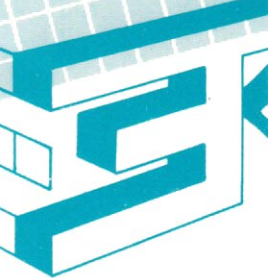


Mode d'emploi



# 3D MEGACODE

CPC 464-664

Auteurs: P. DUBLANCHET  
S. KOLECHI



**ERE**  
ERE INFORMATIQUE



Vous venez d'acquérir un logiciel ERE INFORMATIQUE et nous vous en félicitons.

Afin de profiter au mieux des qualités de ce produit, nous vous conseillons :

- De ranger la cassette dans son étui après utilisation ;
- De ne pas la stocker à proximité d'une source de chaleur trop intense ni dans le champ magnétique d'un haut-parleur ou d'un poste de télévision.
- De nettoyer régulièrement les têtes de votre magnétophone à l'aide d'un tissu imbibé d'alcool à 90° ou d'un produit spécialisé.



Programmes, sons et graphisme sont la propriété de Ere Informatique. Leur reproduction, traduction, adaptation, par quelque moyen que ce soit est strictement interdite.

La location de ce produit est strictement interdite.

© ERE INFORMATIQUE 1985



## SOMMAIRE

	Page
<b>1. Généralités</b>	
Chargement	6
Sauvegardes	7
<b>2. Casette</b>	8
<b>3. Aide</b>	9
<b>4. 3D</b>	13
<b>5. Graphiques</b>	21
<b>6. Horloge</b>	25
<b>7. Divers</b>	27
<b>8. Note à l'attention des concepteurs de logiciels</b>	29
<b>9. Table des instructions</b>	30

## GENERALITES

3D MEGACODE est un programme utilitaire composé de 6 parties appelées « PACK ». Chaque pack est constitué d'un certain nombre d'instructions basic que vous pouvez utiliser soit en mode direct, soit à l'intérieur d'un programme. Toutes ces instructions doivent être précédées du signe « I » que l'on obtient en appuyant simultanément sur (SHIFT) et sur « @ » (touche située tout de suite à droite de la lettre « P »).

Chaque pack se charge comme n'importe quel autre programme ; avec l'instruction « RUN'' nom de dossier ». Les 6 packs sont enregistrés à la suite les uns des autres sur la cassette, dans l'ordre suivant :

- CASS (utilisable sur CPC 464 uniquement)
- AIDE
- 3D
- GRA
- HORLO
- DIVER

Chaque pack se place automatiquement dans la mémoire juste avant « HIMEM », ce qui veut dire que vous pouvez charger le nombre de packs que vous désirez, dans n'importe quel ordre.

Si, dans votre programme, vous utilisez l'instruction « SYMBOL AFTER », faites-le avant le chargement d'un pack.

## SAUVEGARDES

Quel que soit le nombre de packs utilisés, la sauvegarde de vos programmes s'effectue de la manière suivante :

— CLEAR

— PRINT HIMEM

— Notez la valeur de HIMEM et insérez au début de votre programme basic :

1. MEMORY valeur de HIMEM

2. LOAD "I"

3. CALL valeur de HIMEM + 1.

Puis sauvez votre programme par :

SAVE "MEGACODE", B, valeur de HIMEM + 1, 42619 -  
valeur (version disque)

SAVE "MEGACODE", B, valeur de HIMEM + 1, 43902 -  
valeur (version cassette)

SAVE "nom du programme"

# **INSTRUCTIONS CASSETTE**

## **(CPC 464 uniquement)**

### **(499 octets)**

#### **INSTRUCTION : IBLOCK, kilo octets**

**EFFET :** Définit le nombre de kilo-octets par bloc d'enregistrement. A l'origine, chaque enregistrement est divisé en blocs de 2 K. Cette instruction permet donc d'agrandir la capacité des blocs et, par conséquent, d'accélérer la vitesse de chargement.

**ATTENTION !** Un programme enregistré avec des blocs de plus de 2 K doit être chargé après avoir utilisé l'instruction ILOAD,1, quant aux programmes en binaire, ils doivent être rechargés à la même adresse que lors de la sauvegarde.

#### **INSTRUCTION ILOAD,1**

**EFFET :** Quand cette instruction est exécutée, les programmes enregistrés avec des blocs de plus de 2K peuvent être chargés normalement par un « LOAD'' » ou un « RUN'' ».

#### **INSTRUCTION ILOAD,0**

**EFFET :** Remet l'instruction « LOAD » dans son état d'origine. Les programmes enregistrés avec des blocs de plus de 2K ne peuvent plus être chargés.

#### **INSTRUCTION SPEED WRITE 2**

**EFFET :** A l'origine, cette instruction permet de choisir la vitesse d'enregistrement entre 2 000 et 1 000 bauds. Avec ce paramètre supplémentaire, il est maintenant possible d'enregistrer à 3 600 bauds !



