

Banc d'essai : Les éditeurs de caractères (3/5)

Les programmes de Bertrand Brocard et d'André Chéramy par André C.

Voici le troisième volet de cette petite série d'articles concernant les éditeurs de caractères sur Oric. Le programme de Bertrand Brocard est un programme commercial. Il offre des possibilités très intéressantes. Le mien fait pâle figure en comparaison! Pourtant, il est simple à utiliser et c'est l'un des rares logiciels de redéfinition à permettre de déplacer des caractères dans un même jeu ou d'un jeu à l'autre.

Caractor de Bertrand Brocard (© 1983 ARG informatique)

Quelques infos : On trouve ce programme sur oric.org <<http://www.oric.org/software/caractor-843.html>> et sur mon site <<http://andre.cheramy.net/telechargement/Programmes/Redefcar.zip>>. Il est aussi sur la disquette mensuelle accompagnant cette série d'articles. Ce programme comporte un seul fichier CARACTOR.COM de 47 secteurs. Aucun article ne lui a été consacré dans le Ceo-Mag.

Merci à Bertrand Brocard, qui nous autorise à diffuser les logiciels de Cobra Soft / ARG Informatique. C'est la version transférée "sans erreur" de Simon G. qui est en ligne et qui est testée ici.

Il s'agit d'un programme en Basic pour Oric-1 et Atmos, vendu en cassette avec un manuel. Il peut sans problème être transféré sur disquette. Cet utilitaire est assez intéressant, dans la mesure où il permet en plus de la redéfinition de caractères proprement dite, d'assembler plusieurs caractères pour former un petit dessin plus complexe, de type "sprite". En outre, il permet de réaliser des micros dessins animés. N'ayant pas sous la main le manuel de Caractor, mes tests se sont effectués à tâtons, ce qui n'est pas plus mal, car cela exclu tout ce qui n'est pas explicite à l'écran.

Le programme démarre avec un écran-titre animé et défilement du copyright (Figure 1).

Puis un écran d'info et un menu sont affichés (Figure 2). Au Menu on trouve (Figure 3) :

1) Redéfinition d'un caractère

Un caractère à la fois, mais on peut évidemment en redéfinir autant qu'on veut, pourvu que ce soit dans le jeu n°0 "Standard". Il n'y a pas d'accès au jeu n°1 "Alterné" (ou graphique). C'est une lacune. Après avoir choisi un éventuel modèle (pratique pour faire une série de caractères pour une animation), il faut choisir au clavier le caractère à redéfinir (Figures 4 & 5, page suivante). Deuxième limitation donc, puisque les caractères non présents sur le clavier sont exclus. Les commandes disponibles sont affichées dans le bas de l'écran, mais de manière pas toujours explicite (manuel manquant) (Figure 6, page suivante).

Atout positif : L'édition des points ne se fait pas par bascule d'une unique touche : ici un pixel est allumé avec "espace" et éteint avec "B". Ce n'est pas plus lent et cela génère moins d'erreurs. Autre point positif, le curseur n'est pas un carré plein, mais une croix fine : résultat, il ne cache pas le dessin (ça c'est top !).

Le dessin s'effectue dans un buffer et n'est validé (mis à la bonne place en Ram) que si on passe à la redéfinition d'un autre caractère avec "C" ou que l'on termine la session d'édition avec "F" (Figure 7, page suivante). Il n'est donc pas mis



