

Périphériques d'entrée (2/3)

par André C.



2ème partie: Menu pour joypad

Dans la première partie, nous avons vu une routine, à placer en tête de programme, permettant de gérer un mini-menu pour choisir le périphérique d'entrée à utiliser. Ce mini-menu renseigne une variable PE (périphérique d'entrée) qui prend la valeur PE=0 pour le joypad et PE=1 pour le clavier. Dans la suite du programme, à chaque fois qu'on aura besoin du périphérique d'entrée, il suffira de consulter cette variable PE pour savoir s'il faut appeler un sous-programme joypad ou clavier. Simple donc.

Les commandes Basic du Super-Oric étant les mêmes que celles de l'Atmos, je ne vous ferais pas l'injure de vous rappeler comment on se sert de GET, KEYS\$, etc. Par contre, nous allons voir comment fabriquer un mini-menu joypad, pour choisir entre plusieurs applications. Libre à vous ensuite d'adapter la routine proposée à d'autres usages: Il suffit de vous reporter à l'article précédent (Ceo-Mag n°254

La routine "Menu Joystick"

```

10 S0$="          MENU          "
20 S1$="  Voulez-vous voir la demo: "
30 S2$="    1 - MLUCH             "
40 S3$="    2 - RUSH HOUR v2      "
50 S4$="    3 - COMPTEUR NUMERIQUE "
60 S5$="      Exprimez votre choix "
70 S6$="  en utilisant les fleches "
80 S7$="  verticales, puis Select "
90 CLS:PAPER4:INK3:POKE#26A,10:PV=12
100 PRINT@2,8;S0$
110 PRINT@2,11;S1$
120 PRINT@2,12;S2$
130 PRINT@2,13;S3$
140 PRINT@2,14;S4$
150 PRINT@2,18;S5$
160 PRINT@2,19;S6$

```

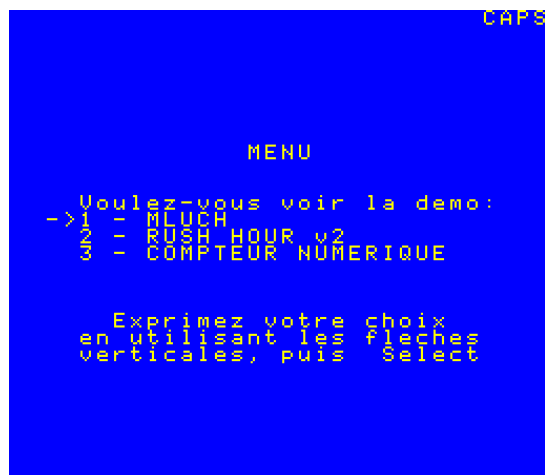
page 14-15) pour y consulter la table où figurent toutes les valeurs retournées par USR(0) en fonction de la touche joystick pressée.

Ce mini-menu vous sera immédiatement utile, car comme vous le savez, il y a beaucoup de place dans une cartouche Super-Oric et plusieurs applications peuvent facilement y trouver place. Par contre, vous ne disposez sans doute pas de plusieurs cartouches "physiques", modifiées pour recevoir une mémoire flash à la place de la Rom Nintendo d'origine, ni sans doute de nombreuses mémoires flash pour recevoir toutes les applications déjà disponibles. Or d'une part, il est un peu pénible de permuter ces mémoires flash et d'autre part les pattes de puces ne vont pas aimer longtemps, même si vous êtes soigneux.

Puisqu'il s'agit d'un mini-menu piloté par le joypad, il sera évidemment judicieux de faire en sorte que la cartouche ne regroupe que des applications fonctionnant avec joypad. Par exemple, Rush Hour devra être mis dans une autre cartouche conçue pour les applications "parlant clavier". En effet, les deux jeux Rush Hour actuels, en LORES 0 et en LORES1 furent mises au point pour valoriser l'interface clavier de Thierry et nécessitent un clavier.