# INSTRUCTIONS D'AIDE A LA PROGRAMMATION

(3 238 octets)

**GENERALITES** : Ces instructions sont en fait des extensions à l'instruction « LIST ». Elles permettent de rechercher et de « déboguer » beaucoup plus facilement un long programme basic.

Les paramètres sont les mêmes que pour l'instruction « LIST » :

Première ligne, dernière ligne et numéro du canal. Seule la syntaxe change, voici des exemples avec l'instruction « ILT » qui resteront valable pour « ILE » et « ILR » :

ILT	• Listage complet sur tout l'écran
ILT,10,200	• Listage de la ligne 10 à la ligne 200
ILT,10,200,8	• Comme précédemment, mais sur imprimante

**Remarques** : • Quand vous mettez un numéro de première ligne, le numéro de dernière ligne est obligatoire.

• Entre parenthèses, figurent les paramètres optionnels.

**INSTRUCTION** : ILT (,1<sup>re</sup> ligne, dernière ligne) (,N° du canal)

**EFFET** : Liste uniquement les titres, c'est-à-dire toutes les lignes commençant par une apostrophe.

**EXEMPLE** : 10 ' Début  
20 PRINT "Programme"  
30 ' Milieu  
40 PRINT "Programme"  
50 ' Fin

l'instruction « ILT », listera les lignes 10, 30 et 50.

**INSTRUCTION : ILR** (,1<sup>re</sup> ligne, dernière ligne) (,N° du canal)

**EFFET** : Liste en repérant par un petit triangle placé en début de ligne, toutes les lignes appelées par « GOTO » ou « GOSUB » (ainsi que ON GOTO, ON GOSUB, etc.).

**EXEMPLE** : 10 GOTO 30  
20 PRINT  
30> PRINT "ICI"  
40 GOSUB 50  
50> PRINT "ICI AUSSI" : RETURN

**INSTRUCTION : ILE** (,1<sup>re</sup> ligne, dernière ligne) (,N° du canal)

**EFFET** : Liste uniquement les lignes appelées par « GOTO » ou « GOSUB ».

**EXEMPLE** : Dans le listing précédent, seules les lignes 30 et 50 auraient été listées.

**INSTRUCTION : ILG** (,1<sup>re</sup> ligne, dernière ligne), N° de ligne cherchée (,N° du canal)

**EFFET** : Cherche dans le listing les appels par « GOTO » ou « GOSUB » d'une ligne spécifiée par le paramètre « N° de ligne cherchée ».

**EXEMPLE** : Dans le listing précédent, l'instruction « ILG,30 » aurait listé uniquement la ligne 10.

**INSTRUCTION ILBL** (,1<sup>re</sup> ligne)

**EFFET** : Liste ligne par ligne dans une fenêtre en haut de l'écran. Les flèches de déplacement du curseur permettent de se déplacer dans le listing (en largeur comme en hauteur). Pour sortir de ce mode, appuyez sur la touche « ENTER ».

**INSTRUCTION : ILBT** (,1<sup>re</sup> ligne)

**EFFET** : Liste titre par titre dans une fenêtre en haut de l'écran. Les flèches de déplacement du curseur permettent de se déplacer dans le listing. Pour sortir de ce mode appuyez sur la touche « ENTER ».

**Remarque** : Vous pouvez passer de l'instruction « ILBL » à l'instruction « ILBT » et vice-versa en appuyant sur la touche « COPY ».

**INSTRUCTION : IROT**, 1<sup>re</sup> ligne, 2<sup>e</sup> ligne

**EFFET** : Inverse deux numéros de ligne.

**INSTRUCTION : ICHG**, nouveau numéro, ancien numéro

**EFFET** : Change un numéro de ligne.

**INSTRUCTION : IMOV**, destination, 1<sup>re</sup> ligne, dernière ligne

**EFFET** : Déplace un bloc de lignes d'un endroit à un autre. Le paramètre « destination » doit être le numéro de ligne précédant l'emplacement désiré. Les paramètres 1<sup>re</sup> et dernière ligne délimitent le bloc à déplacer.

**EXEMPLE** : Avant l'instruction IMOV,100,10,30 :

```
10 PRINT
20 INPUT N
30 PRINT N
40 REM
50 REM
100 PRINT "Ligne 100"
110 END
```

Après :

```
40 REM
50 REM
100 PRINT "Ligne 100"
101 PRINT
102 INPUT N
103 PRINT N
110 END
```

**INSTRUCTION : IDEFCAR, numéro de caractère**

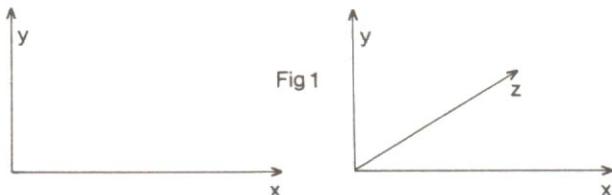
**EFFET** : Permet de redéfinir un caractère très facilement. Si vous voulez redéfinir le caractère 255, tapez : IDEFCAR,255 puis « RETURN ». Apparaît alors à l'écran le caractère 255 dans une grille de 8 sur 8. Positionnez ensuite le curseur sur les points à modifier puis appuyez sur « COPY » pour marquer un point et sur la barre d'espace pour l'effacer. Quand le caractère est complètement modifié, appuyez sur « ENTER ». L'instruction « SYMBOL » apparaîtra sous la grille avec les nouvelles valeurs. Vous pouvez valider la ligne directement en appuyant sur « ENTER » ou insérer un numéro de ligne avant de valider.