Fig. 1 : Ecran titre

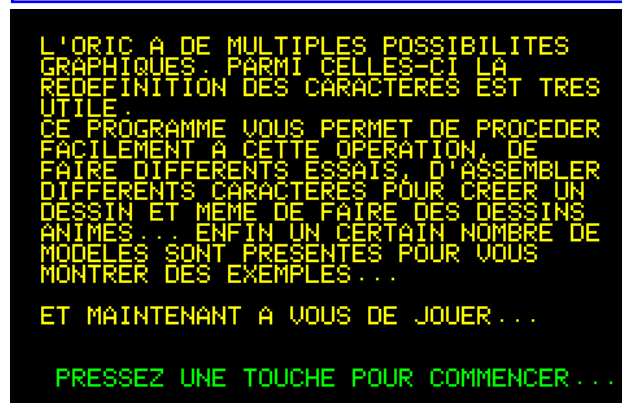


Fig. 2 : Introduction

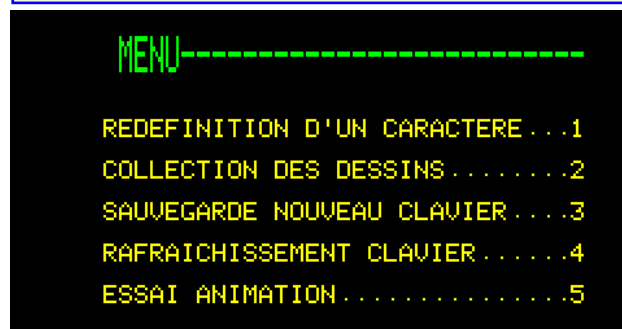


Fig. 3 : Menu

à jour au fur et à mesure de l'édition, comme avec certains autres programmes de redéfinition. Et c'est peut-être mieux ainsi (*), mais cela a un petit inconvénient, car il n'y a aucun affichage en taille réelle du caractère en cours de redéfinition. Après validation du caractère (passage en Ram), celui-ci est affiché sous forme de frise. Mais si l'on est en train de redéfinir un caractère qui n'est pas présent à l'écran, il est bien difficile de savoir ce que donne vraiment la redéfinition. (*) Voir plus bas avec l'affichage de la collection de "sprites" qui n'affecte pas le jeu de caractères en Ram.

Attention, la touche "0" de remise à zéro efface complètement le caractère en cours d'édition (mise à zéro du buffer) et affiche une grille vide.

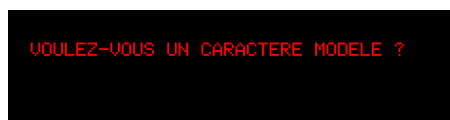


Fig. 4 : Première option



Fig. 5 : Seconde option



Fig. 8 : Retour au menu



Fig. 6 : Edition du caractère



Fig. 7 : Validation de la redéfinition



Fig. 9 : Collection de caractères redéfinis et de "sprites".

Je n'ai pas réussi à comprendre la commande "X" de déplacement (elle semble inactive). Peut-être s'agit-il de copier un caractère dans un autre ? Là, le manuel semble indispensable. Le retour au menu s'effectue avec la touche "M" (Figure 8, page précédente).

2) Collection des dessins

Il s'agit d'une collection très intéressante comprenant des caractères redéfinis : carreau, cœur pique, trèfle, notes de musique, bombe, boule, poisson, poussin, bonhommes, caractères accentués (inexistants en 1983) (Figure 9, page précédente) et des "sprites" composés de 2 à 10 caractères : ciseaux & pointillés, chameau, chat, chiens, avions & bombes, char d'assaut, voilier, locomotive, robot, clef, soucoupes volantes, avions, chimère, dragon de feu (Figure 9, page précédente). Cette collection de dessins contient plus de 20 exemples très ingénieux.

Je n'ai pas réussi à comprendre s'il était possible d'éditer plusieurs caractères pour former un "sprite", ni a fortiori comment enregistrer ces sprites avec un nom descriptif (pour augmenter la collection) (Figure 10). Une fois de plus, il serait nécessaire de consulter le manuel. Cette collection n'est peut-être qu'une sorte de démo pour montrer ce qu'il est possible de faire. Pour l'enrichir, il faudrait sans doute jeter un coup d'œil du côté du programme Basic.

