

## PRÉCAUTIONS MINIMALES POUR QU'UN LOGICIEL AIT QUELQUES CHANCES D'ÊTRE UTILISABLE PAR D'AUTRES QUE SON AUTEUR

**François BOULE**

1. COMMENCER TOUT PROGRAMME PAR UNE INITIATION DE L'ÉCRAN (par exemple en BASIC : CLS, SCREEN, CONSOLE Ø,24)

2. CONSACRER UNE TOUCHE A TOUT MOMENT (en particulier s'il y a appel de réponse) à une SORTIE du programme qui laisse l'ÉCRAN et la MÉMOIRE aussi propres qu'on les a trouvés en commençant, qui ferme les FICHIERS etc. ; ceci afin d'éviter l'usage intempestif des CNTR C et autres RESET. On peut par exemple consacrer à cela l'une des touches STOP ou RAZ, et indiquer leur usage dans un coin de l'écran.

3. Permettre de court-circuiter les défilements de pages-titres, pages-consignes etc. qui deviennent lassants et/ou inutiles après plusieurs passages, par l'usage d'une touche spéciale (ENTREE p.ex.) ; signaler cette possibilité en indiquant au moment opportun un ENT dans un coin de l'écran.

4. ÉVITER le recours à INPUT (en BASIC) ou à LISLISTE (en LOGO) chaque fois que c'est possible (voir plus loin) et préciser au moins par un ENT quand il est nécessaire de terminer une réponse par ENTREE

5. Éviter le recours au crayon-optique SEUL si possible (ou alors prévoir des zones larges 'et de couleur claire).

6. LIMITER le plus possible le TEXTE à l'écran ; prévoir un minimum d'accompagnement ÉCRIT SUR PAPIER.

7. Préférer les FONDS SOMBRES pour préserver les yeux.

Voici quelques petits utilitaires pour MO5 pouvant contribuer au confort de l'utilisateur

## MAJUSCULES / MINUSCULES

Rien ne permet de savoir, avant de frapper une touche, si le clavier est disposé pour les majuscules ou les minuscules. Pour créer dans un coin de l'écran un témoin de l'état du clavier : ( le signe \_ indique un espace)

```
LOCATE 35,1,0 : COLOR 1,7 : IF PEEK(&H2019)>127 THEN PRINT "-m-"
ELSE PRINT "-M-"
```

Mais il est également possible de "forcer" le clavier en majuscules ou en minuscule, par exemple au moment de demander une réponse ; ce qui permet d'éviter soit un message d'erreur, soit une interprétation permissive de la réponse

Blocage par programme du clavier en minuscules POKE &H2019,128 (les majuscules sont quand même accessibles, grâce à la touche jaune appuyée simultanément)

Blocage en majuscules ;

```
POKE &H2019,0
```

## DURÉE DE. REBOND DU CLAVIER

Si l'on maintient appuyée une touche son effet est reproduit au bout d'une certaine durée ; celle-ci peut être modifiée par programme :

- soit raccourcie, pour obtenir un effet quasi continu (par exemple déplacement d'un objet à l'aide des touches-flèches),
- soit allongée, pour éviter les effets perturbateurs d'un appui prolongé (utilisation par de petits enfants)

```
POKE &H2076, n
```

n est le nombre de dixièmes de secondes entre 2 entrées de signal  
(n= 1 à n=255)

implicitement n=7

## MINI-ÉDITEUR (pour remplacer input)

L'entrée de réponses par INPUT a divers inconvénients elle doit être conclue par ENTRÉE, même si la réponse ne comporte qu'un caractère utile ; si l'on attend un nombre et que l'utilisateur frappe une lettre, l'interpréteur renvoie un "Redo..." assez fâcheux ; la réponse n'est