# INSTRUCTIONS GRAPHIQUES 3 DIMENSIONS (3 865 octets)

## PREMIERE PARTIE : INSTRUCTIONS 3D.

**IMOVE3D, IMOVER3D, IPLOT3D, IPLOTR3D,  
IDRAW3D, IDRAWR3D.**

Il s'agit des instructions graphiques d'origine, avec un paramètre supplémentaire. Ce paramètre « Z » définira la troisième dimension.



La syntaxe des nouvelles instructions est la même que pour les instructions d'origine, avec en plus, le paramètre « Z ». **IDRAW3D, coorX, coorY, coorZ (,Pen).**

Entre parenthèses, figurent les paramètres optionnels.

**EXEMPLE :** 5 IPERS,600,300  
10 MODE 1  
20 FOR i=0 TO 10 000 STEP 20  
30 IDRAW3D,400,0,i  
35 IMOVER3D, -400,0,20  
40 NEXT

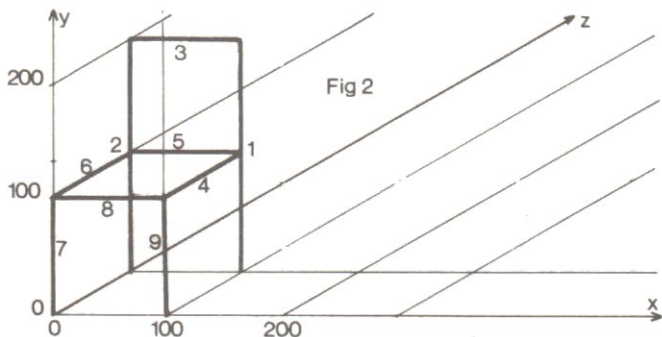
L'instruction « IPERS » définit le point de fuite aux coordonnées X, Y. Le meilleur moyen de bien comprendre son effet est de lancer le programme précédent en changeant les coordonnées de l'instruction en ligne 5.

## DEUXIEME PARTIE : CREATION D'UN OBJET GRAPHIQUE

Le grand intérêt de ce pack réside surtout dans la possibilité de définir des objets et de les afficher sur l'écran en trois dimensions. Pour bien comprendre chaque instruction, nous allons créer une chaise, l'afficher et la mouvoir dans les trois dimensions.

### 1. DEFINITION D'UN OBJET

La première chose à faire est de dessiner l'objet sur une feuille en indiquant les coordonnées X, Y et Z :



Notre chaise est donc composée de 9 segments de droite numérotés de 1 à 9. Il faut maintenant réserver de la place en mémoire pour stocker les divers paramètres de la chaise. Pour cela nous allons employer l'instruction « DIM ».

**DIM** nom de l'objet%(nombre de segments\*7 + 4)

« nom de l'objet » est une variable qui servira de nom à l'objet. Pour notre exemple nous prendrons « chaise ».

« nombre de segments » est le nombre de droites ou de points qui composent l'objet.

**EXEMPLE :** DIM chaise%(9\*7 + 4)

La place mémoire étant réservée, il s'agit maintenant de définir de quoi sera composé l'objet avec l'instruction :

**IDEFOBJ,@ nom de l'objet%(0), dimension, éléments, abs/rel, Pen**

Les paramètres sont très importants, car c'est d'eux dont dépend la syntaxe des « instructions d'éléments ». Voyons les un par un :

Nom de l'objet : Il s'agit tout simplement de la variable « chaise », précédée du signe « @ » et suivie de « %(0) »

Dimension : 2 = objet en 2 dimensions. 3 = objet en trois dimensions. Si vous choisissez 2, vous n'aurez pas à donner de paramètre « Z » dans les « instructions d'éléments ». Si vous choisissez 3, vous devrez impérativement préciser le paramètre « Z ».

Éléments : Ce paramètre définit de quoi sera composé l'objet. Vous avez le choix entre 3 possibilités :

0 = points. Vous ne pourrez dessiner l'objet qu'avec des points. Les coordonnées seront X, Y et Z si vous avez choisi 3D.

1 = suite de segments de droite. Vous devrez dessiner l'objet par une succession de droites liées les unes aux autres. Les coordonnées seront X, Y et Z si vous avez choisi 3D.

2 = segments de droite indépendants. L'objet sera dessiné par des droites indépendantes les unes des autres. Les coordonnées seront X, Y, (Z pour 3D), X', Y', (Z' pour 3D).