Chacun de ces exemples est formé de 1 à 10 caractères. Ces caractères sont indiqués sous la mention "CASES" et leur disposition est celle qu'il faut utiliser pour recomposer le "sprite". On comprend que ces caractères redéfinis ne soient pas validés dans le jeu de caractères normaux en Ram. En effet, les caractères indiqués sous la mention "CASES" ne pourraient alors pas être identifiés (brouillés par la redéfinition). Les figures ci-jointes montrent ce qui se passe lorsqu'un caractère, ici le "A" (précédemment redéfini sous forme d'un petit bonhomme) est validé en Ram. Il est heureusement possible de récupérer les octets de redéfinition des caractères formant les "sprites", grâce au petit menu présent en bas à droite de l'écran (Figure 11). Notez que dans l'exemple ci-dessous, le caractère "A" a été précédemment redéfini (octets 13, 13, 63, 44, 44, 18, 18, 51) et que les octets présentement affichés sont ceux du caractère du "sprite" (octets 0, 0, 4, 14, 31, 31, 14, 4) (Figure 12). La première redéfinition du caractère "A" est celle présente dans le jeu de caractères en Ram et la seconde redéfinition est celle du caractère "A" dans le buffer de redéfinition du "sprite".

3) Sauvegarde nouveau clavier

On se serait plutôt attendu à une "Sauvegarde du jeu standard" de caractères. Ces caractères sont sauvés en copiant la Ram de #B400 à #B800 sous le nom immuable de CLAVIER (attention aux écrasements) (Figure 13). Le programme original était commercialisé sur cassette et la version testée étant identique à cet original, la sauvegarde se fait toujours sur cassette (Figure 14, page suivante). Mais le fichier obtenu peut facilement être récupéré sur disquette par un SAVE"CLAVIER",A#B500,E#B7FF (ces adresses sont plus exactes).

Le problème est que le menu omet de proposer une relecture des fichiers sauvegardés ! Il faut donc penser à les charger en mémoire avant de lancer CARACTOR (Figure 15).

