projet d'écran initial est également élaboré. Lorsque ces préparatifs sont terminés, vous pouvez passer à la seconde étape : dresser un organigramme. Celui-ci vous permettra de diviser votre programme en étapes logiques.

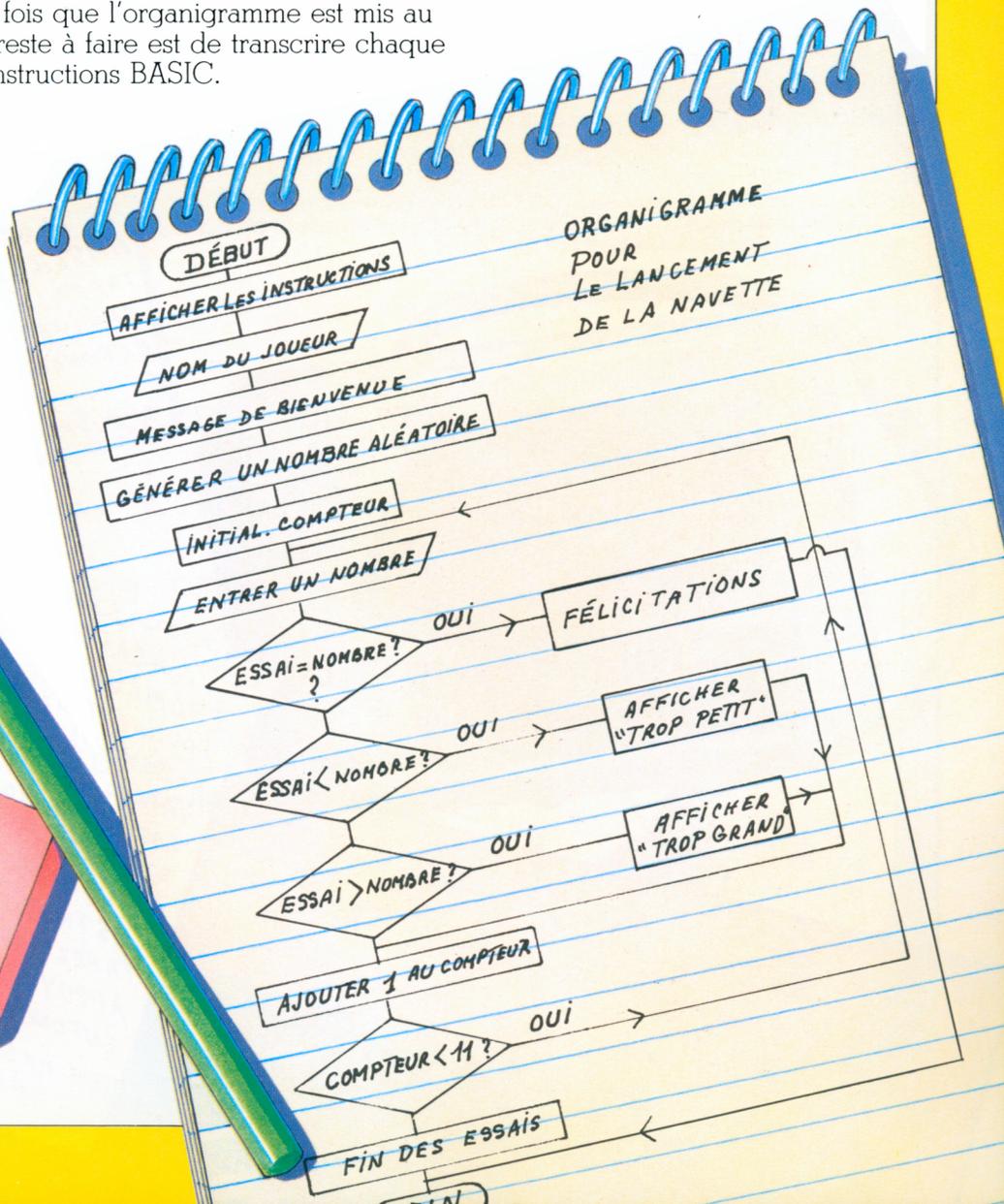


50 ASTÉROÏDES DANS DES POSITIONS ALÉATOIRES. LA NAVETTE PART DE LA GAUCHE. ELLE PEUT MONTER, DESCENDRE, ALLER VERS LA DROITE DE L'ÉCRAN. LE JEU SE TERMINE SI LA NAVETTE HEURTE UN ASTÉROÏDE OU SORT DE L'ÉCRAN. DES MESSAGES APPARAÎSSENT POUR SIGNALER LE SUCCÈS (OU L'ÉCHEC).



Organigramme

Un organigramme sert à éclater le problème de départ en blocs d'instructions plus petits. L'organigramme ci-dessous correspond au premier programme du livre. Les losanges désignent les points de prises de décision. Par exemple : « La solution proposée est-elle trop grande ? » ou « Avez-vous utilisé vos dix essais ? ». Les réponses ne peuvent être que OUI ou NON. En fonction de la réponse, l'ordinateur empruntera l'une ou l'autre voie. Les rectangles sont utilisés pour indiquer les opérations à effectuer, comme par exemple l'affichage de l'écran de départ. Les organigrammes vous aident à vous poser les bonnes questions ; vous pouvez donner ensuite les instructions appropriées à l'ordinateur. En préparant avec soin l'organigramme, vous serez certain de n'avoir oublié aucune étape. Une fois que l'organigramme est mis au point, tout ce qu'il reste à faire est de transcrire chaque partie en lignes d'instructions BASIC.



Sauvegarde des programmes

Même si vous connaissez très bien un programme, vous passerez beaucoup de temps à le taper sur le clavier de votre ordinateur, ligne après ligne, à chaque fois que vous souhaitez l'utiliser. En outre, il est possible que vous tapiez des erreurs, ce qui vous oblige à le vérifier avec soin et à le corriger (ou « *débuger* » en jargon informatique). En conséquence, une fois que votre programme fonctionne, il est préférable de l'enregistrer pour pouvoir facilement le ré-utiliser plus tard. La procédure d'enregistrement est simple. Les instructions pour enregistrer (sauvegarder) ou lire (charger) un programme sont regroupées sur le bloc-note ci-contre. Il est prudent de sauvegarder deux versions de votre programme avant d'éteindre votre ordinateur ; cela évite les mauvaises surprises si le premier enregistrement a mal fonctionné, comme cela arrive parfois. Étiquetez votre cassette ou votre disquette en notant le nom du programme. Avec des cassettes ou des disquettes correctement étiquetées, vous pourrez retrouver facilement votre programme lorsque vous voudrez l'utiliser à nouveau.

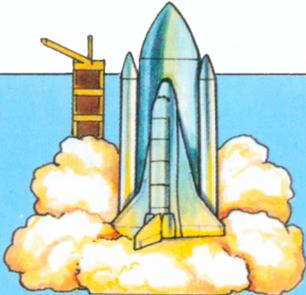
(1) Opération inutile sur les ordinateurs THOMSON utilisés avec disquettes.

(2) Opérations inutiles sur CPC 664 et CPC 6128 car le chargement s'effectue par défaut sur disquette et non sur cassette.



SAUVEGARDE
THOMSON
TOUCHE "REC" (●) ET "PLAY" (▶)
DU MAGNETOCASSETTE (1)
TAPEZ : SAVE "CASS: NOM" (VERSION
CASSETTE) OU SAVE "NOM" (VERSION
DISQUETTE).
APPUYEZ SUR "ENTRÉE".
AMSTRAD
TAPEZ : SAVE "NOM"
APPUYEZ SUR "ENTER"
ENFONCEZ LES TOUCHES "REC"
ET "PLAY" DU MAGNETOCASSETTE
APPUYEZ SUR "ENTER" (2)

CHARGEMENT
THOMSON
TOUCHE "PLAY" (▶) DU MAGNETO.
CASSETTE (1)
TAPEZ : LOAD "CASS: NOM" (VERSION
CASSETTE) OU LOAD "NOM" (VERSION
DISQUETTE).
APPUYEZ SUR "ENTRÉE"
AMSTRAD
TAPEZ : LOAD "NOM"
APPUYEZ SUR "ENTER"
ENFONCEZ LA TOUCHE "PLAY"
DU MAGNETOCASSETTE (2)
APPUYEZ SUR "ENTER" (2)



LANCEMENT DE LA NAVETTE

Vous êtes le commandant de la navette spatiale. Pour lancer la navette, vous devez introduire dans l'ordinateur central le niveau de puissance nécessaire. Vous n'avez que dix essais pour trouver la bonne valeur sinon votre mission est terminée. L'ordinateur vous guidera en vous indiquant si la valeur tapée est trop grande ou trop petite. Si vous réussissez, le tableau de bord de la navette vous indiquera votre altitude jusqu'à ce que vous atteigniez l'orbite finale. Bonne chance !