ABS/REL : Ce paramètre définit simplement de quelle manière seront calculées les coordonnées : 0 = absolu, 1 = relatif.

Pen : Si le paramètre est à zéro, vous ne devez pas donner de paramètre « PEN » dans les instructions d'éléments. S'il est à 1, vous devez obligatoirement préciser le paramètre « PEN ».

**EXEMPLE** : IDEFOBJ,@ chaise%(0),3,2,0,1

3D segments ind. absolu avec Pen

## 2. CONSTRUCTION DE L'OBJET

Maintenant que l'objet est défini, il faut le construire. Pour cela, imaginez une liste d'éléments qui composeront l'objet. Nous avons choisi dans l'instruction « IDEFOBJ », des segments indépendants. Ce qui veut dire que notre chaise sera obligatoirement dessinée avec des droites indépendantes. Nous devons donc entrer dans notre liste les 9 éléments qui constituent notre chaise. Pour cela, nous disposons de 5 instructions d'éléments :

**IADEL**, @ nom de l'objet%(0), paramètres dépendants de l'instruction « IDEFOBJ » (voir plus loin toutes les possibilités).

**EFFET** : Ajoute un élément à une liste. C'est l'instruction la plus utilisée, les autres servant à corriger et à modifier le dessin en cours de programme.

**IDELEL**, @ nom de l'objet%(0)

**EFFET** : Supprime le dernier élément d'une liste.

**IMODEL**, @ nom de l'objet%(0), numéro d'élément, paramètres selon « IDEFOBJ »

**EFFET** : Modifie un élément déjà présent dans une liste. « Numéro d'éléments » indique l'élément à modifier en comptant à partir du premier entré.

**ISUPEL**, @ nom de l'objet%(0), numéro d'élément

**EFFET** : Supprime un élément précis dans une liste.

**IINSEL**, @ nom de l'objet%(0), numéro d'élément, paramètres selon « IDEFOBJ »

**EFFET** : Insère un nouvel élément dans une liste. Le numéro d'élément indique l'élément devant précéder le nouveau.



## Possibilités de paramètre selon

« IDEF OBJ » :

INSTRUCTION « IDEF OBJ »	PARAMETRES POUR LES INSTRUCTIONS D'ELEMENTS
IDEF OBJ, @ chaise% (0), 3, 0, 0, 0	X, Y, Z
IDEF OBJ, @ chaise% (0), 3, 1, 0, 0	X, Y, Z
IDEF OBJ, @ chaise% (0), 3, 2, 0, 0	X, Y, Z, X', Y', Z'
IDEF OBJ, @ chaise% (0), 3, 0, 0, 1	X, Y, Z, PEN
IDEF OBJ, @ chaise% (0), 3, 1, 0, 1	X, Y, Z, PEN
IDEF OBJ, @ chaise% (0), 3, 2, 0, 1	X, Y, Z, X', Y', Z', PEN
IDEF OBJ, @ chaise% (0), 2, 0, 0, 0	X, Y
IDEF OBJ, @ chaise% (0), 2, 1, 0, 0	X, Y
IDEF OBJ, @ chaise% (0), 2, 2, 0, 0	X, Y, X', Y'
IDEF OBJ, @ chaise% (0), 2, 0, 0, 1	X, Y, PEN
IDEF OBJ, @ chaise% (0), 2, 1, 0, 1	X, Y, PEN
IDEF OBJ, @ chaise% (0), 2, 2, 0, 1	X, Y, X', Y', PEN

Pour simplifier la saisie des exemples vous pouvez utiliser l'instruction KEY du basic AMSTRAD. Exemple : KEY 138, « @ chaise% (0) » redéfinit la touche • du pavé numérique en « @ chaise% (0) », ce qui vous évite de taper à chaque fois toute la séquence.

### EXEMPLE :

```
10 DIM      chaise% (9*7 + 4)
20 IDEF OBJ, @ chaise% (0), 3, 2, 0, 1
30 IADEL, @ chaise% (0), 100, 200, 100, 100, 0, 100, 3
40 IADEL, @ chaise% (0), 0, 0, 100, 0, 200, 100, 3
50 IADEL, @ chaise% (0), 0, 200, 100, 100, 200, 100, 3
60 IADEL, @ chaise% (0), 100, 100, 0, 100, 100, 100, 2
70 IADEL, @ chaise% (0), 100, 100, 100, 0, 100, 100, 2
80 IADEL, @ chaise% (0), 0, 100, 100, 0, 100, 0, 2
90 IADEL, @ chaise% (0), 0, 0, 0, 0, 100, 0, 1
100 IADEL, @ chaise% (0), 0, 100, 0, 100, 100, 0, 1
110 IADEL, @ chaise% (0), 100, 100, 0, 100, 0, 0, 1
```

Comparez les paramètres des instructions « IADEL » avec la figure 2 et vous comprendrez ce que signifie les coordonnées X, Y et Z.

### 3. AFFICHAGE DE L'OBJET

Pour afficher un objet construit, utilisez l'instruction « IDES,@ nom de l'objet(0) ».

