

Récupérez les caractères redéfinis de tous les logiciels

1 - Méthode

par André C.

Dans les deux articles précédents ("**Récupérez les écrans des logiciels protégés : 1 - Méthode**" et "**2 - Application : Démo Talisman**") nous avons vu comment mettre à profit le dump de la mémoire généré par les émulateurs. Cela s'applique aussi à la récupération des polices de caractères. A priori, on pourrait croire que la procédure ne présente aucun intérêt pour les programmes non-protégés, puisqu'on peut déjà accéder au contenu de la mémoire. Mais, comme vous le verrez, il peut être indispensable d'établir un rapport entre ce qu'on voit à l'écran et ce qu'il y a en mémoire. D'où l'intérêt du dump.

Limites et difficultés

Souvent, les caractères ont été redéfinis en vue de créer une police de caractères (par exemple pour faire un écran-titre). Mais ce n'est pas toujours le cas et même si vous parvenez très facilement à récupérer les jeux 0 et/ou 1, vous ne serez pas forcément au bout de vos peines :

- 1) En effet, les caractères sont souvent redéfinis en vue d'être assemblés pour former des sprites. Il faut donc savoir comment ça s'organise. Solution : Le débogueur de votre émulateur Oric (touche F11 avec Euphoric) vous permet d'examiner l'écran et de comprendre comment sont formés les sprites (avec quels caractères et quelle disposition).
- 2) Autre difficulté, les programmes redéfinissent souvent les caractères au fur et à mesure des tableaux du jeu. Un bon exemple est donné par "The Hellion" (voir les articles "Les 101 écrans de The Hellion" de Simon G. (CEO-mag n°240 à 274). Il faut donc déclencher un dump de la mémoire au moment où les nouveaux caractères à récupérer apparaissent à l'écran (touche F9 avec Euphoric).
- 3) Enfin, si en général tout va bien avec les programmes Basic, les choses se compliquent souvent avec les programmes écrits en langage machine. En effet, les auteurs optimisent forcément leur code et sont amenés à stocker les caractères/sprites dans un format différent du format habituel et d'utiliser des routines

d'affichage spécifiques. Il va sans dire que dans ces cas-là, il faudrait beaucoup investir dans la matière grise...

La méthode de récupération

C'est exactement la même procédure que pour les écrans. Seules les adresses changent.

Comme vous le savez, l'Oric comporte deux jeux de caractères :

- Le jeu 0 dit "normal", situé de #B500 à #B7FF (3 pages de #100 octets), qui compte 96 caractères du code Ascii 32 (espace) au code Ascii 127 (rectangle noir de 6x8 pixels utilisé pour le curseur).
- Le jeu 1 dit "semi-graphique", situé de #B900 à #BAFF (2 pages de #100 octets), qui ne compte que 64 caractères du code Ascii 32 (espace) au code Ascii 95 (rectangle noir de 6x8 pixels, identique au code Ascii 127 du jeu 0).

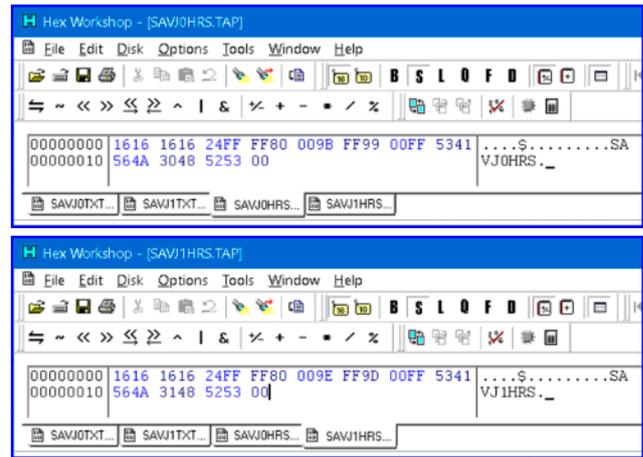
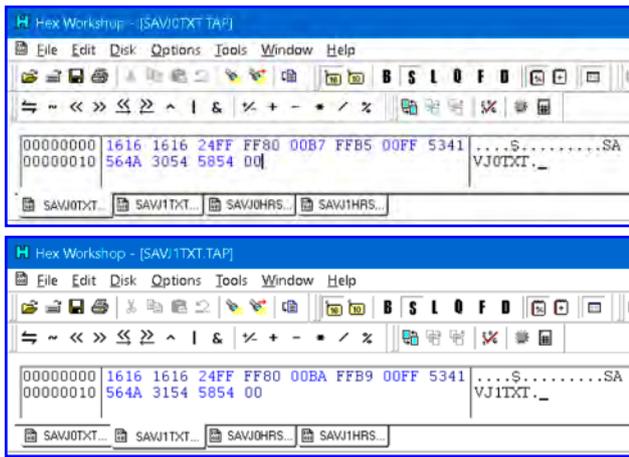
Attention, il s'agit **des adresses initiales**, au boot, c'est-à-dire **en mode TEXT**. Si le programme passe en mode HIREs, les deux jeux sont déplacés respectivement vers la zone #9900 à #9BFF pour le jeu 0 et vers la zone #9D00 à #9EFF pour le jeu 1 (voir le tableau de la page suivante).