LANCEMENT DE LA NAVETTE

NAVETTE PRETE AU DECOLLAGE

VOUS DEVEZ LA LANCER !

VOUS N'AVEZ QUE DIX ESSAIS

TAPEZ MAINTENANT VOTRE NOM ET APPUYEZ
SUR 'ENTREE' :

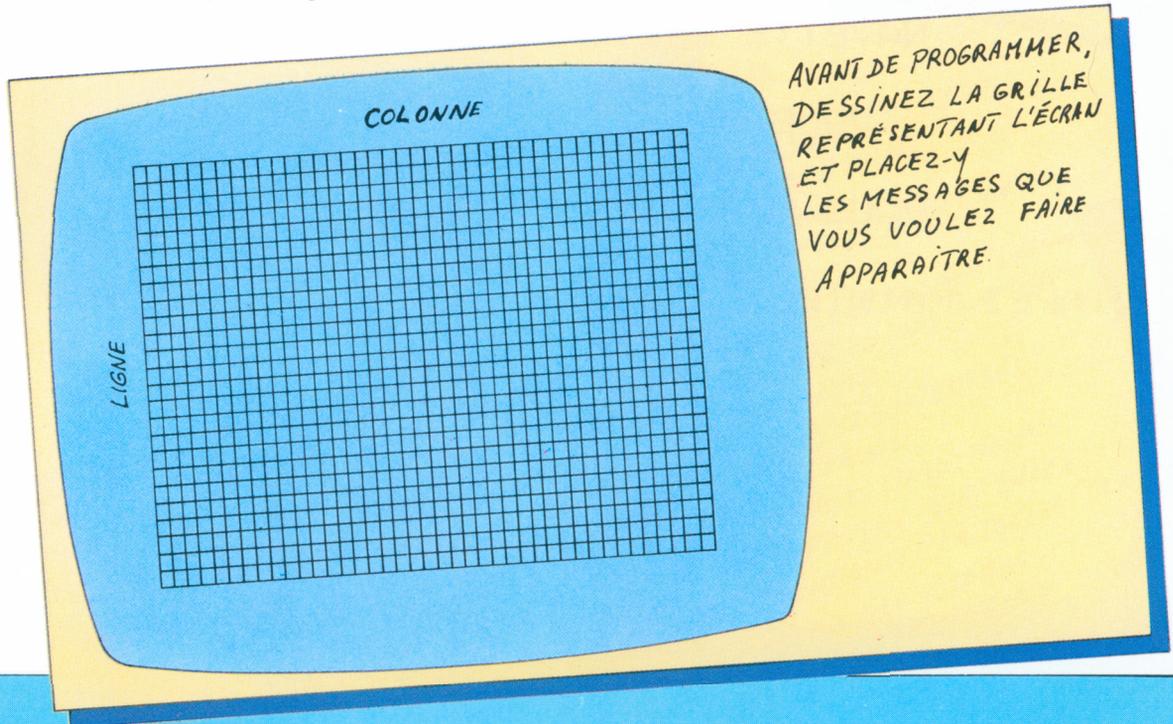
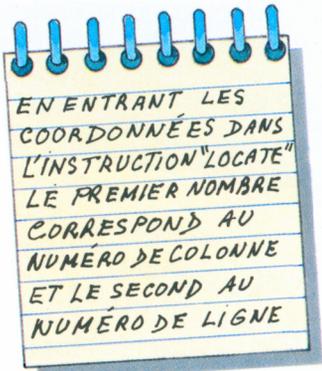
Ce programme utilise la possibilité qu'a l'ordinateur de générer des nombres aléatoires. La plupart des lignes de ce programme constituent des messages destinés au joueur. Ils apparaîtront au moment opportun dans le cours du jeu. Les neuf premières lignes du programme constituent l'écran de départ.

```

5 REM lancement de la navette
10 CLS: CLEAR 1000, ,5: SCREEN 6,0,0
11 DEFGR$(0)=24,60,102,195,195,102,60,24
12 DEFGR$(1)=255,255,255,255,255,255,255,255
13 DEFGR$(2)=192,96,48,255,255,48,96,192
14 DEFGR$(3)=192,240,252,255,255,252,240,192
20 LOCATE 10,4: PRINT "LANCEMENT DE LA NAVETTE"
30 LOCATE 10,5: PRINT "*****"
40 LOCATE 8,8: PRINT "NAVETTE PRETE AU DECOLLAGE"
50 LOCATE 11,10: PRINT "VOUS DEVEZ LA LANCER!"
60 LOCATE 8,13: PRINT "VOUS N'AVEZ QUE DIX ESSAIS"
70 LOCATE 8,16: PRINT "TAPEZ MAINTENANT VOTRE NOM"
80 LOCATE 9,18: PRINT "ET APPUYEZ SUR 'ENTREE' : "

```

La ligne 5 rappelle le titre du programme en utilisant l'instruction **REM**. Vous retrouverez ainsi facilement dans quelques mois à quoi servait ce programme, simplement en relisant cette ligne. L'écran est effacé par l'instruction **CLS** placée en ligne 10. **CLEAR** réserve de la place en mémoire pour les chaînes de caractères (1 000 octets) et pour les caractères graphiques redéfinis (5 caractères) dont nous aurons besoin plus loin dans le programme. Les instructions **DEFGR\$()** déterminent la forme de chacun de ces caractères. Pour afficher les messages, nous aurions pu utiliser directement l'instruction **PRINT** mais l'écran n'aurait pas été bien attrayant. En conséquence, nous commençons par utiliser l'instruction **LOCATE** qui permet de disposer agréablement le texte à l'écran. Les nombres qui suivent cette instruction représentent les coordonnées horizontales et verticales du début de la ligne de texte. Nous pouvons utiliser jusqu'à 40 caractères par ligne et 25 caractères par colonne. Ainsi la ligne 20 produira l'affichage du titre sur la quatrième ligne, à partir de la dixième colonne. Les autres messages sont positionnés de la même manière.



```

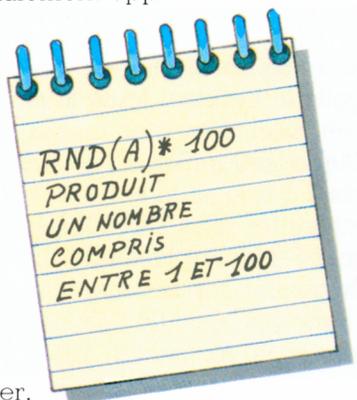
90 LOCATE 15,20:INPUT " ",N$
100 CLS
110 PRINT "BIENVENUE A BORD CAPITAINE ";N$
120 LOCATE 1,4:PRINT "POUR LANCER LA NAVETTE, VOUS DEVEZ"
130 LOCATE 1,6:PRINT "DEVINEZ LE NIVEAU DE PUISSANCE REQUIS"
140 LOCATE 1,8:PRINT "(N'IMPORTE QUEL NOMBRE ENTRE 1 ET 100)"
150 LOCATE 4,12:PRINT "TAPEZ-LE ET APPUYEZ SUR 'ENTREE' : "
160 LOCATE 4,14:PRINT "ALLEZ-Y !"
    
```

En ligne 90, le programme attend que le joueur tape son nom. L'instruction **INPUT** point d'interrogation pour indiquer à l'utilisateur que fait normalement apparaître un l'ordinateur attend une réponse au clavier. Toutefois, dans notre programme, nous avons placé des guillemets et une virgule entre **INPUT** et **N\$**. Cela fait disparaître le point d'interrogation, et le texte placé entre les guillemets (ici, un espace seulement) est affiché à la place.

Après l'instruction **INPUT**, l'écran est effacé et la série d'instructions comprise entre les lignes 110 et 160 compose le premier écran de jeu. Tout ce que vous taperez ensuite sera placé dans la variable chaîne de caractères appelée **N\$**. Le signe **\$** signale à l'ordinateur qu'il peut accepter des lettres et/ou des chiffres.

