

Périphériques d'entrée (3/3)

par André C.



3ème partie: Deux modes de Saisie avec joystick

Dans la première partie, nous avons vu une routine, pour choisir le périphérique d'entrée à utiliser. Dans la seconde partie, un mini-menu piloté par le joystick a été présenté. Il illustre comme faire l'équivalent du GET Basic, qui attend une réponse avant de continuer. Mais le programme peut aussi saisir une réponse "au vol" et continuer que l'on appuie ou non sur une touche du joystick. L'équivalent de KEY\$ en quelque sorte. Nous allons donc voir dans le détail ces deux modes de saisie "avec attente" et "au vol".

Saisie "au vol" avec joystick

C'est bien simple: Par nature, l'instruction USR(0) retourne "au vol" le code de la dernière touche appuyée sur le joystick. C'est tellement vrai et instantané, qu'il faut trouver des astuces émuler un GET (donner l'impression figer le programme), qui attend une réponse sans se décourager...

La syntaxe la plus simple pour saisir une réponse "au vol" avec le joystick est la suivante: J=0:WAIT 10:J=USR(0) Il ne reste plus ensuite qu'à tester s'il y a eu une réponse (J ≠ 0) et laquelle (selon tableau Ceo-Mag n°254 page 15). Le WAIT instaure un petit délai, afin que l'utilisateur ait le temps de répondre.

Exemple de saisie "au vol"

Pour faire simple, la lettre "A" va se déplacer au bas de l'écran, de gauche à droite et de droite à gauche, indéfiniment. Si une touche quelconque du joystick est pressée, un missile sera alors tiré de bas en haut. Et la promenade de la lettre "A" reprendra, indéfiniment.

La routine "Joystick au vol"

```

100 \ Debut du prg
110 CLS:POKE#26A,2
120 \...
890 \ Promenade
900 X=2:ST=1
910 PLOT X,25,"A"
920 J=0:WAIT 10:J=USR(0)
930 IF J<>0 THEN GOSUB 2000
940 PLOT X,25," "
950 X=X+ST
960 IF X=29 THEN ST=-1
970 IF X= 2 THEN ST=+1
980 \ ...
1990 GOTO 910
1999 END
2000 \ Tir
2010 FOR Y=24 TO 1 STEP -1
2020 PLOT X,Y,"|"
2030 PLOT X,Y," "
2040 NEXT:RETURN

```

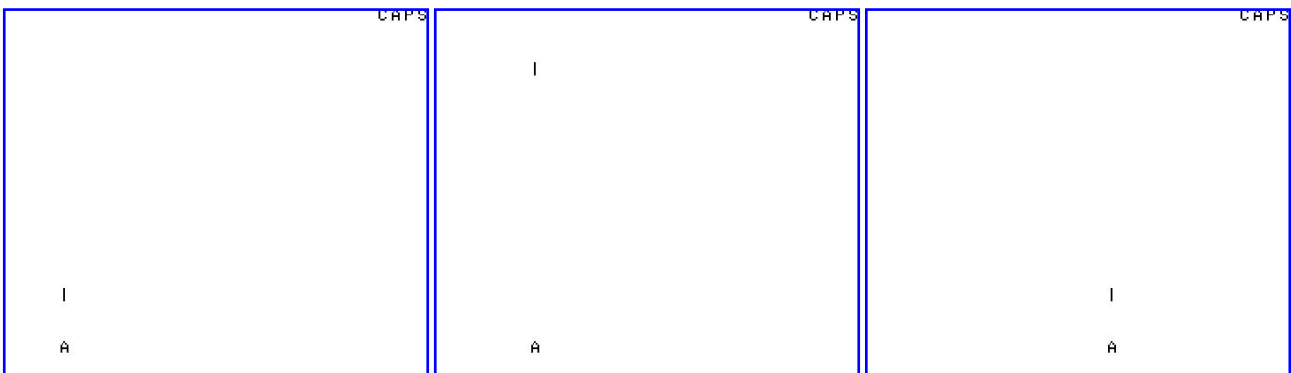
Commentaires sur ce listing

Ligne 110: Ecran vide & pas de curseur

Lignes 900-1990 Corps du programme, dont on ne voit qu'une partie qui consiste à promener la lettre "A" dans le bas de l'écran. Pour alléger ce mini-programme, j'ai renoncé à redéfinir cette lettre.

Ligne 900: Initialisation de l'abscisse de "A" et de la l'incréméntation.

Ligne 910: Affichage de "A" sur la ligne y=25.



Lignes 920-930: Lecture "au vol" du joystick. Si aucune touche n'a été enfoncée, le corps du programme poursuit sa course à la ligne 940. Si une touche quelconque a été enfoncée, on fait un petit détour par le sous programme "Tir" (lignes 2010-2040).