EXEMPLE : 120 IDES,@chaise%(0)

Ajoutez cette ligne au programme précédent et faites le « tourner ».

### 4. DEPLACEMENT DE L'OBJET

Nous allons voir maintenant les instructions qui permettent de déplacer l'objet n'importe où sur l'écran en 3 dimensions, de le faire tourner sous n'importe quel angle et dans n'importe quel axe et de le sauvegarder en mémoire pour pouvoir le réutiliser ultérieurement.

**INSTRUCTIONS :** ITRANS,@ nom de l'objet%(0), incrX, incrY (,incrZ avec 3D)

**EFFET :** Translate l'objet en additionnant les paramètres aux coordonnées X, Y et Z. Les modifications ne sont apportées qu'en mémoire. Pour voir l'objet à sa nouvelle place il faut redemander l'affichage par l'instruction « IDES ».

EXEMPLE :

```
115 ITRANS,@chaise%(0),0,0,1200
```

```
120 FOR j = 1 TO 5
```

```
130 FOR i = 1 TO 7
```

```
140 IDES,@chaise%(0)
```

```
150 ITRANS,@chaise%(0),0,0,-200
```

```
160 NEXT
```

```
170 ITRANS,@chaise%(0),150,0,1400
```

```
180 NEXT
```

Supprimez la ligne 120 du programme précédent et ajoutez-y ces quelques lignes.

**INSTRUCTION IROTAXZ**,@nom de l'objet%(0), coorX, coorY, angle

**EFFET** : Fait tourner l'objet autour de l'axe Z en prenant comme centre les coordonnées X, Y. L'angle de rotation est en radian. Pour l'obtenir en degrés, faites précéder l'instruction de « DEG ». ( Fonctionne aussi en 2D )

**EXEMPLE** :

```
120 DEG
130 ITRANS,@chaise%(0),200,100,500
140 FOR i = 1 TO 8
150 IDES,@chaise%(0)
160 IROTAXZ,@chaise%(0),250,20,45
170 NEXT
```

Remplacez les dernières lignes du programme précédent par celles-ci puis faites « tourner » le programme. Amusez-vous à changer les paramètres en observant les modifications, c'est encore la meilleure méthode pour bien comprendre le fonctionnement de ces instructions. Sachez seulement qu'une coordonnée Z en dessous de zéro fera apparaître le message d'erreur : « subscript out of range ».

**INSTRUCTION : IROTAXX**,@nom de l'objet%(0), coorY, coorZ, angle

**EFFET** : Fait tourner l'objet autour de l'axe X en prenant comme centre les coordonnées Y, Z. Le principe est le même que pour « IROTAXZ ».

**EXEMPLE** :

```
120 DEG
130 ITRANS,@chaise%(0),0,0,600
140 FOR i = 1 TO 8
150 IDES,@chaise%(0)
160 IROTAXX,@chaise%(0),100,450,45
170 NEXT
```

Faites comme pour « IROTAXZ ».

**INSTRUCTION : IROTAXY**, @nom de l'objet%(0), coorX, coorZ, angle

**EFFET** : Fait tourner l'objet autour de l'axe Y en prenant comme centre les coordonnées X, Z. Voyez « IROTAXZ » pour plus d'informations.

**EXEMPLE :**

```
120 DEG
130 ITRANS,@chaise%(0),200,0,500
140 FOR i = 1 TO 8
150 IDES,@chaise%(0)
160 IROTAXY,@chaise%(0),250,350,45
170 NEXT
```

**INSTRUCTION IMEMOBJ**, @nouveau nom%(0), nom de l'objet%(0)

**EFFET** : Sauvegarde en mémoire un objet, dans un autre tableau, afin de le retrouver ultérieurement. « Nouveau nom » est une variable. Comme « nom de l'objet », cette variable doit être dimensionnée par une instruction « DIM ».

**EXEMPLE :**

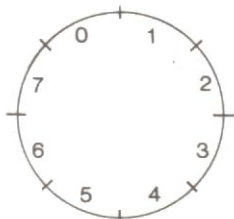
```
120 DEG : DIM effa(9*7 + 4)
130 ITRANS,@chaise%(0),200,0,500
140 IMEMOBJ,@effa%(0),@chaise%(0)
150 IDES,@effa%(0),0
160 IDES,@chaise%(0)
170 IMEMOBJ, @effa%(0),@chaise%(0)
180 IROTAXY, @chaise%(0),250,350,45
190 GOTO 150
```

Ce programme montre un exemple de l'utilisation de l'instruction « IMEMOBJ ». L'objet est mémorisé en ligne 170 dans la variable « effa », afin de pouvoir être effacé en ligne 150. Ceci permet une plus grande rapidité dans le déplacement de l'objet.