Sans ce signe, l'ordinateur n'accepterait que des chiffres. Lorsqu'on lui demandera d'afficher **N\$** à la ligne 110 (**PRINT N\$**), l'ordinateur affichera ce qui aura été tapé au clavier.

Notez qu'à la ligne 110, **N\$** n'est pas entre guillemets et est précédé d'un point virgule.



```

170 NIVEAU=INT((RND(1))*100)+1
180 FOR G=1 TO 10
185 LOCATE 4,16:PRINT "ESSAI";G
190 LOCATE 4,18:INPUT ESSAI
200 IF ESSAI=NIVEAU THEN GOTO 280
210 IF ESSAI<NIVEAU THEN LOCATE 8,20:PRINT "TROP BAS.
    ESSAYEZ ENCORE. "
220 IF ESSAI>NIVEAU THEN LOCATE 8,20:PRINT "TROP HAUT.
    ESSAYEZ ENCORE. "
230 NEXT G
240 CLS
250 LOCATE 1,4:PRINT "VOUS N'AVEZ PAS TROUVE APRES 10
    ESSAIS. "
260 LOCATE 1,6:PRINT "TAPEZ 'RUN' (ENTREE) POUR RECOM
    MENCER. "
270 STOP
    
```

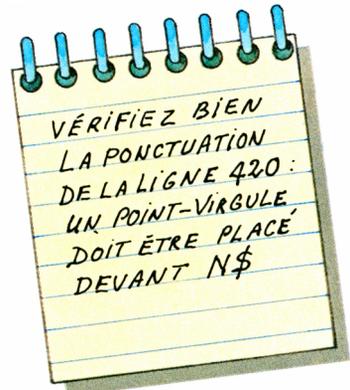
La ligne 170 produit un nombre aléatoire compris entre 1 et 100 grâce à la commande **RND(1)**. Le nombre que vous devrez deviner est enregistré par l'ordinateur dans la variable **NIVEAU**.

La ligne 180 comptabilise les essais du joueur en limitant leur nombre à 10. En ligne 200, le programme compare votre réponse au résultat attendu. Si l'égalité est vérifiée, il passe à la ligne 280. Si le nombre exact n'a pas été trouvé, les lignes 210 et 220 donnent une indication au joueur. Après dix essais, le programme passe à la ligne 240, les messages correspondants sont affichés et le jeu s'arrête.

```

○ 280 CLS
○ 290 HAUT=0
○ 300 CIBLE=10000
○ 310 LOCATE 10,2:PRINT "DECOLLAGE..."
○ 320 LOCATE 10,10:PRINT "ALTITUDE : ";HAUT
○ 340 FOR D=1 TO 100
○ 350 NEXT D
○ 360 HAUT=HAUT*2+1
○ 365 PLAY"DO"
○ 370 IF HAUT<CIBLE THEN GOTO 320
    
```

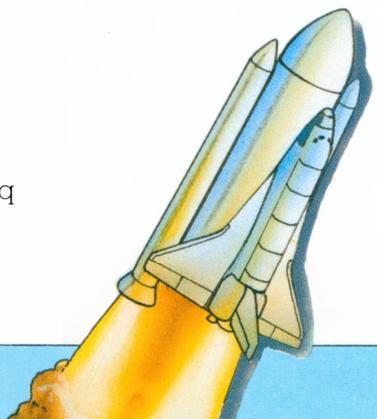
L'écran se transforme en tableau de bord. La navette décolle et pendant toute la phase ascensionnelle, l'altitude est affichée jusqu'à ce que la navette atteigne son orbite. La ligne 290 initialise l'altitude à zéro, et la ligne 300 situe l'orbite à une altitude de 10 000. Les trois lignes suivantes affichent les mots DECOLLAGE et ALTITUDE ainsi que le nombre donnant l'altitude. Les instructions **LOCATE** positionnent correctement les différents messages à l'écran. La ligne 360 calcule l'altitude en doublant l'ancienne valeur et en ajoutant 1. L'ordinateur est si rapide que les nombres défileraient trop vite à l'écran et ne pourraient être lus. Aussi est-il nécessaire de ralentir l'ordinateur en ajoutant une boucle de temporisation aux lignes 340 et 350. Ces instructions demandent à l'ordinateur de compter jusqu'à 1 000 avant d'exécuter autre chose. Pour obtenir un affichage plus rapide, placez un nombre inférieur à 1 000 après l'instruction TO de la ligne 340. La ligne 370 ordonne à l'ordinateur de retourner à la ligne 320 tant que l'orbite n'est pas atteinte.

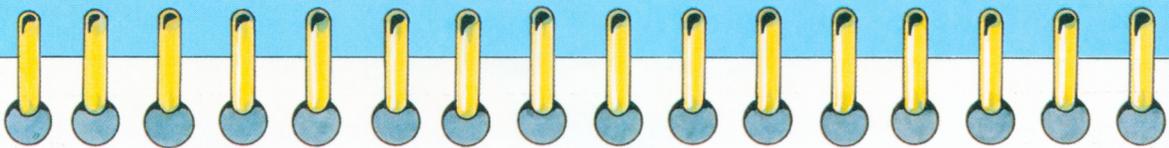


```

○ 380 CLS
○ 390 FOR R=5 TO 10
○ 400 LOCATE 10,R:PRINT "NAVETTE EN ORBITE!"
○ 410 NEXT R
○ 420 LOCATE 5,1:PRINT "FELICITATIONS CAPITAINE ";N$
    
```

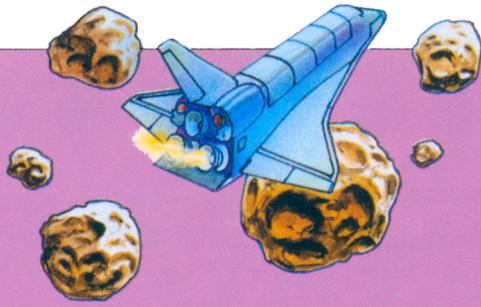
La dernière partie du programme affiche le message de félicitations. La boucle **FOR... NEXT** commençant à la ligne 390 constitue un procédé simple pour afficher cinq fois « NAVETTE EN ORBITE ». La variable **R** sert de compteur de boucles et varie de 5 à 10. L'instruction **LOCATE** utilise cette variable pour décaler l'affichage à chaque fois. Sans cette boucle, nous aurions dû taper cinq fois la ligne 400, ce qui aurait pris plus de place dans la mémoire de l'ordinateur.





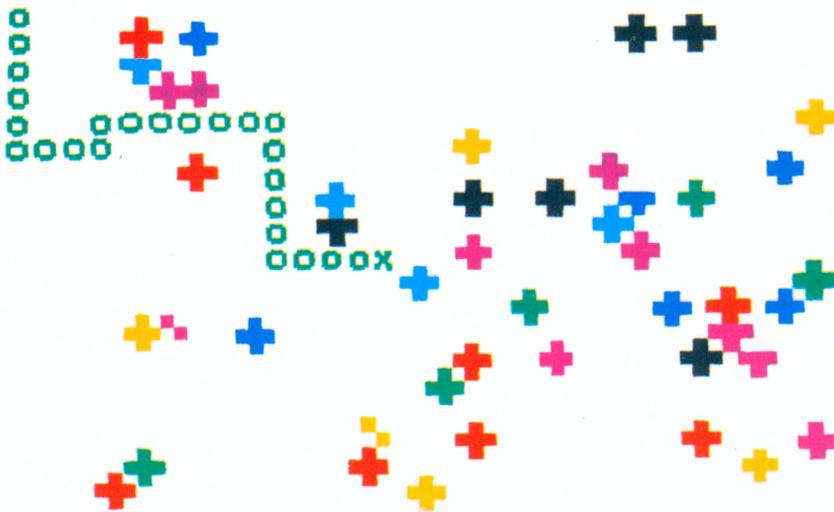
Améliorez votre programme

L'une des meilleures façons d'apprendre à programmer consiste à modifier le programme original après l'avoir tapé. Par exemple, vous pouvez lui ajouter des effets sonores. Sur l'ordinateur THOMSON, l'instruction **PLAY** a été conçue pour cela. Si vous ajoutez des lignes de programme entre les lignes 320 et 340, vous obtiendrez un effet sonore pendant toute la phase du décollage. Reportez-vous au mode d'emploi de votre ordinateur pour comprendre en détail le fonctionnement des différents paramètres de l'instruction **PLAY**.