testée qu'après son entrée, donc après affichage. Pour éviter cela, la réponse est enregistrée par INPUT\$(1) et testée caractère par caractère :

```

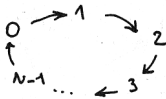
100 CLS : SCREEN 6,0,0
110 PRINT "Donnez votre réponse puis : " ;
115 COLOR 1,3 : PRINT " ENTREE " : COLOR 6,0
120 PRINT "Correction avec <-> " : PRINT : T$ = ""
125 '
130 A$ = INPUT$(1) : A = ASC(A$)
135 IF A=8 THEN IF T$="" THEN 130 ELSE COLOR 0,0 : PRINT
      CHR$(8)+CHR$(32)+CHR$(8) ; : T$=LEFT$(T$,LEN(T$)-1) : COLOR 6,0
140 IF A=13 THEN 155          selon qu'on attend
145 IF (A<48)OR(A>57) THEN 130 un nombre,
145 IF (A<65)OR(A>90) THEN 130 un mot en majuscules
145 IF (A<97)OR(A>122) THEN 130 un mot en minuscules
150 PRINT A$ ; : T$=T$+A$ : GOTO 130
155 ' suite du programme :
      la réponse est dans T$ ; si l'on attend un nombre, convertir par T=
      VAL(T$)

```

La touche-flèche permet de corriger avant entrée définitive. Dans le cas d'une réponse OUI/NON tester sur le code du premier caractère par INPUT\$(1) : 79 ou 199 pour OUI, 78 ou 110 pour NON, ignorer les autres codes.

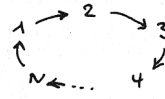
## QUELQUES FONCTIONS SIMPLES

Le SUIVANT d'une permutation circulaire :



suisvant

de Z :  $Z+1-N*INT((Z+1)/N)$



suisvant

de Z :  $Z+1-N*INT(Z/N)$

TIRAGE ALÉATOIRE dans un ensemble

un nombre parmi {1,2,... NI

$X=INT(RND*N)+1$

un nombre parmi (A(1),..., A(N))

$X=A(INT(RND*N)+1)$

**CHOIX ALÉATOIRE DE p OBJETS PARMI n (tirage sans remise)**

On place les objets dans A(1), A(2), ... A(N)

(s'il s'agit de mots, dans A\$(1) etc.)

```
FOR I=1 TO P : K=INT(RND*N)+1 : M=A(k) : A(K)=A(I) : A(I)=M : NEXT I
```

Les p objets cherchés sont alors placés dans A(1), A(2), ... A(P)

## MENU À DÉROULEMENT

La page de titre présente le TITRE du programme et différentes options du menu. En appuyant sur la barre d'espace on distingue une option après l'autre (permutation circulaire) ; pour en choisir une, on appuie sur la touche ENTRÉE :

```

100 CLS : CONSOLE 0,24 : SCREEN 6,0,0
105 READ N : FOR I=0 TO N :READ OP$(I) : NEXT I
115 ATTRB 1,1 : LOCATE (20-LEN(OP$(0))) : COLOR 6,4 PRINT OP$(0)
120 ATTRB 0,0 : FOR I=1 TO N : LOCATE 6,3+2*I PRINT OP$(I) : NEXT I
130 COLOR 6 : LOCATE 2,22 : PRINT "sélection :" ; :COLOR 4,6 : PRINT
  " ESP " ; : COLOR 6,0 :PRINT "_décision :" ; . COLOR 0,3 :PRINT
  " ENT " : I=1
140 LOCATE 6,3+2*I : COLOR 0,7 : PRINT OP$(I)
150 CH$=INPUT$(1) : CH=ASC(CH$) : IF (CH=32)OR(CH=13) THEN 160 ELSE
  150
160 IF CH=32 THEN COLOR 6.4 : LOCATE 6,3+2*I :PRINT OP$(I) I=I+1-
  N*INT(I/N) : GOTO 140
170 'L'option choisie est dans i, suite du programme
110 DATA 4, "_TITRE ","-OPT.1 ","OPT.2 ","_OPT.3 ","_OPT.4"

```

François BOULE