## INSTRUCTIONS GRAPHIQUES (2 425 octets)

**INSTRUCTION : ICIRCLE**, rayon, arcs (,Pen)

**EFFET :** Trace un cercle ou un arc de cercle d'un rayon R en prenant comme centre la position du curseur graphique. Le paramètre « ARCS » définit quelle(s) partie(s) du cercle doit (doivent) être tracée(s). Le cercle se divisant en 8 arcs numérotés comme suit :



Chaque arc de cercle est représenté par un bit du paramètre. Quand le bit est à 1, l'arc est tracé et ne l'est pas quand il est à 0. Pour plus de facilité, nous vous conseillons de représenter le paramètre en binaire : Pour tracer les arcs 3 et 0, écrire &×00001001

(76543210)

Le paramètre PEN est facultatif.

**EXEMPLE :**

```
4 MODE 1
```

```
5 ORIGIN 320,200
```

```
10 FOR i = 1 TO 150 STEP 6
```

```
20 ICIRCLE,i,&×01010101,2
```

```
30 ICIRCLE,i,&×10101010,3
```

```
40 NEXT
```

**INSTRUCTION : IFILL (, Pen1, Pen2)**

EFFET : Colorie une zone fermée à partir du curseur graphique. Les paramètres PEN1 et PEN2 (facultatifs) précisent la ou les couleurs du coloriage. S'il y a 2 paramètres différents, l'ensemble aura un aspect trame.

Remarque : Pour travailler, FILL utilise le buffer cassette. Ce qui laisse disponible la ram libre. Si vous tentez de colorier un écran trop complexe, (rempli de caractères, par exemple) un message « MEMORY FULL » apparaîtra. Le coloriage peut être interrompu à tout moment par la touche « ESC ».

EXEMPLE :

```
4 MODE 1
5 ORIGIN 0,0
10 FOR i = 1 TO 15
20 MOVER 40,28
30 ICIRCLE,50,255,1
40 NEXT
50 MOVE 10,10
60 FOR i = 1 TO 15
70 MOVER 40,28
80 IFILL, INT(i/4),i
90 NEXT
```

**INSTRUCTION ITCAR, hauteur, largeur (, Pen1, Pen2)**

EFFET : Définit la taille des caractères. La hauteur peut changer de 1 à 10 et la largeur de 1 à 8. Les paramètres PEN1 et PEN2 (facultatifs) permettent de définir des couleurs tramees, comme pour l'instruction « FILL ».

EXEMPLE :

```
5 MODE 1 : x = 1
10 FOR i = 1 TO 6
20 LOCATE x,x
30 ITCAR,i,i,i,i + 1
40 PRINT "ERE"
50 x = x + i
60 NEXT
70 ITCAR,1,1
```

### **INSTRUCTION : IINVERSE**

EFFET : Inverse les encres affectées à PEN et à PAPER.

EXEMPLE :

```
4 CLS
5 FOR i=1 TO 70
10 PRINT "bonjour" ; : IINVERSE
20 NEXT
```

### **INSTRUCTION : IScroll,1**

EFFET : Passe le scrolling de l'écran en mode Pixel.

EXEMPLE :

```
5 MODE 1
10 WINDOW 1,10,30,6,19
20 PAPER 1,2 : PEN 1,3 : CLS 1
25 FOR I=1 TO 14
30 PRINT I, "ERE INFORMATIQUE"
35 NEXT
38 IScroll,1
40 FOR I=1 TO 14 : PRINT I : NEXT
50 IScroll,0
```

### **INSTRUCTION : IScroll,0**

EFFET : Remet le scrolling de l'écran en mode caractère

### **INSTRUCTION IBOX, largeur, hauteur (,Pen)**

EFFET : Trace une boîte carrée ou rectangulaire. Le coin en bas et à gauche étant la position du curseur graphique. Le paramètre « PEN » est facultatif.

EXEMPLE :

```
5 CLS
10 MODE 1
20 ORIGIN 100,100
25 FOR i=1 TO 30
30 IBOX,100,100,1
40 IBOX,-100,100,2
50 IBOX,-100,-100,3
60 IBOX,100,-100,2
70 MOVER 10,10
80 NEXT
```

**INSTRUCTION ISQUA**, coorX, coorY, cote, angle (,Pen)  
EFFET : Trace un carré à partir des coordonnées X et Y, dans n'importe quel angle. A l'origine, le calcul est fait en radians. Si vous voulez définir l'angle en degrés, ajoutez « DEG » dans votre programme avant l'instruction « ISQUA ». Le paramètre « Pen » est facultatif.

**EXEMPLE :**

```
10 MODE 1
```

```
20 DEG
```

```
30 FOR i= 1 TO 360 STEP 4
```

```
40 ISQUA,320,200,100,i,INT(i/24)
```

```
50 NEXT
```



## **INSTRUCTIONS D'HORLOGE** (759 octets)

**INSTRUCTION : ITIME**, heures, minutes, secondes  
EFFET : Met l'horloge à l'heure

**INSTRUCTION : IPRTIME**, paper, Pen (,coorX, coorY)  
EFFET : Affiche l'heure, en permanence, aux coordonnées X, Y. Si ces coordonnées ne sont pas précisées, l'heure sera affichée en haut à droite de l'écran. Les paramètres PAPER et PEN sont obligatoires. Pour éviter que l'heure s'efface à chaque CLS, il suffit de protéger l'écran dans une fenêtre comme ceci : WINDOW 0,1,40,2,25.