ASTÉROÏDES

La navette a atteint son orbite avec succès. Soudain, vous pénétrez dans un champ d'astéroïdes. Vous devez donner à l'ordinateur de la navette des consignes précises de navigation afin de pouvoir échapper à une collision qui serait fatale. Il est impossible de reculer.



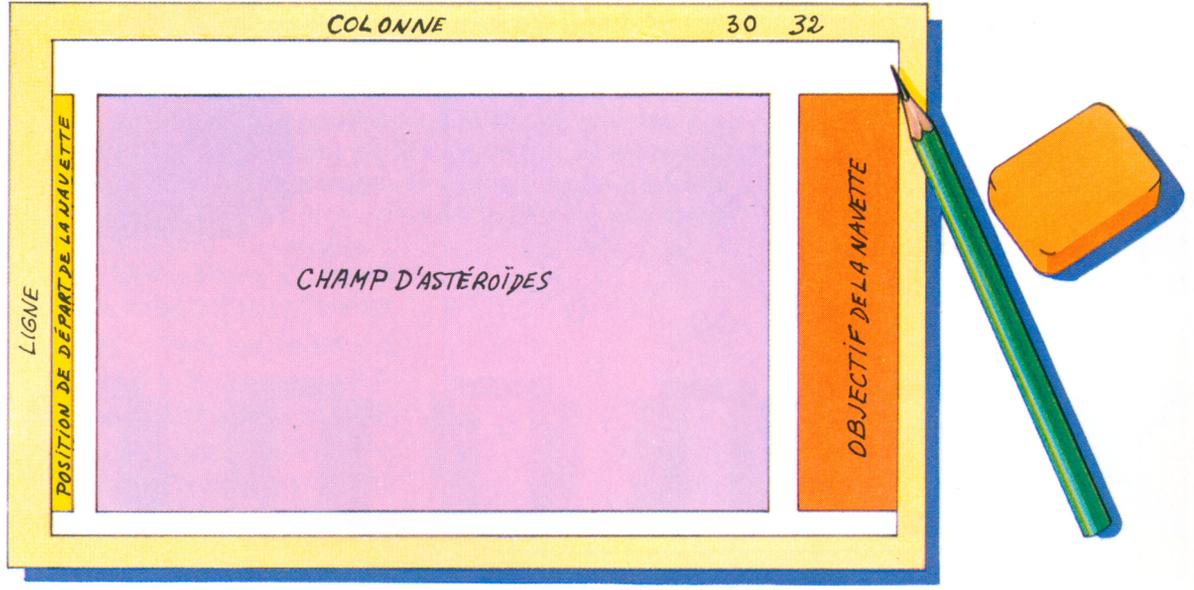
DIRECTION (H, E, D)

Ce programme utilise toujours le texte de l'ordinateur mais exploite les caractères semi-graphiques redéfinis. C'est le rôle des instructions 11 à 15. Le programme affiche les astéroïdes et la navette à l'écran, puis déplace la navette dans la direction voulue par le joueur. Comme précédemment, le programme commence avec une instruction **REM** et efface l'écran.

```
1000 REM ***** ASTEROIDES *****
1010 CLS
```

La première partie du programme affiche les astéroïdes à des positions aléatoires sur l'écran, si bien que ces positions sont différentes à chaque partie.

Avant de placer les astéroïdes, vous devez définir l'aspect de l'écran. En mode texte, vous disposez de 40 colonnes et de 25 lignes. Logiquement, la navette commence son voyage par le côté gauche de l'écran (colonne 0 ou 1 selon l'ordinateur). Le champ d'astéroïdes occupe les colonnes 3 à 30 et la navette atteindra son objectif lorsqu'elle arrivera en colonne 30 sans avoir heurté d'astéroïdes. Les deux lignes supérieures de l'écran sont réservées aux instructions et aux messages devant être donnés par le joueur pendant le déroulement de la partie. En outre, la navette ne doit pas être autorisée à sortir dans le bas de l'écran pour éviter les astéroïdes. Tout cela détermine l'organisation de l'écran ci-dessous. Si vous désirez obtenir un plan différent, vous aurez à modifier quelques nombres dans les instructions **LOCATE**



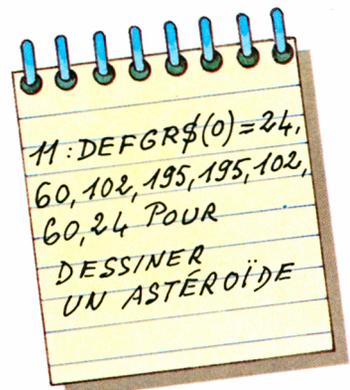
```

1020 DIM A(50),D(50)
1030 FOR S=1 TO 50
1040 A(S)=INT(RND(1)*28)+3
1050 D(S)=INT(RND(1)*18)+3
1055 XXX=INT(RND(1)*4)
1060 LOCATE A(S),D(S):PRINT GR$(0)
1070 NEXT S

```

A la ligne 1020, l'instruction **DIM** réserve de la place pour deux tableaux qui mémoriseront les positions horizontales et verticales des astéroïdes. Le nombre placé entre parenthèses indique le nombre d'astéroïdes : 50 donne un affichage satisfaisant. Les lignes 1040 et 1050 génèrent des positions aléatoires pour les astéroïdes dans les limites de l'écran. A la ligne 1060, l'astéroïde est dessiné en utilisant des caractères semi-graphiques.

La boucle **FOR... NEXT** des lignes 1030 à 1070 affiche ce caractère 50 fois à l'écran, ce qui construit le champ d'astéroïdes.



APRÈS LA LIGNE 1070,
 TAPEZ "RUN" ET LE CHAMP
 D'ASTÉROÏDES DOIT
 APPARAÎTRE A L'ÉCRAN.
 SI VOUS OBTENEZ UN MESSAGE
 D'ERREUR, RELISEZ
 SOIGNEUSEMENT LE LISTING
 CARACTÈRE PAR CARACTÈRE.

Après avoir dessiné les astéroïdes, le programme doit donner une position aléatoire de départ à la navette. L'instruction **RND** est utilisée à nouveau et le nombre généré est entier et compris entre 2 et 20 (ligne 1060). La navette elle-même est représentée par un X. La partie suivante du programme permet au joueur de commander les mouvements de la navette sur l'écran.

```

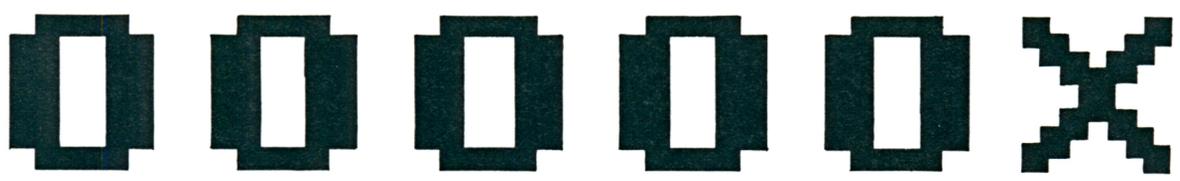
1100 COL=0:LIGNE=2+INT(RND(1)*18)
1110 LOCATE COL,LIGNE:PRINT "X"
    
```

Trois directions (*haut, bas, droite*) suffisent pour commander la navette, celles-ci se déplaçant de la gauche vers la droite. Vous indiquez la direction en tapant au clavier H, B ou D (respectivement haut, bas, droite). Votre commande est mémorisée dans la variable chaîne de caractère **D\$** (instruction **INPUT**). Si une mauvaise touche est enfoncée, la ligne 1200 renvoie le programme à la ligne 1160 jusqu'à ce qu'une touche valide soit utilisée. A ce moment, le programme saute à la ligne 1210.

```

1160 LOCATE 1,1:INPUT "DIRECTION (H, B, D) ",D$
1170 IF D$="H" THEN HAUT=-1:DROITE=0:GOTO 1210
1180 IF D$="B" THEN HAUT=1:DROITE=0:GOTO 1210
1190 IF D$="D" THEN HAUT=0:DROITE=1:GOTO 1210
1200 GOTO 1160
1210 LOCATE 1,1:PRINT SPC(30):LOCATE 1,1:INPUT "DISTANCE ",
DISTANCE
1220 LOCATE COL,LIGNE:PRINT "O"
    
```

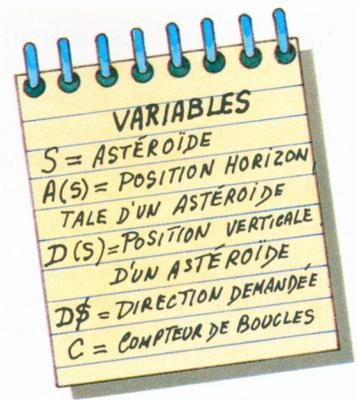
Les lignes 1170 à 1190 attribuent les valeurs correctes aux variables **haut** (= 1 pour descendre, = - 1 pour monter, = 0 s'il n'y a pas de déplacement vertical) et **droite** (= 0 s'il n'y a pas de déplacement, = 1 si déplacement). Le déplacement de la navette s'effectue après que la distance a été tapée (ligne 1210). Notez qu'il n'est pas nécessaire ici d'utiliser une variable chaîne de caractères car une distance ne peut être exprimée qu'avec des chiffres. Pour finir cette partie, le dessin de la navette (un X) est remplacé par un O si bien que la trajectoire de la navette à travers le champ d'astéroïdes et la position de la navette elle-même ne peuvent être confondues.