Le cahier des charges

Il est très simple. Nous allons prendre un exemple concret, facilement adaptable à vos besoins. Il s'agit de proposer un choix entre 3 démonstrations: MLUCH, RUSH HOUR v2 et Compteur numérique. Il faudra donc afficher un menu, indiquer comment faire son choix (déplacements et validation), recueillir la réponse et enfin charger la démonstration choisie (qui doit donc être en mode AUTO). Notez que ces nouveaux programmes de démonstrations seront sur la prochaine disquette trimestrielle.



```

170 PRINT@2,20;S7$
180 PRINT@2,PV;"->"
190 R=0:R=USR(0)
200 WAIT 22
210 IF R=32 THEN 290
220 IF R<>4 AND R<>8 THEN 190
230 PRINT@2,PV;" "
240 IF R=4 THEN R=10
250 R=R-9:PV=PV+R
260 IF PV>14 THEN PV=14
270 IF PV<12 THEN PV=12
280 PRINT@2,PV;"->":GOTO 190
290 IF PV=12 THEN CLOAD"ML"
300 IF PV=13 THEN CLOAD"RH2D"
310 IF PV=14 THEN CLOAD"DEMO1"
320 END

```

Commentaires sur ce listing

1) Afficher un menu et indiquer comment faire son choix (déplacements et validation)

Les lignes 10 à 50 définissent le texte du menu (choix proposés).

Les lignes 60 à 80 indiquent qu'il faut utiliser les flèches et la touche Select.

Les lignes 90 à 180 affichent tout ça (de manière bête et méchante, il faut bien le dire). Il serait judicieux de mettre une boucle en place, surtout si votre menu devait comporter plus de lignes. Si vous voulez décorer un peu (couleurs, double hauteur, etc.), reportez-vous aux articles précédents, qui indiquent comment procéder.

2) Recueillir la réponse

Remarquez que la ligne 180 affiche une petite flèche "->", positionnée à la ligne PV (position verticale). Cette variable PV est initialisée au départ à la valeur 12, c'est à dire devant le 1er choix et sera incrémentée ou décrétementée selon vos appuis sur les flèches verticales du joypad.

Les lignes 190 et 200 initialisent la variable R (réponse) et lisent celle de USR(0) qui reflète la touche pressée. Le petit WAIT stabilise la saisie.

Ligne 210: Si la réponse est 32, alors le touche Select a été pressée. On charge l'application correspondante en 290.

La ligne 220 reboucle si une des flèches bas (4) ou haut (8) n'a pas été pressée.

Ligne 230: La flèche "->" est effacée.

Lignes 240 et 250: Petit truc pour faciliter la conversion d'un code abscons en code plus simple: 4 (bas) devient

10, puis 1 tandis que 8 (haut) devient -1. La valeur de la variable PV devient donc facile à ajuster.

Les lignes 260 et 270 évitent que la petite flèche "->" ne sorte du menu, qui s'étage ici des lignes 12 à 14 de l'écran.

La ligne 280 affiche la petite flèche "->" à sa nouvelle position et reboucle pour lire la nouvelle touche pressée.

3) Charger la démonstration choisie

Lignes 290 à 310: Selon la valeur de la variable PV, on charge l'application choisie qui démarre aussitôt. Attention le nom à indiquer est celui qui figure dans l'entête du fichier tap correspondant.

Ligne 320: C'est pour faire propre.

Réalisation

J'ai déjà expliqué maintes fois, la procédure: Taper le texte dans Notepad ou un petit traitement de texte rudimentaire, en mode Ascii. Si vous faites un Copié/collé, vérifiez bien que les guillemets verticaux de MS-DOS ne sont pas convertis en guillemets typographiques (ouvrants et fermants). Il ne reste plus qu'à convertir votre fichier text en fichier tap à l'aide de Txt2bas.exe de Fabrice (syntaxe txt2bas -v1 nom.txt nom.tap). Puis, à l'aide d'un éditeur hexadécimal (genre Winhex, Hex Editor, HexWorshop etc.), de mettre le fichier obtenu en mode AUTO (le 8ème octet de l'entête doit être non nul) et éventuellement d'ajouter un nom de fichier, MENU par exemple, entre les 13ème et 14ème octets (une paire de #00). Tant que vous êtes dans votre éditeur hexadécimal, ajoutez donc à la suite les tap des applications à lancer. Dans notre exemple nous aurons donc, à la queue leu leu, les tap MENU, ML, RH2D et DEMO1, fichier à sauvegarder par exemple sous le nom "demos.tap". Reste à mouliner ce nouveau fichier tap à l'aide de Buildcart.exe (syntaxe buildcart demos), qui vous livrera un joli fichier Demos.smc à utiliser avec un émulateur Snes ou mieux à shooter dans une mémoire flash pour utiliser un Super-Oric réel. Note: le fichier oric2.sys (Rom Super-Oric) doit être présent dans le même répertoire.

Conclusion

Il ne vous faudra que quelques minutes pour adapter cette routine à vos impératifs (nombre et noms des applications) et à la mouliner comme indiqué ci-dessus. J'en ai fait moi-même l'essai, c'est vraiment facile.