**INSTRUCTION : IPRTIME,0**  
EFFET : Supprime l'affichage de l'heure.

**INSTRUCTION : IWAIT**, temps en secondes  
EFFET : Suspend l'exécution du programme pendant un certain temps. Ce temps peut atteindre 65 535 secondes. Il est toujours possible d'interrompre cette suspension avec la touche « ESC ».

**INSTRUCTION : IALARM**, heures, minutes, secondes, répétition.  
EFFET : Met l'alarme en fonction. Précise l'heure à laquelle l'ordinateur sonnera. Quand le paramètre « REPETITION » est 1, la sonnerie se répètera jusqu'à l'appui d'une touche quelconque. Si ce paramètre est 0, l'ordinateur ne sonnera qu'une fois.

**INSTRUCTION : IALARM,0**  
EFFET : Met l'alarme hors fonction.

## PROGRAMMES D'EXEMPLE :

Ce programme affiche l'heure au centre de l'écran et attend midi avant de « rendre la main ».

```
10 MODE 1
```

```
15 ITIME,11,49,45
```

```
20 IPRTIME,2,3,16,12
```

```
30 IWAIT,15
```

```
40 IPRTIME,0
```

```
50 CLS
```

Ce programme met l'horloge à 1 h 29 m 45 s, puis l'alarme à 1 h 30 m.

```
10 MODE 1
```

```
20 LOCATE 3,10 : PRINT "L'horloge sonnera" : LOCATE
```

```
11,14 : PRINT "à 1 h 30 m"
```

```
30 IALARM,1,30,0,0
```

```
40 ITIME,1,29,45
```

```
50 IPRTIME,2,3
```

```
60 PRINT : PRINT
```

## INSTRUCTIONS DIVERSES (573 octets)

### INSTRUCTION : ITESTT

EFFET : Passe l'instruction basic résidente « TEST » en mode texte. Après cette instruction, l'instruction TEST (1,1) ira lire le CARACTERE se trouvant en haut et à gauche de l'écran, et non plus le point graphique.

EXEMPLE :

```
10 MODE 1
20 ITESTT
30 PRINT "recopier cette phrase"
35 PRINT
40 FOR i=1 TO 21
50 PRINT CHR$(TEST(i,1)) ;
60 NEXT
70 ITESTG
```

### INSTRUCTION : ITESTG

EFFET : Remet l'instruction basic résidente « TEST » en mode graphique.

## **INSTRUCTION ICOPYT**

**EFFET :** Recopie sur n'importe quelle imprimante, la totalité de l'écran TEXTE. La copie peut être interrompue à n'importe quel moment en appuyant sur la touche « ESC ».

**INSTRUCTION : ISRT**, variable alphanumérique définie  
**EFFET :** Trie par ordre alphabétique un tableau alphanumérique à une dimension. Supposons un tableau de noms propres appelés NOM\$(100), après l'instruction ISRT, NOM\$(0), tous les noms seront triés par ordre alphabétique.

**EXEMPLE :** Ce programme vous demande 10 chaînes de caractères puis les trie, avant de les afficher dans l'ordre alphabétique.

```
10 DIM a$(10)
20 FOR i=0 TO 10
30 INPUT a$(i)
35 NEXT
40 ISRT, a$(0)
50 FOR i=0 TO 10
60 PRINT a$(i)
70 NEXT
```

## **INSTRUCTION : IEXPLO**

**EFFET :** Produit un bruit d'explosion

## **NOTE A L'ATTENTION DES AUTEURS DE LOGICIEL**

Comme vous avez certainement dû le constater, 3D MEGACODE est un outil quasi indispensable pour la réalisation de programmes BASIC performants.

Si vous avez réalisé, à l'aide de 3D MEGACODE, un programme dont vous pensez qu'il possède une valeur commerciale certaine, contactez-nous. Nous pourrions alors étudier ensemble la possibilité de diffuser ce programme, en accord avec les auteurs de 3D MEGACODE.

Il va sans dire qu'en cas de commercialisation, vous-même et les auteurs originaux toucheront une redevance proportionnelle aux ventes.

Pour de plus amples détails, veuillez écrire à ERE INFORMATIQUE, à l'attention de M. le directeur artistique.