TRAJECTOIRE DE LA NAVETTE

POSITION DE LA NAVETTE

Les lignes 1230 à 1330 constituent réellement le cœur du programme. Elles déplacent la navette et vérifient qu'aucune collision n'a lieu. Pour cela, nous avons construit une grande boucle **FOR... NEXT** utilisant la variable **C** comme compteur de boucles. La distance introduite à la ligne 210 détermine le nombre de passages dans la boucle. La nouvelle position de la navette est calculée aux lignes 1240 et 1250, en utilisant les valeurs des variables **haut** et **droite** initialisées dans la partie précédente. La navette se déplace d'une case à la fois et la lettre « O » est affichée à la précédente position pour représenter le chemin parcouru.



```
1230 FOR C=1 TO DISTANCE
1240 LIGNE=LIGNE+HAUT
1250 COL=COL+DROITE
1260 FOR S=1 TO 50
1270 IF COL=A(S) AND LIGNE=D(S) THEN GOTO 1370
1290 NEXT S
1300 IF LIGNE>20 OR LIGNE<2 THEN 1390
1310 LOCATE COL,LIGNE:PRINT"O"
1320 IF COL>30 THEN 1410
1330 NEXT C
1340 GOTO 1110
```

Les lignes 1260 à 1290 utilisent une autre boucle **FOR... NEXT** pour vérifier si la navette a rencontré un astéroïde. La ligne 1270 compare les coordonnées de la navette avec celles des astéroïdes stockés dans les tableaux A(S) et D(S). La ligne 1270 vérifie s'il y a eu collision avec le caractère graphique affiché à la ligne 1060.

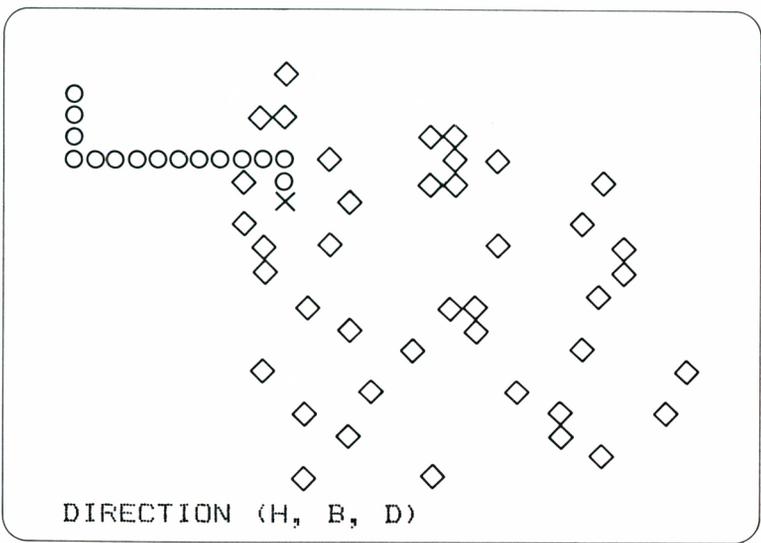
La boucle passe ainsi en revue les astéroïdes. La ligne 1300 vérifie si la navette est restée en orbite, c'est-à-dire sur l'écran. Si toutes les conditions sont vérifiées, la navette est affichée dans sa nouvelle position. Le dernier test consiste à vérifier si la navette a atteint la colonne 30. Si oui, le jeu se termine (partie suivante). Si non, la ligne 1340 renvoie le micro-processeur à la ligne 1110.



Il ne reste plus qu'à ajouter à notre programme les messages de fin. Nous les avons répartis sur deux lignes pour donner une apparence plus attrayante. L'instruction **END** termine ce programme.

```

1370 CLS:LOCATE 1,10:PRINT "VOUS AVEZ HEURTE UN AS
TEROIDE."
1380 LOCATE 1,13:PRINT "VOTRE VAISSEAU EST DEFINIT
IVEMENT      ENDOMMAGE.":STOP
1390 CLS:LOCATE 1,10:PRINT "VOUS AVEZ QUITTE VOTRE
ORBITE."
1400 LOCATE 1,13:PRINT "VOUS ETES PERDU DANS L'ESP
ACE.":STOP
1410 CLS:LOCATE 1,10:PRINT "BRAVO! VOUS LES AVEZ T
OUS EVITES!"
1420 LOCATE 1,13:PRINT "VOUS DEVEZ PILOTE PROFES
SIONNEL."
    
```



Lorsque vous jouez, n'oubliez pas de quelle manière les astéroïdes sont affichés à l'écran. La navette est composée d'un seul caractère graphique. Si ce caractère touche celui représentant un astéroïde, il y a collision.

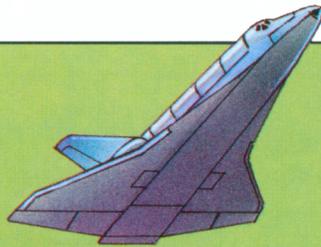
Vous pouvez rendre le jeu plus difficile en augmentant le nombre d'astéroïdes. Toutefois, au-delà de 120, il est pratiquement impossible de trouver un chemin. Modifiez simplement les lignes 1020, 1030 et 1260.



Donnez-lui des couleurs

Pour l'ordinateur THOMSON, les instructions **SCREEN** et **COLOR** déterminent les couleurs de l'écran et des caractères. En combinant les instructions **COLOR** et **RND** lors de l'affichage des astéroïdes, vous pouvez obtenir un champ d'astéroïdes multicolores.

```
1060 CL=RND(1)*5:LOCATE A(S),D(S):COLOR CL:PRINT G  
R$(0)
```



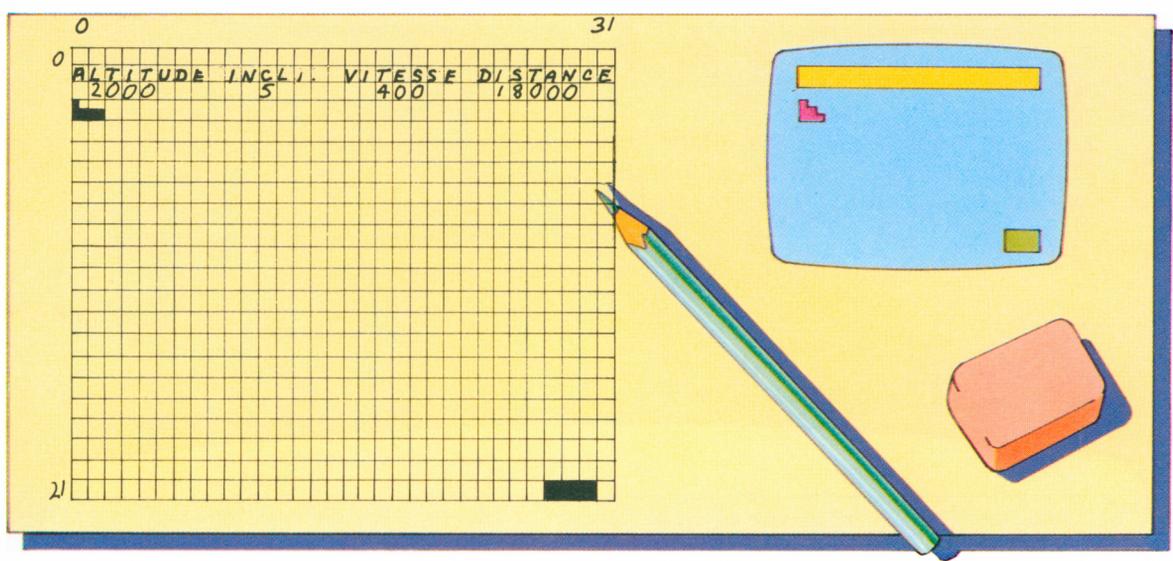
ATTERRISSAGE DE LA NAVETTE

En raison d'un orage d'astéroïdes, votre mission a été abrégée et vous devez revenir à la base. La piste d'atterrissage est en vue mais le pilote automatique est en panne. Vous devez poser la navette sans accident en utilisant vos deux commandes manuelles. Le tableau de bord vous donnera les informations nécessaires sur votre trajectoire.

