

Mise au Point Software pour Cartouche Super-Oric (2)

par André C.



A l'aide des précieuses informations envoyées par Fabrice F. (voir le Ceo-Mag de septembre pages 7 & 8), je me suis lancé dans quelques petits essais de programmation du Super-Oric. En voici le compte-rendu qui devrait vous aider à programmer vous aussi votre Super-Oric (réel ou virtuel) sans difficultés.

Pour mes mises au point, je suis passé par un émulateur de la console Snes (Snes9x, v1.42 du 4 Décembre 2003, <<http://www.snes9x.com>>, sur les conseils de Fabrice). En effet, l'utilisation de mon Super-Oric réel, nécessite **pour l'instant** de re-programmer la mémoire flash de la cartouche à chaque modification dans le programme. Cette opération, qui s'effectue sur Telestrat (Eh ! J'espère que vous suivez, sinon, reprenez votre pile de Ceo-Mag !) est un peu plus lourde que de tester d'abord sur le Super-Oric virtuel. Je ne l'utiliserai donc que pour finaliser mon programme (il s'agit de Supervex, auquel je n'ai pas travaillé depuis 9 mois, essentiellement parce que mon travail de rédac-chef ne m'en laisse pas le temps).

Le Super-Oric de Fabrice qui est équipé d'un clavier lui permet de taper directement ses essais. Ce que je ne peux pas faire avec le joystick du mien !

Bref, voici comment je procède. Sous Euphoric en configuration Atmos, je tape le programme suivant :

```
100 CLS:PAPER 3:INK 4
110 PRINT@2,10;"*****"
120 PRINT@5,12;"Vive le Super-Oric !"
130 PRINT@2,14;"*****"
140 PRINT@2,20;" "
```

Notez que les étoiles sont au nombre de 28, car l'écran texte du Super-Oric comporte 28 lignes de 32 caractères

(sans marges ni en haut ni à gauche), au lieu de 27 lignes de 40 caractères comme sur l'Oric. J'aurais évidemment pu mettre 32 étoiles, mais pour faire plus joli, je laisse 2 espaces à gauche et à droite). Sur l'Oric, il aurait fallu mettre 36 étoiles pour avoir un effet similaire. Je sauve sous le nom 'SOessai01.tap' en mode 'AUTO'. Comme vous l'imaginez, ce petit programme tourne avec le Basic 1.1 de l'Atmos. Voici ce que ça donne :



Dans la cartouche Super-Oric que Fabrice m'a prêtée, je récupère le Basic 1.1 qu'il a adapté à la console Snes, c'est le fichier 'Base2.bin' (#8000 octets). Histoire de figoler, je remplace l'octet d'offset #7FD7 qui devient #08 au lieu de #0B. Si vous relisez l'article déjà cité, vous pouvez y voir que cet octet indique la taille de la Rom, #08 pour 2Mbits (256 Ko), alors que #0B correspond à 16 Mbits (2Mo). Pour mes petits essais, je me contente d'une petite Rom. Chose curieuse d'ailleurs, l'émulateur que j'ai utilisé (Snes9x) semble se moquer de ce paramètre, puisqu'il accepte n'importe quoi entre #00 et #FF ! Mais pour éviter toute mauvaise surprise ultérieure (version, déboguée, autre émulateur Snes ou Snes réel), je m'astreins à positionner correctement ce paramètre.

Ce fichier (Base2.bin, de checksum #2C6E) ainsi que la version que j'ai modifiée (Base2a.bin, checksum #2C6B) seront sur la prochaine disquette trimestrielle (vous pouvez bien sûr me les demander avant si besoin).

Il ne reste plus qu'à construire la Rom Super-Oric. C'est très simple, mais il faut disposer d'un éditeur hexadécimal. J'utilise Hex Workshop v3.11 du 3 juillet 2000, <<http://www.bpssoft.com>>.

