

Périphériques d'entrée (1/3)

par André C.



1ère partie: Choix du périphérique d'entrée.

Le Super-Oric, comme la Snes, est doté en standard d'un Joypad. Cependant, il est possible que votre Super-Oric fasse partie des privilégiés possédant un clavier. Il est donc judicieux de prévoir en tête d'application un écran permettant de choisir le périphérique d'entrée qui sera utilisé pour cette application. Ceci évitera par la suite d'avoir à tester les deux entrées et donc accélérera le programme.

Le plus simple est d'imaginer un petit menu donnant le choix entre clavier et joypad, la réponse pourrait être captée seulement sur le joypad, puisque tous les Super-Oric en sont pourvus. C'est ce qui a été fait pour MluchSO. Mais il serait peut-être plus élégant de capter aussi bien les réponses clavier que joypad. En effet, intuitivement, celui qui veut utiliser un clavier a tendance à répondre sur le clavier.

Donc impossible d'attendre une réponse comme par exemple avec les commandes GET ou INPUT.

Il faut au contraire capter une réponse au vol comme avec KEY\$. De même, pour le joypad, la recette indiquée dans le Ceo-Mag n°174 page 15, ne marche pas, puisqu'on reste en attente d'une réponse avec une boucle REPEAT ... UNTIL. Enfin, avant de passer à l'écran suivant, il faudra reboucler tant qu'une réponse joypad ou clavier n'aura pas été reçue.

Bref rappel des diverses commandes d'entrée:

A) Pour le clavier:

```
GET R
GET R$
INPUT ["message";] R
INPUT ["message";] R$
R$=KEY$
R=PEEK(#208)
R=PEEK(#209).
```

Chacune de ces commandes a ses avantages et ses inconvénients et doit être utilisée dans un contexte particulier. Notons toutefois que la commande GET R est peut utilisée, car on reçoit une "?Type Mis-

match Error" au cas où une touche non numérique serait frappée. De même pour les deux PEEK qui retournent un code abscons (par exemple #AC pour la flèche gauche ou #A7 pour la touche Shift droit). Enfin les commandes INPUT doivent être suivies d'un ENTER pour valider la réponse.

Restent donc les commandes **GET R\$** ("avec attente") et **R\$=KEY\$** ("au vol") qui sont les plus souples et les plus souvent utilisées.

B) Pour le joystick: Ici le choix se réduit à **R=USR(0)**. Mais il y a tout un contexte d'utilisation à respecter. Tout comme il vaut mieux initialiser au préalable le R\$ de GET R\$ avec un R\$="", ici, il vaut mieux initialiser au préalable le R de R=USR(0) avec un R=0. Ensuite, il faut décoder R selon le tableau suivant:

Touche joystick	Binaire	Décimal
Flèche Droite	1	1
Flèche Gauche	10	2
Flèche Bas	100	4
Flèche Haut	1000	8
Start	10000	16
Select	100000	32
Y	1000000	64
B	10000000	128
R	100000000	256
L	1000000000	512
X	10000000000	1024
A	100000000000	2048

```

100 CLS
110 PRINT@0,12;" Appuyez sur une touche du "
120 PRINT@0,13;" périphérique que vous désirez "
130 PRINT@0,14;" utiliser pour ce programme "
140 PRINT@0,15;" (clavier ou joystick) "
150 R$=KEY$:R=USR(0)
160 IF R$="" AND R=0 THEN 150
170 IF R$<>"" THEN PRINT"Clavier"
180 IF R<>0 THEN PRINT"Joypad"

```

Voilà! En pratique, les lignes 170 et 180 pourront être modifiées pour initialiser un drapeau indiquant le périphérique d'entrée. Par exemple:

```

170 IF R$<>"" THEN PE=1 ELSE PE=0
180 ` PE=0 pour Joypad par défaut et PE=1 pour clavier

```

La saisie "au vol" est délicate, car elle a lieu au moment où la ligne R=USR(0) est exécutée. Par exemple le petit programme:

```

100 PRINT "Touche?"
110 R=USR(0)
120 PRINT R

```

Même si vous êtes Lucky Luke, ce programme retournera invariablement un 0 (les variables étant réinitialisées par la commande RUN).

L'ajout d'un WAIT ne donnera un résultat que si une touche est maintenue appuyée jusqu'à ce que la commande R=USR(0) soit exécutée. Ce n'est pas pratique.

L'utilisation d'une boucle REPEAT ... UNTIL ou d'un test sur du genre IF R<>0 avec rebouclage assure une réponse viable mais revient à attendre une réponse. Il faudra donc la jouer fine pour parvenir à ses fins (sic).

Coup de chance, dans le cas qui nous occupe, nous voulons une réponse avant de passer à la suite du programme. On peut donc imaginer un test du genre:

```

IF R$="" AND R=0 THEN...
avec rebouclage sur la saisie.

```

Voici donc une proposition de routine initiale pour Super-Oric, réduite à sa plus simple expression (il vous est possible de la "décorer").