ALTITUDE	INCLINAISON	VITESSE	DISTANCE
1200	5	400	11500



Comme dans le programme précédent, la première étape consiste à préparer l'écran sur une feuille de papier. Une fois encore, la navette va traverser l'écran de la gauche vers la droite, et de haut en bas. L'aire d'atterrissage est représentée par un rectangle placé dans le coin inférieur droit. Le haut de l'écran est réservé au tableau de bord et il indique l'altitude, la pente, la vitesse de la navette ainsi que la distance la séparant du terrain. Ces considérations déterminent un écran se présentant ainsi :



```

○ 2000 REM ***** ATERRISSAGE *****
2010 CLS
○ 2020 D=18000:S=400:H=2000:F=2
2030 C$=" "
○ 2040 LOCATE 1,1:COLOR 1:PRINT "ALTITUDE ";:COLOR
2:PRINT "INCLINAISON ";:COLOR 3:PRINT "VITESSE ";:
COLOR 4:PRINT "DISTANCE "
○ 2050 LOCATE 27,21:PRINT GR$(1);GR$(1);GR$(1)

```

La ligne 2020 détermine les valeurs initiales pour la distance, la vitesse, l'altitude et la pente. La ligne 2030 crée une chaîne de caractère **C\$** composée de 5 espaces et dont le rôle consistera à effacer les anciennes valeurs du tableau de bord. Le tableau de bord est affiché par la ligne 2040. Les instructions **COLOR** agrémentent l'affichage. N'oubliez pas de taper autant d'espaces qu'indiqué à la ligne 2040 afin d'obtenir une bonne disposition sur l'écran. L'aire d'atterrissage est dessinée par la ligne 2050. On utilise un caractère graphique représentant un carré plein.

```

2060 X=1:Y=3
2070 LOCATE X,Y:PRINT GR$(2);GR$(3)
    
```

La position de la navette à l'écran est déterminée par les coordonnées X et Y initialisées à la ligne 2060. X correspond à la position horizontale ; Y, à la position verticale. La navette est également composée de caractères graphiques.

```

2080 LOCATE 2,2:PRINT C$
2090 LOCATE 2,2:COLOR 1:PRINT H
2100 LOCATE 15,2:PRINT C$
2110 LOCATE 15,2:COLOR 2:PRINT F
2120 LOCATE 24,2:PRINT C$
2130 LOCATE 24,2:COLOR 3:PRINT S
2140 LOCATE 32,2:PRINT C$
2150 LOCATE 32,2:COLOR 4:PRINT D
    
```

Les lignes 2080 à 2150 affichent les valeurs numériques de l'altitude, de la pente, de la vitesse et de la distance. Là aussi, l'instruction **COLOR** est utilisée pour améliorer la présentation. La variable chaîne de caractère **C\$** utilisée aux lignes 2080, 2100, 2120 et 2140 efface les anciennes valeurs affichées avant qu'elles ne soient mises à jour. Cela donne l'impression d'un affichage « Temps Réel » tandis que la navette se déplace vers la droite et vers le bas pour atteindre l'aire d'atterrissage. La superposition de caractères est une méthode souvent utilisée dans les programmes informatiques pour produire des graphiques animés.

```

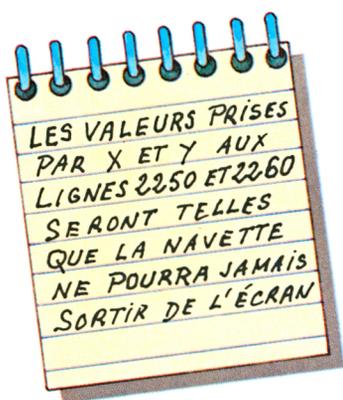
2160 IF S<200 THEN LOCATE 5,10:PRINT "NAVETTE IMMOBILE":STOP
2170 IF D<-999 THEN LOCATE 0,10:PRINT "LA NAVETTE A DEPASSE L'AIRES D'ATTERRIS- SAGE.":STOP
2180 IF H<=0 THEN GOTO 2290
    
```

Les lignes 2160 à 2180 vérifient que la vitesse n'est pas trop faible et que la navette n'a pas dépassé l'aire d'atterrissage d'une distance supérieure à 999. La ligne 2180 vérifie si la navette a atteint l'aire d'atterrissage avec une altitude nulle. Si oui, le programme passe à la ligne 2290 où un ultime test vérifie si la navette s'est bien posée sur l'aire d'atterrissage.

```

2190 R$="":R$=INKEY$
2200 IF R$="N" THEN F=F+1:S=S+5
2210 IF R$="W" THEN F=F-1:S=S-5
2220 IF F=0 THEN F=1:S=S-20
2230 H=H-F:S=S+(F-5)*3:D=D-INT(S/10)
2235 IF H<0 THEN H=0
2240 LOCATE X,Y:PRINT C$
2250 X=27-INT(D/(18000/28))
2260 Y=20-INT(H/(2000/18))
2270 LOCATE X,Y:COLOR 1:PRINT GR$(2);GR$(3)
2280 GOTO 2080
    
```

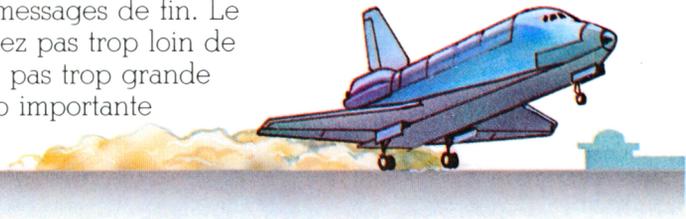
Les lignes 2190 à 2280 permettent au joueur de contrôler la descente de la navette et de modifier sa trajectoire en fonction des valeurs tapées. À la ligne 2190, l'instruction **INKEY\$** scrute le clavier et détecte l'enfoncement d'une touche. Le caractère correspondant à la touche est placé dans **R\$**. Les lignes 2200 et 2210 vérifient s'il s'agit d'un N ou d'un W (respectivement pour accélérer et décélérer). Si c'est le cas, les valeurs de pente et de vitesse sont modifiées. Notez que l'instruction **INKEY\$** n'arrête pas le programme tant que rien n'est tapé au clavier. À la ligne 2220, le programme vérifie que la pente n'atteint pas 0 : si c'était le cas, la navette s'immobiliserait dans l'espace. La ligne 2230 calcule les nouvelles valeurs de l'altitude, de la vitesse et de la distance ; la ligne 2240 efface les anciennes valeurs en utilisant **C\$**. Les nouvelles coordonnées X et Y de la navette sont calculées en 2250 et 2260. La ligne 2270 place la navette à sa nouvelle position et la ligne 2280 fait reprendre le programme à son début.



```

2290 IF D>1000 THEN LOCATE 5,10:PRINT "LA NAVETTE  
S'EST POSEE AVANT L'AIRE D'ATERRISSAGE.":STOP
2300 IF S>300 THEN LOCATE 5,10:PRINT "VITESSE TROP  
GRANDE.":STOP
2310 IF F>5 THEN LOCATE 5,10:PRINT "ANGLE D'ATTAKU  
E INACCEPTABLE.":STOP
2320 LOCATE 1,10:PRINT "ATERRISSAGE PARFAIT."  
2330 STOP
    
```

Puis le programme se termine sur les messages de fin. Le programme vérifie si vous ne vous posez pas trop loin de la base ($D > 1000$), si la vitesse n'est pas trop grande ($S > 300$), et si la pente n'est pas trop importante ($F > 5$).



Améliorez vos programmes

L'amélioration finale consiste à enchaîner les trois programmes. Vous trouverez le listing complet ainsi que les modifications effectuées ci-dessous.