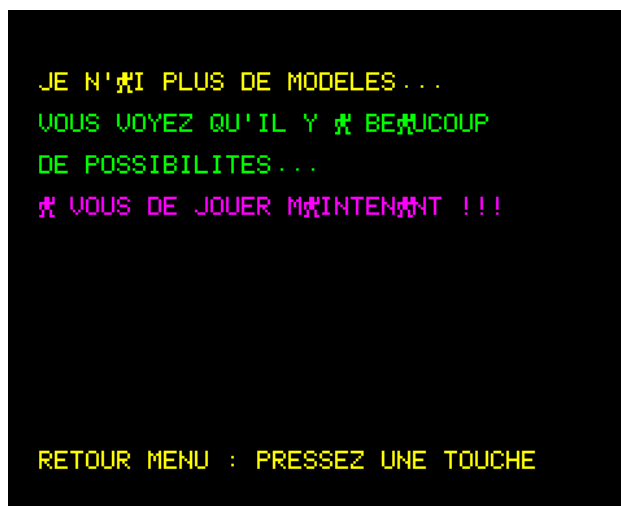


Fig. 10 : Avis de fin de la collection

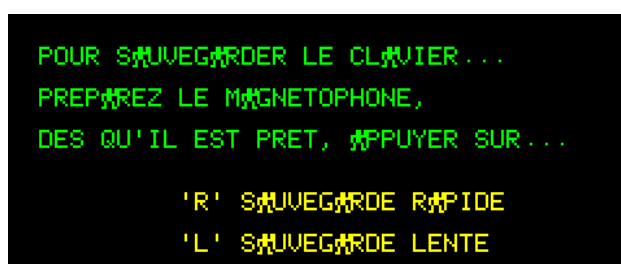


Fig. 11 : Menu de sauvegarde

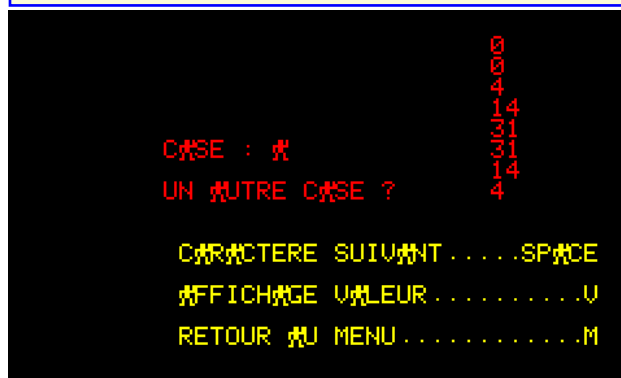


Fig. 12 : Affichage des octets de redéfinition

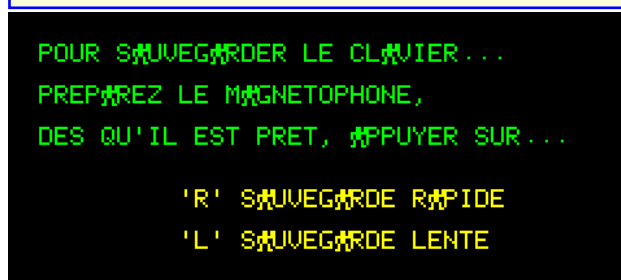


Fig. 13 : Sauvegarde sur cassette

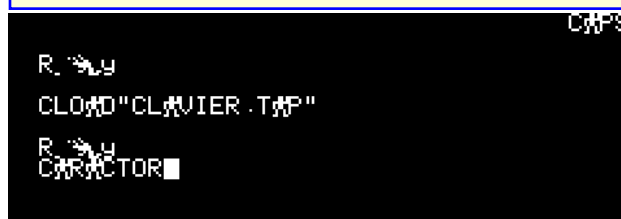


Fig. 15 : Relecture manuelle

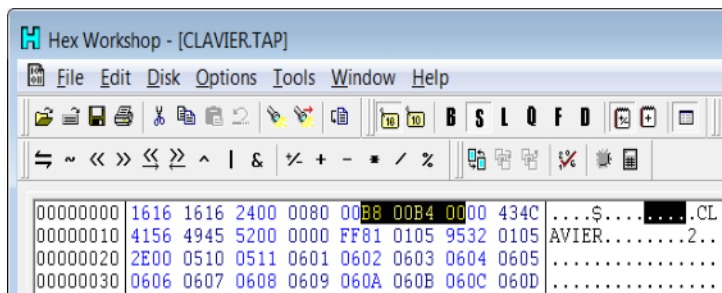


Fig. 14 : Le début du fichier CLAVIER.TAP

4) Rafraîchissement clavier

Toujours cette curieuse appellation. L'ensemble de jeu de caractères standard devrait être remis dans son état normal par copie de l'original en Rom dans la Ram. Mais, il y a une bogue et le programme plante... (Figure 16).

5) Essai animation

Apparemment il est possible de faire une petite animation (9 images maximum) et de la visualiser (Figure 17). Il faut d'abord redéfinir un à un les caractères qui composeront cette animation. Puis le programme permet avec cette option de tester l'effet produit.

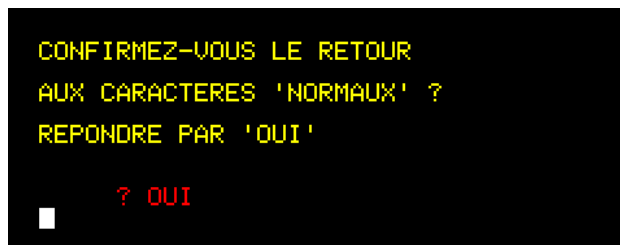


Fig. 16 : Plantage



Fig. 17 : Option mini-animation

Conclusion : C'est un programme qui sort de l'ordinaire, mais qui demanderait à être repris, débogué et amélioré. Il offre des options originales et possède des caractéristiques intéressantes. Dommage qu'elles soient bridées par de petites imperfections.

REDEFMOVC d' André Chéramy

On trouve ce programme sur la disquette Redefmovc.dsk (disquette trimestrielle de mars 2003), sur oric.org <<http://www.oric.org/software/redefmovc-2333.html>> et sur mon site : <<http://andre.cheramy.net/telechargement/Programmes/Redefcar.zip>>. Il est aussi sur la disquette mensuelle accompagnant cette série d'articles. Un article lui a été consacré dans le Ceo-Mag n°154, pages 41-45. Il comporte 4 fichiers (REDEFMOVC.COM, REDEFMOVC.HLP, REDEFMOVC.DEM et REDEFMOVC.CHS) soit un total de 40 secteurs.