Ligne 940: Effacement de "A", incrémentation de X pour la prochaine position de "A". Si le bord droit de la trajectoire autorisée a été atteint, l'incrémentation devient négative. Au contraire, si c'est le bord gauche, l'incrémentation redevient positive.

Ligne 1990: Après une suite imaginaire du programme (descente d'ovnis où tout ce que vous pouvez imaginer), on reboucle au début (affichage de "A").

Lignes 2010-2040: Sous programme "Tir". La lettre "A" envoie un missile de bas en haut.

Exemples de saisies "avec attente"

Dans tous les cas imaginables, on place le J=USR(0) dans une boucle courte, jusqu'à ce que J soit différent

```
100 CLS:POKE#26A,2:PAPER 3:INK 4
110 FOR X=11 TO 20
120 READ C:PLOT X,10,C
130 NEXT
140 X=11:Y=11:PLOT X,Y,94:' '^"
150 J=0:WAIT 33:REPEAT:J=USR(0):UNTIL J<>0
160 IF J=32 THEN 250:' Select
170 IF J>2 THEN 150:' Reponse non valide
180 PLOT X,Y,32:' " " pour effacer la fleche
190 IF J=1 THEN X=X+1:' Droite
200 IF X>20 THEN X=20
210 IF J=2 THEN X=X-1:' Gauche
220 IF X<11 THEN X=11
230 PLOT X,Y,94:' '^"
240 GOTO 150
250 C=SCRN(X,Y-1):PLOT 11,13,C:GOTO 150
260 DATA 48,49,50,51,52,53,54,55,56,57
```

Commentaires sur ce listing

Ligne 100: Ecran vide, pas de curseur, écriture en bleu sur fond jaune.

Lignes 110-130: Lecture dans les DATA et affichage des caractères de 0 à 9.

Ligne 140: Affichage sous les chiffres du pointeur déplaçable "↑".

Ligne 150: Lecture "au vol" du joystick.

Ligne 160: Si la touche Select a été pressée pour choisir un chiffre, suite en 250.

Ligne 170: Outre Select, déjà traité, seules les flèches droite (code 1) ou gauche (code 2) sont autorisées.

Ligne 180: Une flèche ayant été touchée, il va y avoir déplacement. On commence par effacer le pointeur aux anciennes coordonnées.

Lignes 190-200 Si la flèche droite a été pressée (code 1), on incrémente l'abscisse X, jusqu'à une valeur maxi de 20.

rents de zéro. On peut donc avoir soit:

```
100 J=0:WAIT 33
110 REPEAT:J=USR(0):UNTIL J<>0
```

ou:

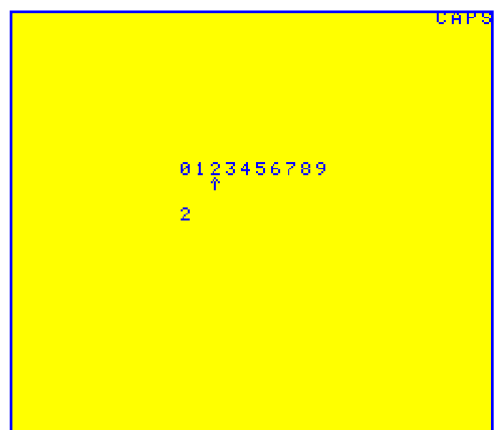
```
100 J=0:WAIT 33
110 J=USR(0):IF J=0 THEN 110
```

ou bien sûr une variante sur le même principe.

Les deux listings publiés dans les Ceo-Mag 254, pages 14-15 et 257, pages 20-21 étaient basés sur la structure IF J=0 THEN rebouclage.

Le WAIT est très important. Si vous avez le doigt un peu lourd, il faudra augmenter ce délai. Avec un délai trop faible, 10 par exemple, on obtient une répétition de la réponse (le programme reboucle plusieurs fois, avant que la touche ne soit relâchée).

Voici un petit listing utilisant une boucle basée sur un REPEAT ... UNTIL:



Lignes 210-220 Si c'est la flèche gauche (code 2), on décrémente X, jusqu'à une valeur mini de 11.

Lignes 230-240: Affichage du pointeur aux nouvelles coordonnées et retour à la lecture du joystick.

Ligne 250: Lecture du caractère pointé et affichage de ce caractère un peu plus bas (on peut imaginer d'autres tâches).

Ligne 260: Codes Ascii des 10 caractères de 0 à 9.

Conclusion

Le principe est tellement simple que vous ne devriez avoir aucun problème à émuler un KEYS ou un GET dans un programme pour joystick.