Chainage des programmes

Un sous-programme est ajouté à la fin des trois programmes renumérotés ; il commence à la ligne 3000 et assure le chaînage d'une partie à une autre. Le message « ALERTE » apparaît à l'écran lorsqu'on passe d'une partie à une autre. Les lignes 440 et 1440 appellent ce sous-programme et les lignes 430 et 1430 déterminent le mot approprié devant être affiché.

THOMSON : listing complet

```
5 REM lancement de la navette
10 CLS: CLEAR 1000,,5: SCREEN 6,0,0
11 DEFGR$(0)=24,60,102,195,195,102,60,24
12 DEFGR$(1)=255,255,255,255,255,255,255,255
13 DEFGR$(2)=192,96,48,255,255,48,96,192
14 DEFGR$(3)=192,240,252,255,255,252,240,192
20 LOCATE 10,4: PRINT "LANCLEMENT DE LA NAVETTE"
30 LOCATE 10,5: PRINT "*****"
40 LOCATE 8,8: PRINT "NAVETTE PRETE AU DECOLLAGE"
50 LOCATE 11,10: PRINT "VOUS DEVEZ LA LANCER!"
60 LOCATE 8,13: PRINT "VOUS N'AVEZ QUE DIX ESSAIS"
70 LOCATE 8,16: PRINT "TAPEZ MAINTENANT VOTRE NOM"
80 LOCATE 9,18: PRINT "ET APPUYEZ SUR 'ENTREE' : "
90 LOCATE 15,20: INPUT " ",N$
100 CLS
110 PRINT "BIENVENUE A BORD CAPITAINES ";N$
120 LOCATE 1,4: PRINT "POUR LANCER LA NAVETTE, VOUS
    DEVEZ"
130 LOCATE 1,6: PRINT "DEVINEZ LE NIVEAU DE PUISSANCE
    REQUIS"
140 LOCATE 1,8: PRINT "(N'IMPORTE QUEL NOMBRE ENTRE
    1 ET 100)"
150 LOCATE 4,12: PRINT "TAPEZ-LE ET APPUYEZ SUR 'ENTREE' : "
160 LOCATE 4,14: PRINT "ALLEZ-Y !"
170 NIVEAU=INT((RND(1))*100)+1
180 FOR G=1 TO 10
185 LOCATE 4,16: PRINT "ESSAI";G
190 LOCATE 4,18: INPUT ESSAI
200 IF ESSAI=NIVEAU THEN GOTO 280
210 IF ESSAI<NIVEAU THEN LOCATE 8,20: PRINT "TROP BAS.
    ESSAYEZ ENCORE."
220 IF ESSAI>NIVEAU THEN LOCATE 8,20: PRINT "TROP HAUT.
    ESSAYEZ ENCORE."
230 NEXT G
240 CLS
250 LOCATE 1,4: PRINT "VOUS N'AVEZ PAS TROUVE APRES
    10 ESSAIS."
260 LOCATE 1,6: PRINT "TAPEZ 'RUN' (ENTREE) POUR RECOMMENCER."
270 STOP
280 CLS
290 HAUT=0
300 CIBLE=10000
310 LOCATE 10,2: PRINT "DECOLLAGE..."
320 LOCATE 10,10: PRINT "ALTITUDE : ";HAUT
340 FOR D=1 TO 100
350 NEXT D
360 HAUT=HAUT*2+1
365 PLAY"DO"
370 IF HAUT<CIBLE THEN GOTO 320
380 CLS
390 FOR R=5 TO 10
```

```
400 LOCATE 10,R: PRINT "NAVETTE EN ORBITE!"
410 NEXT R
420 LOCATE 5,1: PRINT "FELICITATIONS CAPITAINES ";N$
430 F$="ASTEROIDES"
440 GOSUB 3000
1000 REM ***** ASTEROIDES *****
1010 CLS
1020 DIM A(50),D(50)
1030 FOR S=1 TO 50
1040 A(S)=INT(RND(1)*28)+3
1050 D(S)=INT(RND(1)*18)+3
1055 XXX=INT(RND(1)*4)
1060 LOCATE A(S),D(S): PRINT GR$(0)
1070 NEXT S
1100 COL=0: LIGNE=2+INT(RND(1)*18)
1110 LOCATE COL,LIGNE: PRINT "X"
1160 LOCATE 1,1: INPUT "DIRECTION (H, B, D) ",D$
1170 IF D$="H" THEN HAUT=-1: DROITE=0: GOTO 1210
1180 IF D$="B" THEN HAUT=1: DROITE=0: GOTO 1210
1190 IF D$="D" THEN HAUT=0: DROITE=1: GOTO 1210
1200 GOTO 1160
1210 LOCATE 1,1: PRINT SPC(30): LOCATE 1,1: INPUT "DISTANCE",
    DISTANCE
1220 LOCATE COL,LIGNE: PRINT "O"
1230 FOR C=1 TO DISTANCE
1240 LIGNE=LIGNE+HAUT
1250 COL=COL+DROITE
1260 FOR S=1 TO 50
1270 IF COL=A(S) AND LIGNE=D(S) THEN GOTO 1370
1290 NEXT S
1300 IF LIGNE>20 OR LIGNE<2 THEN 1390
1310 LOCATE COL,LIGNE: PRINT "O"
1320 IF COL>30 THEN 1410
1330 NEXT C
1340 GOTO 1110
1370 CLS: LOCATE 1,10: PRINT "VOUS AVEZ HEURTE UN ASTEROIDE."
1380 LOCATE 1,13: PRINT "VOTRE VAISSEAU EST DEFINITIVEMENT
    ENDOMMAGE.";STOP
1390 CLS: LOCATE 1,10: PRINT "VOUS AVEZ QUITTE VOTRE ORBITE."
1400 LOCATE 1,13: PRINT "VOUS ETES PERDU DANS L'ESPACE.";STOP
1410 CLS: LOCATE 1,10: PRINT "BRAVO! VOUS LES AVEZ TOUTS EVITES!"
1420 LOCATE 1,13: PRINT "VOUS DEVEZ DEVENIR PILOTE PROFESSIONNEL."
1430 F$="ATTERRISSAGE"
1440 GOSUB 3000
2000 REM ***** ATTERRISSAGE *****
2010 CLS
2020 D=18000: S=400: H=2000: F=2
2030 C$=" "
```

```

2040 LOCATE 1,1:COLOR 1:PRINT "ALTITUDE ";:COLOR
2:PRINT "INCLINAISON ";:COLOR 3:PRINT "VITESSE ";:
COLOR 4:PRINT "DISTANCE "
2050 LOCATE 27,21:PRINT GR$(1);GR$(1);GR$(1)
2060 X=1;Y=3
2070 LOCATE X,Y:PRINT GR$(2);GR$(3)
2080 LOCATE 2,2:PRINT C#
2090 LOCATE 2,2:COLOR 1:PRINT H
2100 LOCATE 15,2:PRINT C#
2110 LOCATE 15,2:COLOR 2:PRINT F
2120 LOCATE 24,2:PRINT C#
2130 LOCATE 24,2:COLOR 3:PRINT S
2140 LOCATE 32,2:PRINT C#
2150 LOCATE 32,2:COLOR 4:PRINT D
2160 IF S<200 THEN LOCATE 5,10:PRINT "NAVETTE IMMO
BILE":STOP
2170 IF D<-999 THEN LOCATE 0,10:PRINT "LA NAVETTE
A DEPASSE L'AIRE D'ATTERRIS- SAGE.":STOP
2180 IF H<=0 THEN GOTO 2290
2190 R$="":R$=INKEY$
2200 IF R$="N" THEN F=F+1:S=S+5
2210 IF R$="W" THEN F=F-1:S=S-5
2220 IF F=0 THEN F=1:S=S-20
2230 H=H-F:S=S+(F-5)*3:D=D-INT(S/10)
2235 IF H<=0 THEN H=0
2240 LOCATE X,Y:PRINT C#

```

```

2250 X=27-INT(D/(18000/28))
2260 Y=20-INT(H/(2000/18))
2270 LOCATE X,Y:COLOR 1:PRINT GR$(2);GR$(3)
2280 GOTO 2080
2290 IF D>1000 THEN LOCATE 5,10:PRINT "LA NAVETTE
S'EST POSEE AVANT L'AIRE D'ATTERRISSAGE.":STOP
2300 IF S>300 THEN LOCATE 5,10:PRINT "VITESSE TROP
GRANDE.":STOP
2310 IF F>5 THEN LOCATE 5,10:PRINT "ANGLE D'ATTACH
E INACCEPTABLE.":STOP
2320 LOCATE 1,10:PRINT "ATTERRISSAGE PARFAIT."
2330 STOP
3000 REM ***** CHAINAGE *****
3010 FOR D=1 TO 1000
3020 NEXT D
3050 CLS
3060 FOR L=1 TO 10
3065 PLAY "D0"
3070 LOCATE 2,1:COLOR INT(RND(1)*3+1):PRINT "ALERT
E ! MANOEUVRES A EFFECTUER !"
3080 NEXT L
3090 FOR D=1 TO 10
3095 PLAY "D0"
3100 NEXT D
3110 RETURN

```

Glossaire

CHAINE Une chaîne de caractères est une suite de lettres, de symboles, de chiffres, placés entre guillemets. Une variable chaîne de caractères identifie une chaîne de caractères ; son nom doit obligatoirement se terminer par le symbole \$. L'ordinateur ne cherche jamais à interpréter ce qui est placé dans une chaîne de caractères.