Comme son nom l'indique (mal), ce programme a 2 fonctions : Redéfinir un ou plusieurs caractères et déplacer un ou plusieurs caractères à l'intérieur d'un jeu ou d'un jeu à l'autre. Un menu initial permet de choisir entre ces 2 fonctions (Figure 18). Il y a un défaut dans ce menu: l'ordre des choix aurait dû être :

- 1) Instructions (I)
- 2) Redéfinition de caractères (R)
- 3) Déplacement de caractères (D)
- 4) Fin (F)

En effet, même si le programme est auto-documenté, il vaut mieux commencer par lire le court écran d'aide affiché par l'option "Instructions". Autre défaut d'ailleurs, il devrait être possible de consulter cet écran en cours d'édition ou de déplacement de caractères, ce qui n'est pas le cas.

Cet utilitaire est écrit en Basic, mais cela ne nuit ni à son temps de réaction, ni à la pérennité des caractères redéfinis qui peuvent être sauvegardés sur disquette, même après avoir quitté le programme.

a) Instructions

Il est vivement conseillé de lire l'écran d'instructions (Figure 19). En effet, il existe des trous dans l'auto-documentation du programme.

b) Redéfinition de tous les caractères des jeux n°0 (codes 32 à 127) et n°1 (codes 32 à 95).

Le programme est auto-informatif : la procédure à suivre (simple) s'affiche au fur et à mesure des besoins. La touche