Il faut concaténer (mettre à la suite, bout à bout) le

fichier 'Base2a.bin' et le (ou les) fichiers 'tap' (ici le fichier 'SOessai01.tap'). Enfin ajuster la taille totale du fichier résultant en ajoutant des #00 afin quelle corresponde à la taille indiquée dans l'entête. Ici j'ai indiqué #08 pour 256 Ko (#40000 octets), je complète donc jusqu'à l'offset #3FFFF (les 256 Ko octets vont en effet de l'offset #00000 à l'offset #3FFFF).

Je sauve mon travail sous le nom 'SOessai01.swc' et je charge cette image de cartouche dans l'émulateur Snes9x et boote. Voici ce que ça donne :

```

CAPS
*****
Vive le Super-Oric !
*****
Ready

```

Par curiosité, je modifie le programme pour obtenir :

```

10 LIST
100 CLS:PAPER 3:INK 4
110 PRINT@2,10;"*****"
120 PRINT@5,12;"Vive le Super-Oric !"
130 PRINT@2,14;"*****"
140 PRINT@2,20;" "

```

Je sauve en 'AUTO' sous le nom 'SOessai02.tap', confectionne une nouvelle cartouche virtuelle 'Soessai02.swc' en procédant comme précédemment et teste. J'obtiens :

```

CAPS
ORIC EXTENDED BASIC V1.1
© 1983 TANGERINE

37631 BYTES FREE

10 LIST
100 CLS:PAPER 3:INK 4
110 PRINT@2,10;"*****"
*****
120 PRINT@5,12;"Vive le Super-Oric !"
130 PRINT@2,14;"*****"
*****
140 PRINT@2,20;" "
Ready

```

Notez, que contrairement à ce qu'on a avec l'Oric, ici, il n'y a aucune marge.

Bon, comme mon but est quand même de mettre au point mon petit jeu 'Supervex', pour lequel j'ai besoin d'utiliser le joypad, je fais encore quelques essais dans ce sens et tape le programme suivant :

```

100 CLS:PAPER 3:INK 4
110 PRINT@2,10;"*****"
120 PRINT@5,12;"Vive le Super-Oric !"
130 PRINT@2,14;"*****"
140 PRINT@2,20;" "
1000 REPEAT
1010 J=USR(0)
1020 UNTIL J<>0
1030 PRINT J:J=0:WAIT 50
1040 GOTO 1000

```

Je sauve sous le nom 'SOessai03.tap' et fabrique la cartouche Super-Oric 'SOessai03.swc' que je teste :

```

CAPS
*****
Vive le Super-Oric !
*****
1
2
4
8
16
32
64
128
256
512
1024
2048
4096