CHR\$(code) L'instruction **CHR\$(code)** permet d'envoyer à l'écran le caractère correspondant au numéro placé entre parenthèses : le code. Ces caractères correspondent aux symboles gravés sur les touches du clavier ainsi qu'à des caractères spéciaux propres à chaque ordinateur.

DIM Cette instruction BASIC détermine la taille d'un tableau. Elle est suivie par un nombre placé entre parenthèses indiquant à l'ordinateur le nombre d'éléments dans ce tableau.

FOR... NEXT Cette séquence demande à l'ordinateur de répéter une opération un certain nombre de fois. Par exemple, la ligne **FOR X = 1 TO 5:PRINT 2*X:NEXT X** demandera à l'ordinateur d'écrire la table de multiplication par 2 pour les 5 premiers chiffres.

GOTO Cette instruction commande à l'ordinateur de sauter directement au numéro de la ligne spécifiée. Elle est souvent utilisée en conjonction avec l'instruction **IF... THEN**. *Attention* : si vous utilisez cette instruction trop souvent, vous rencontrerez de nombreuses difficultés pour suivre le déroulement logique du programme.

IF... THEN Les instructions placées après **THEN** sont exécutées seulement si la condition placée après **IF** est réalisée.

INKEY\$ Cette instruction demande à l'ordinateur de lire le clavier afin de déterminer si une touche a été enfoncée.

INPUT Lorsque l'ordinateur rencontre l'instruction **INPUT** il affiche un point d'interrogation à l'écran. L'utilisateur tape sa réponse et appuie sur **ENTREE** (ou RETURN).

INT(n) Cette fonction demande à l'ordinateur de ne considérer que la partie entière de nombre décimal n . Par exemple, **INT(3.1415)** donne 3.

LEFT\$ Cette instruction permet d'extraire d'une chaîne de caractères ses premiers caractères. **RIGHT\$** effectue une opération symétrique mais en commençant par la fin de la chaîne. **MID\$** est utilisée pour extraire un caractère ou un groupe de caractères n'importe où dans la chaîne. Toutes ces instructions sont suivies par des nombres indiquant le nombre de caractères devant être extraits. **MID\$** utilise un paramètre complémentaire pour indiquer le rang dans la chaîne de départ à partir duquel les caractères sont extraits.

LIST Cette commande fait défiler à l'écran l'ensemble du programme dans l'ordre croissant des numéros de ligne. Vous pouvez lister une ligne, une partie du programme ou le programme complet.

ON N GOTO... Cette instruction est suivie par une série de numéros de lignes repérées par des virgules. Si **N** vaut **1**, le programme saute au *premier* numéro de ligne suivant **GOTO**. Si **N** vaut **2**, le programme saute au *second* numéro, et ainsi de suite en fonction de la valeur de **N**.

PEEK Cette instruction permet de lire le contenu d'une case mémoire particulière. Elle est souvent associée à **POKE**.

POKE Cette instruction place dans une case mémoire particulière une donnée numérique entière comprise entre 0 et 255.

RND Cette instruction demande à l'ordinateur de générer un nombre aléatoire.
Par exemple **RND(A)** produire un nombre compris entre 0 et 1 avec 5 à 7 chiffres après la virgule. Pour obtenir un nombre entier dans une plus large gamme, il faut le multiplier et utiliser l'instruction **INT**.

STEP Cette instruction complète les instructions **FOR... NEXT**. Elle indique la valeur devant être ajoutée au compteur de boucles à chaque passage. Par exemple :
FOR X = 0 TO 20 STEP 5:PRINT X:NEXT X
produira l'affichage des nombres 0,5,10,15,20

Tableau Un tableau est un ensemble de données identifiées par le même nom de variable suivi d'un indice placé entre parenthèses. Vous pouvez comparer un tableau à une série de boîtes placées dans la mémoire de l'ordinateur et recevant chacune une donnée numérique.

Variable Lorsque vous donnez une information à l'ordinateur, vous devez lui donner un nom sous lequel il la mémorisera. Ce nom est appelé *variable* car la quantité qu'il désigne peut changer durant l'exécution du programme. Lorsque vous désirez atteindre l'information elle-même, vous utilisez son nom de variable.

Par exemple : **A = 6** place la valeur 6 dans la variable A. Il existe deux types de variables : une variable *numérique* pour mémoriser des nombres ; une variable *chaîne de caractères* pour mémoriser des caractères placés entre guillemets. Dans ce dernier cas, le nom de la variable se termine par le symbole \$.

Table de conversion

THOMSON	AMSTRAD
CLEAR 1000,,5	CLEAR SYMBOL AFTER 250
SCREEN 6,0,0	PAPER 0 BORDER 0 INK 1,20 PEN 1
DEFGR\$(n)=	SYMBOL n,
PLAY"DO"	SOUND 1,478,40,12
PRINT GR\$(n)	PRINT CHR\$(n+250)

Index

A
Affichage 8, 12, 17

C
COLOR 22, 24, 25
Coordonnée 8, 12, 20, 25,
26
Couleur 22, 24, 25
Curseur 7

D
Déverminage 10

E
Edition 7
Entier 13, 29

G
Graphique 17, 18, 21, 25

I
INKEY\$ 26,29
INPUT 13, 19, 30

L
LIST 7
LOAD 10

M
Magnétocassette 10
Message d'erreur 7, 18
Mise au point 10

N
Nombre aléatoire 8, 9, 12,
13, 30
Numéro de ligne 6

O

Organigramme 8, 9

P

Ponctuation 13

R

REM 6, 12, 17

S

SAVE 10

Symboles arithmétiques 6

T

Tableau 18, 20, 30

Temporisation 14

V

Variable 14, 19, 20, 31

Variable chaîne de
caractères 13, 19, 25, 31

LES ÉDITIONS FOUCHER

128, rue de Rivoli, 75001 Paris

N° 10359-186

Collection RENDEZ-VOUS MICRO

- ◇ **Première rencontre**
Guide d'initiation au micro-ordinateur
- ◇ **Pour mieux le comprendre**
Douze programmes expliqués. T07 M05
- ◇ **Voyage en BASIC**
Navette spatiale. M05 AMSTRAD
- ◇ **A la découverte des bases de données**
Jeu d'aventure : labyrinthe et fonds marins
T07 M05 AMSTRAD MSX
- ◇ **Votre second à la maison**
Programmes pratiques, programmes de jeux
T07 M05 AMSTRAD COMMODORE



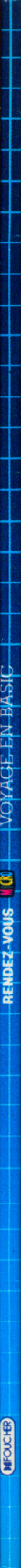
GUIDES PRATIQUES

- **SIL'Z 8 bits** - Modèles 2, 3, 4 et 8
- **T07 (70) . M05**
- **BULL . MICRAL** - 80-20/21/22 (G), 90-20
- **APPLE II** - Europlus e, c
- **IBM PC et compatibles**
Persona, Micral 90-30, Goupil 4...

ISBN 0-86313-206-5
ISBN 2-216-00932-6



147K 486
F 35.00



AMSTRAD

CPC



MÉMOIRE ÉCRITE
MEMORY ENGRAVED
MEMORIA ESCRITA



<https://acpc.me/>

[FRA] Ce document a été préservé numériquement à des fins éducatives et d'études, et non commerciales.

[ENG] This document has been digitally preserved for educational and study purposes, not for commercial purposes.

[ESP] Este documento se ha conservado digitalmente con fines educativos y de estudio, no con fines comerciales.