



La correspondance *BASIC* de Pixel



Une lettre qui vous apprend comment mieux programmer votre CPC

Objet : *Programmation structurée sur nos CPC*

Pixel le 20 juillet 2003

Après tant d'hésitation, je décide de rédiger la première lettre d'une longue série de «*la correspondance BASIC de Pixel*». Ainsi au fil des jours, semaines, mois et années avenir, j'espère susciter votre intérêt pour le BASIC dont est équipé votre Amstrad CPC, et par la même occasion, vous faire découvrir les joies de la programmation BASIC structurée. Comme vous l'avez compris chaque lettre vous exposera des astuces, listings, informations...

Bonne lecture & programmation !!!

L'instruction BASIC à l'honneur

La commande «*Print*»

```
Print "Mon texte"
Print a$
Print a
Print chr$(127)
```

Cette commande permet d'afficher du *texte*, la valeur d'une *variable numérique*, la valeur d'une *variable alphanumérique* ou le *code ASCII*.

Astuce :

le symbole ? est l'équivalent de cette commande

Parole d'un CPCsiste

Ben, ici c'est pour publier vos remarques, coups de gueule ou vos encouragements. Une petite blague en attendant.
Quelle est la différence entre un bonhomme de neige et une bonne femme de neige ?

rép. les boules de neiges !!!

Pixel vous annonce

Pour toutes remarques sur «*La correspondance BASIC de Pixel*» vous pouvez me joindre sur mon e-mail à l'adresse suivante :

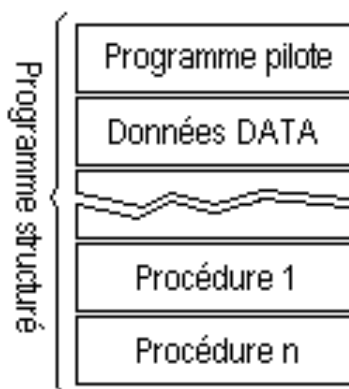
rjean-sebastien@wanadoo.fr



Je programme, tu programmes, nous programmons de façon structurée

Ouf !!! La correspondance est lancée, bon d'accord l'intro n'est pas top mais c'est ma première !!! Bref, vous avez envie de créer des programmes en BASIC, les idées ne vous manquent pas, mais hélas pour concrétiser vous ne savez pas comment vous y prendre. Ne cherchez plus voici la solution (ça fait pub de lessive non !!!). Programmez en BASIC structuré. Mais c'est quoi du BASIC structuré ??? Pour faire simple, la programmation structurée est une façon d'écrire un programme suivant un protocole bien défini. Ce protocole indique qu'un programme structuré est composé généralement d'un ou de plusieurs *programme pilote*, et d'une ou plusieurs *procédure* (petit programme qui réalise toujours la même chose suivant des *paramètres*) Enfin une dernière portion de programme regroupe des données (data en BASIC Amstrad) permettant d'utiliser, et d'exploiter les procédures.

Pour mieux comprendre, regardons le schéma ci-dessous qui montre l'organisation d'un programme structuré



C'est un peu plus clair non !!! Bon qui fait qui, qui fait quoi dans la programmation structurée :

- *Le programme pilote*

Ce programme indique quelles données data doivent être lues, et quelles procédures doivent être exécutées à l'instant t.

- *Les données DATA*

Les données DATA sont des valeurs alphanumériques permettant d'indiquer ce que doit faire la procédure à l'instant t.

- *Les procédures*

Comme expliqué ci-dessus, les procédures sont des programmes qui exécutent toujours la même chose. Comme par exemple dessiner des objets graphiques comme un carré, un rond ou un triangle suivant la donnée DATA lue par la procédure.

Bien c'est joli tout ça mais avec notre bon vieux Basic, comment programme t-on de façon structurée ?

C'est assez simple, Pour exécuter une procédure il suffit d'utiliser la commande BASIC «*GOSUB*», pour rendre la main au programme pilote (la ou il c'était interrompu) on utilise la commande «*RETURN*» pour terminer une procédure. En ce qui concerne la lecture des données (essentielle à l'exécution d'une procédure) on utilise la commande BASIC «*RESTORE*»

Le listing de notre premier programme BASIC structuré

Programme Basic structuré
Pour Amstrad CPC 464,664,6128,464+,6128+
Utilisation d'une procédure graphique permettant l'affichage de formes géométrique (quadrilatère, triangle isocèle)
Programme proposé par Pixel

```
10 REM PROGRAMME PILOTE
20 CLS
30 LOCATE 5,1 :PRINT "Mon premier programme structuré"
40 RESTORE 210
50 GOSUB 510
60 SOUND 1,800,10
70 LOCATE 5,24:PRINT "Une maison"
80 RESTORE 310
90 GOSUB 510
100 SOUND 1,1600,10
110 LOCATE 28,24 :PRINT "Un sapin"
120 GOTO 120

200 REM DONNEES DATA
210 DATA 1,0,100,100,200,200
220 DATA 1,0,110,100,150,180
230 DATA 1,0,160,140,180,180
240 DATA 1,1,80,200,220,230
250 DATA 0,0,0,0,0,0

300 REM DONNEES DATA
310 DATA 1,0,500,100,510,120
320 DATA 1,1,480,120,530,130
330 DATA 1,1,480,130,530,140
340 DATA 1,1,480,140,530,150
350 DATA 1,1,480,150,530,160
360 DATA 0,0,0,0,0,0

500 REM PROCEDURE
510 READ ETAT,FORME,X1,Y1,X2,Y2
520 WHILE ETAT=1
530 IF FORME=1 THEN 560
540 PLOT X1,Y1:DRAW X2,Y1:DRAW X2,Y2:DRAW X1,Y2:DRAW X1,Y1
550 GOTO 570
560 PLOT X1,Y1 :DRAW X2,Y1 :DRAW ((X2-X1)/2)+X1,Y2 :DRAW X1,Y1
570 READ ETAT,FORME,X1,Y1,X2,Y2
580 WEND
590 RETURN
```

avant d'exécuter une procédure. Ainsi vos données seront lues et interprétées par la procédure grâce à la commande BASIC « *READ* »

La programmation structurée est très intéressante par le fait qu'il n'est pas forcément nécessaire d'écrire *les procédures*, si elles existent déjà il suffit simplement de les réutiliser dans vos applications. « *Je suis sûre que certain on déjà commencé à développer des procédures, envoyés-les, afin que la communauté CPCsiste puisse en bénéficier* ».

En fait tout devient simple car finalement il ne vous restera plus qu'à écrire vos programmes pilotes ainsi que les *données DATA* associées.

Alors, vous hésitez ? Voici un programme structuré pour vous convaincre. Pour cet exemple, j'ai écrit une procédure toute simple qui permet d'afficher (suivant *les données DATA*) des objets graphiques ayant la forme d'un quadrilatère ou d'un triangle isocèle. Le *programme Pilote* est très simple, il appelle deux fois la procédure afin de dessiner une maison et un sapin.

Je sais, il n'y a rien d'extraordinaire mais je vais tout de même pas vous donner le listing de mon navigateur, petit os en BASIC que vous allez découvrir dans ces moindres détails. Mais ça, c'est pour plus tard.

Amicalement Pixel

Une info CPC

Je tiens personnellement à féliciter l'équipe *SEMILANCEATE* pour la demo *Fantasy* sur CPC 6128+ qui est vraiment mais alors vraiment superbe.



Bravo à toute l'équipe

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.