```

Explication. Selon la touche pressée sur le joypad, la console retourne une valeur différente pour la variable 'J'. Comme vous le voyez, j'ai un peu 'arrangé' la présentation en ne pressant pas n'importe quoi dans n'importe quel ordre. Reportez-vous aux explications de Fabrice dans le Ceo-Mag n°151, pages 14-17 pour avoir la correspondance entre touche pressée et bit mis à un. En voici d'ailleurs une mise en évidence, avec mon dernier petit programme :

```

100 CLS:PAPER 3:INK 4
110 PRINT@2,10;"*****"
120 PRINT@5,12;"Vive le Super-Oric !"
130 PRINT@2,14;"*****"
140 PRINT@2,20;" "
1000 REPEAT
1010 J=USR(0)
1020 UNTIL J<>0
1021 IF J=1024 THEN PRINT J;" pour X (s)":GOTO 1033
1022 IF J=64 THEN PRINT J;" pour Y (x)":GOTO 1033
1023 IF J=512 THEN PRINT J;" pour L (a)":GOTO 1033
1024 IF J=256 THEN PRINT J;" pour R (z)":GOTO 1033
1025 IF J=2048 THEN PRINT J;" pour A (d)":GOTO 1033
1026 IF J=128 THEN PRINT J;" pour B (c)":GOTO 1033
1027 IF J=32 THEN PRINT J;" pour SELECT (Espace)":GOTO 1033
1028 IF J=16 THEN PRINT J;" pour START (Return)":GOTO 1033
1029 IF J=8 THEN PRINT J;" pour FLECHE HAUT":GOTO 1033
1030 IF J=4 THEN PRINT J;" pour FLECHE BAS":GOTO 1033

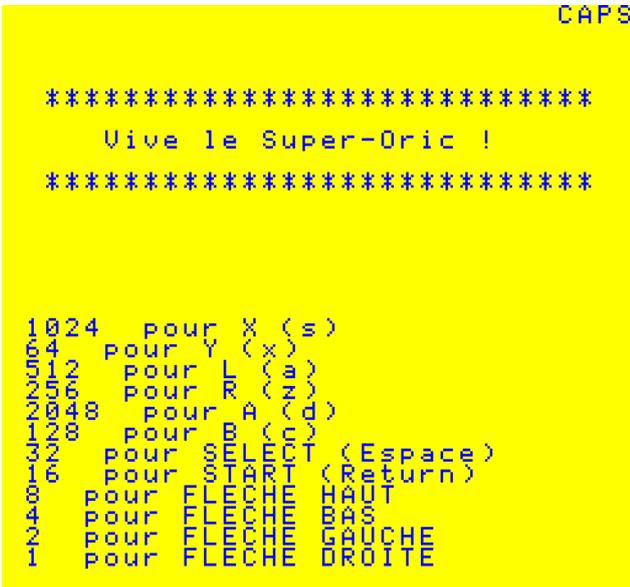
```

```

1031 IF J=2 THEN PRINT J;" pour FLECHE GAUCHE":GOTO 1033
1032 IF J=1 THEN PRINT J;" pour FLECHE DROITE":GOTO 1033
1033 J=0:WAIT 50
1040 GOTO 1000

```

Je sauve sous le nom 'SOessai04.tap' et fabrique la cartouche 'SOessai04.swc' que je teste avec Snes9x et voici ce que j'obtiens :



En fait 'J' est une variable sur 12 bits. Par exemple avec l'émulateur Snes9x, lorsqu'on presse la touche 's' qui correspond à la touche 'X' du joystick, la valeur retournée est 1024 soit 0100 0000 0000 en binaire sur 12 bits (b0 à b11 de droite à gauche). La touche 'émulateur' qu'il faut presser pour obtenir la fonction désirée est indiquée entre parenthèses dans la figure ci-dessus.

NB : Les fonctions 'L' et 'R' sont parfois notées 'TL' et 'TR' pour 'Turn Left' (tourne à gauche) et 'Turn Right' (tourne à droite). Avec Snes9x, ces deux fonctions sont respectivement obtenues en pressant les touches 'a' et 'z' du clavier PC.

En fait pour les bits b6 à b11, je n'ai pas obtenu ce que j'attendais selon le tableau de Fabrice (soit dans l'ordre : 'B', 'A', 'R', 'L', 'Y' et 'X') mais cela est peut-être dû à l'assignation des touches de l'émulateur

Snes ('s' pour 'X', 'x' pour 'Y', 'a' pour 'L', 'z' pour 'R', 'd' pour 'A' et 'c' pour 'B') qui est pour le moins bizarre. Il faudra voir ce que cela donnera sur une vraie console Snes.

C'est tout pour aujourd'hui. La prochaine fois je vous expliquerai comment convertir un écran texte Oric (28 lignes de 40 caractères soit 1120 cases, chiffre qui inclut la 'ligne service' en haut de l'écran) en écran Super-Oric (28 lignes de 32 caractères soit 896 caractères, mais 1792 octets, car chaque caractère utilise en plus de son octet de définition un octet pour les attributs).

Ah ! J'oubliais ! Lors du boot de votre cartouche Super-Oric (réelle ou virtuelle) vous obtiendrez un message d'erreur (voir figure ci-dessus). Ne vous en inquiétez pas, c'est normal, comme expliqué par Fabrice dans l'article de septembre. Lors de la fabrication de notre cartouche, nous n'avons pas renseigné la valeur de la checksum, qui se trouve aux octets d'offset #7FDE-7FDF.