A bientôt

## TABLE DES MATIERES

INSTRUCTION :	PACK :	PAGE :
IADEL	3D _____	16
IALARM	HORLO _____	25
IBLOCK	CASS _____	8
IBOX	GRA _____	23
ICHG	AIDE _____	11
ICIRCLE	GRA _____	21
ICOPYT	DIVERS _____	28
IDEFCAR	AIDE _____	12
IDEFOBJ	3D _____	15
IDELEL	3D _____	16
IDES	3D _____	13
IDRAW3D	3D _____	13
IDRAWR3D	3D _____	13
IEXPLO	DIVERS _____	28
IFILL	GRA _____	22
IINSEL	3D _____	16
IINVERSE	GRA _____	23
ILBL	AIDE _____	10
ILBT	AIDE _____	11
ILE	AIDE _____	10
ILG	AIDE _____	10
ILOAD	CASS _____	8

ILR	AIDE _____	10
ILT	AIDE _____	9
IMEMOBJ	3D _____	20
IMODEL	3D _____	16
IMOV	AIDE _____	11
IMOVE3D	3D _____	13
IMOVER3D	3D _____	13
IPERS	3D _____	13
IPLLOT3D	3D _____	13
IPLOTR3D	3D _____	13
IPRTIME	HORLO _____	25
IROT	AIDE _____	11
IROTAXX	3D _____	19
IROTAXY	3D _____	20
IROTAXZ	3D _____	19
IScroll	GRA _____	23
SPEED WRITE	CASS _____	8
ISQUA	GRA _____	24
ISUPEL	3D _____	16
ITCAR	GRA _____	22
ITESTG	DIVERS _____	27
ITESTT	DIVERS _____	27
ITIME	HORLO _____	25
ITRANS	3D _____	18
ISRT	DIVERS _____	28
IWAIT	HORLO _____	25





## AMSTRAD

<b>Simulation</b>	LE MILLIONNAIRE MISSION DELTA MACADAM BUMPER MANAGER
<b>Réflexion</b>	MICRO SAPIENS CHIROLOGIE
<b>Aventure</b>	AMELIE MINUIT
<b>Action</b>	GUTTER LE SURVIVANT
<b>Utilitaires</b>	3D MEGACODE COMPILATEUR
<b>Wargames</b>	THEATRE EUROPE LA BATAILLE D'ANGLETERRE BATAILLE POUR MIDWAY

## SPECTRUM

<b>Simulation</b>	INTERCEPTEUR COBALT MANAGER MACADAM BUMPER MILLIONNAIRE
<b>Réflexion</b>	MICRO SAPIENS
<b>Utilitaires</b>	ZX TRANS BASIC ETENDU PRINT + VOX 3D MOVER COMPILATEUR INTEGRAL TRAFFIC 1-0
<b>Wargames</b>	THEATRE EUROPE LA BATAILLE D'ANGLETERRE BATAILLE POUR MIDWAY
<b>Action</b>	5 SUR 5
<b>Rôle</b>	DON-JUANS & DRAGUEURS OBJECTIF ELYSEF

## COMMODORE 64

<b>Simulation</b>	MACADAM BUMPER
<b>Réflexion</b>	MICRO SAPIENS
<b>Utilitaire</b>	BASIC ETENDU MUSICAL
<b>Wargames</b>	THEATRE EUROPE LA BATAILLE D'ANGLETERRE BATAILLE POUR MIDWAY
<b>Action</b>	5 SUR 5

## THOMSON

<b>Simulation</b>	LE MILLIONNAIRE MISSION DELTA
<b>Utilitaires</b>	VOX TABLO 5 DEBUG
<b>Educatif</b>	LASA
<b>Action</b>	LOMBRIX

## ORIC-ATMOS

<b>Simulation</b>	LE MILLIONNAIRE MISSION DELTA MACADAM BUMPER BUSINESSMAN
<b>Utilitaires</b>	HADES MASTER PAINT
<b>Rôle</b>	DON JUANS ET DRAGUEURS OBJECTIF ELYSEE
<b>Aventure</b>	RENDEZ-VOUS DE LA TERREUR TERMINUS LA CITE MAUDITE S.A.G.A.

## ZX 81

<b>Simulation</b>	INTERCEPTEUR COBALT
<b>Utilitaires</b>	ZX TRI GRAPHIX 81
<b>Domestiques</b>	BUDGET FAMILIAL
<b>Réflexion</b>	CHIROMANCIE
<b>Action</b>	RIGEL 3D FORMULE 1 PANIQUE CASSE BRIQUE SCORPIRUS

Demandez notre catalogue  
détaillé à votre revendeur

**ERE**  
ERE INFORMATIQUE

# GARANTIE

Si, en dépit des soins apportés à la conception et à la réalisation de ce logiciel, celui-ci présentait quelque défaut le rendant inexploitable, remplissez la présente carte de garantie, faites apposer le cachet de votre revendeur, et retournez la cassette dans son emballage d'origine et avec son mode d'emploi à :

**ERE INFORMATIQUE 27, rue de Léningrad 75008 Paris**  
qui vous l'échangera.

NOM .....

Adresse .....

.....

Code Postal ..... Ville .....

Marque de votre ordinateur : .....

Date d'acquisition : .....

Magnétophone utilisé : .....

Nom du logiciel : .....

Date d'acquisition : .....

Problème rencontré : .....

.....

.....

Date : ..... Cachet du revendeur :

27, rue de Leningrad 75008 Paris  
Tél:(1)387.27.27 +