Fabriquez les en-têtes appropriés

Pour les 2 jeux de caractères et pour les 2 modes d'écran, il faudra donc prévoir 4 modèles d'en-tête, par exemple avec le petit programme SAUVJEUX suivant (voir le fichier .zip joint à cet article) :

```
100 TEXT
110 CSAVE"SAVJ0TXT",A#B500,E#B7FF
120 CSAVE"SAVJ1TXT",A#B900,E#BAFF
130 HIREs
140 CSAVE"SAVJ0HRS",A#9900,E#9BFF
150 CSAVE"SAVJ1HRS",A#9D00,E#9EFF
160 PING
```

Pour avoir un en-tête proprement dit, récupérez les octets du début du fichier .TAP, allant du #16 initial au #00 (inclus) situé juste après le nom du fichier. Dans les conditions décrites, chaque en-tête aura au final une longueur de 23 octets. A l'issue de ce petit travail, vous disposerez



donc de 4 modèles d'en-tête (cf. figures ci-dessus) que vous garderez précieusement. Les 4 fichiers .TAP correspondants sont dans le zip joint à cet article.

Récupérez les DUMPs

La procédure de récupération des jeux de caractères est très simple. Les consignes qui suivent concernent Euphoric, mais des fonctions analogues sont disponibles avec les autres émulateurs.

- 1) Lancez Euphoric et chargez le logiciel à traiter. Si besoin, lancez l'exécution.
- 2) Guettez le moment où les initialisations semblent terminées et où les caractères à récupérer apparaissent à l'écran et pressez la touche F9 pour générer un dump mémoire.
- 3) Pressez immédiatement la touche F11 pour figer l'exécution du logiciel et renommez le fichier DUMP généré (sinon il serait écrasé au prochain appui sur F9).
- 4) Eventuellement pressez à nouveau la touche F11 pour relancer l'exécution et guettez le tableau suivant. Etc.

Note : Il pourrait aussi être utile de garder des recopies d'écran afin de pouvoir identifier s'il s'agit d'un écran TEXT ou HIRES (pas toujours évident). Profitez également de l'appui sur F11 pour examiner le contenu des zones #BB80-#BFDF (écran TEXT) ou #A000-#BFDF (écran HIRES).

Et enfin, récupérez les jeux de caractères

Il n'est pas toujours facile de déterminer si ce sont les caractères du jeu 0 ou du jeu 1 (ou des 2) qui ont été redéfinis, ni si le programme tourne en mode TEXT ou en mode HIRES. Dans l'expectative, on sera parfois conduit à récupérer plusieurs jeux à l'aveuglette et à juger ensuite des résultats. Dans votre éditeur hexadécimal :

- 1) Ouvrez le modèle d'en-tête correspondant à la

zone à récupérer. CTRL+A puis CTRL+C pour copier les 23 octets du fichier.

- 2) Ouvrez un nouveau document. CTRL+V pour y coller l'en-tête. Editez les 8 caractères de forme "SAVJxxxx" pour renommer votre fichier .TAP
- 3) Ouvrez le fichier DUMP à traiter. Sélectionnez les #0300 (jeu 0) ou #0200 (jeu 1) octets correspondant au jeu ciblé à partir de l'adresse de ce jeu (voir le tableau ci-dessous), puis tapez CTRL+C pour copier la zone sélectionnée.
- 4) Pour vous éviter de sauvegarder inutilement un jeu de caractères qui n'a pas été redéfini, demandez à votre éditeur hexadécimal de calculer la checksum de la zone sélectionnée et comparez avec la valeur indiquée dans le tableau ci-dessous (valeurs pour les jeux natifs d'origine Oric).
- 5) Allez dans le nouveau document, à la fin de l'en-tête, CTRL+V, pour y coller le jeu.
- 6) Sauvegardez votre jeu au format .TAP par exemple sous le nom TOTOJ0.TAP.
- 7) Examinez le jeu sauvegardé à l'aide de l'un des nombreux programmes de redéfinition ou de visualisation pour Oric ou à l'aide du programme OricExplorer de Scott Davies.

Type de jeu	Mode écran	Aller à l'offset	Nombre d'octets	Checksum Atmos	Checksum Sedoric
Jeu 0	Text	#B500	#300	#3F70	#4026
Jeu 1	Text	#B900	#200	#3F00	#3F00
Jeu 0	Hires	#9900	#300	#3F70	#4026
Jeu 1	Hires	#9D00	#200	#3F00	#3F00

La checksum du jeu 0 sous Sedoric est différente car 6 caractères ont été redéfinis : à ç é ù è et ê. Dans la deuxième partie de cet article, je prendrai deux exemples, "Rush Hour" et "The Hellion", pour illustrer pas à pas la procédure de récupération des jeux de caractères. à bientôt... ■