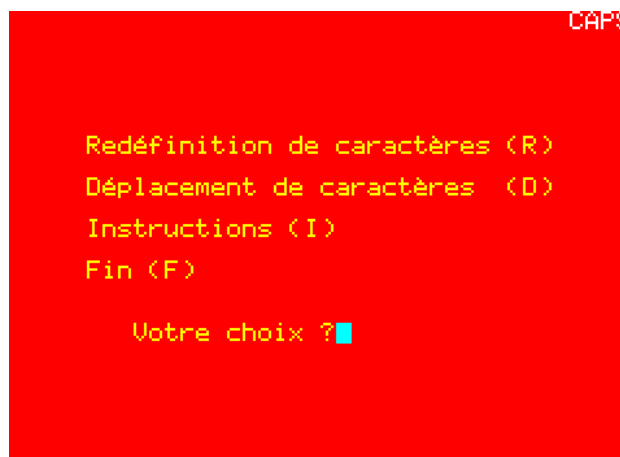


Fig. 18 : Ecran accueil

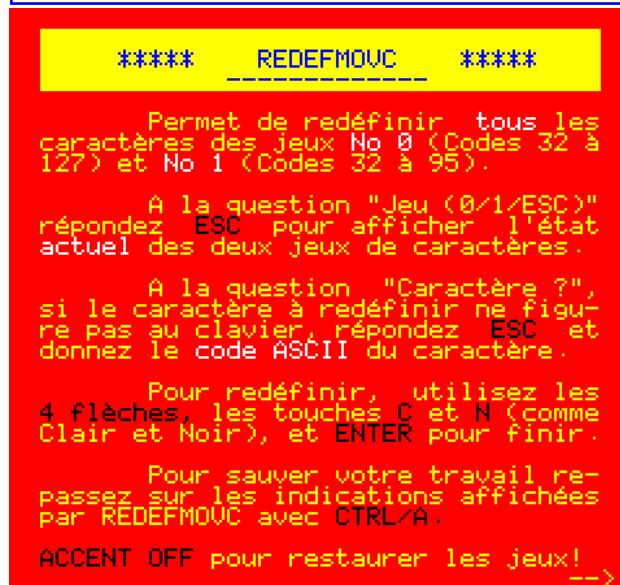


Fig. 19 : Informations

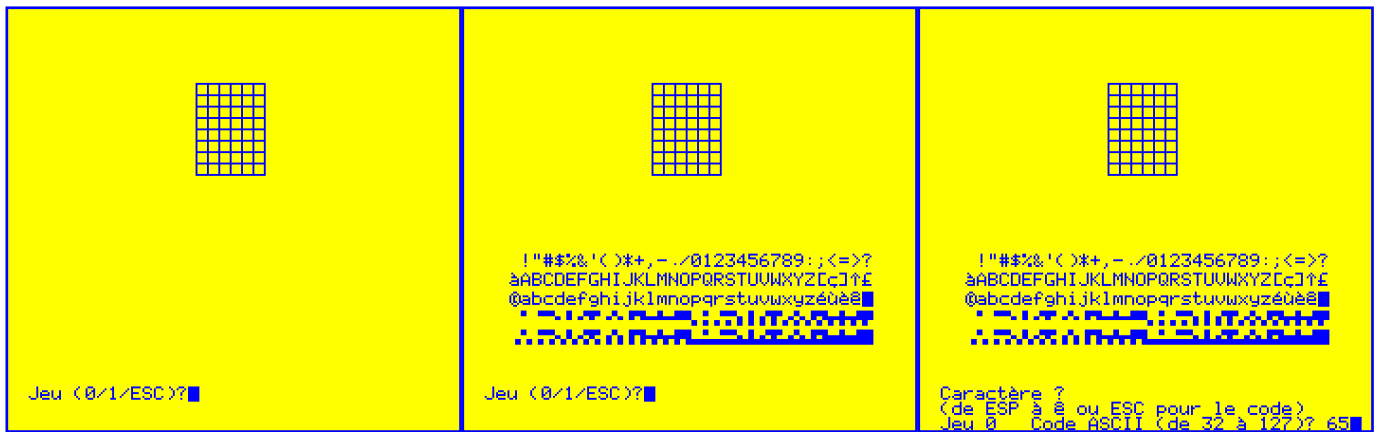


Fig. 20 : Choix du jeu : 0 (standard) ou 1 (alterné) et du caractère à redéfinir

"ESC" affiche l'état actuel des 2 jeux de caractères. Le choix du jeu de caractères se fait avec les touches "0" (jeu normal) ou "1" (jeu alterné). Il faut ensuite indiquer le caractère à redéfinir (au clavier ou code Ascii) (Figure 20).

Le caractère choisi s'affiche dans la grille de 6x8 au centre de l'écran et également sur la gauche en grandeur nature (Figure 21 gauche). L'édition du caractère se fait de manière simple. On noircit une case avec "N" tandis que "C" permet d'avoir une case claire. Les 4 flèches servent aux déplacements du curseur, lequel clignote. A cet égard, le choix des couleurs d'encre (bleue) et de fond (jaune) n'est pas très heureux, même si cette combinaison est lisible et esthétique. En effet, l'inverse vidéo du bleu est le jaune et réciproquement. Il en résulte qu'à cause du clignotement (réalisé par inversion vidéo intermittente), on ne peut savoir si le curseur se trouve sur une case bleue ou sur une case jaune. Autre imperfection, quand on a terminé la redéfinition du caractère (Figure 21 milieu), que faire pour sauvegarder ? La réponse est : presser la touche "RETURN", mais ce n'est pas indiqué et si vous avez lu les instructions (au menu initial), il y a des chances que vous l'ayez oublié.

Cette partie du programme reboucle sur elle-même, ce qui permet de redéfinir autant de caractères qu'on le désire. A chaque tour de boucle une sauvegarde du caractère est proposée sous forme de DATA (Figure 21 droite). On peut se contenter de noter l'adresse en Ram du caractère pour utilisation ultérieure (ça c'est une exclusivité parmi les programmes testés et pourtant, c'est une information essentielle). Quand tous les caractères nécessaires auront été redéfinis, le programme propose de sauvegarder l'ensemble du ou des jeux édités (Figure 22). Il est bien sûr possible de faire une sauve-

garde partielle en utilisant les adresses notée précédemment. L'écran propose aussi de remettre les caractères dans leur état d'origine (reset des caractères) et cela touche tous les caractères. Tant qu'on n'utilise pas cette possibilité, les caractères redéfinis restent tels quels en Ram, même après avoir quitté le programme. Enfin, le programme propose de retourner au menu initial (par exemple pour effectuer des déplacements) ou en répondant "N" de retourner au "Ready".

Note importante : C'est seulement après avoir fait le choix "N" et être retourné au "Ready" qu'il est possible d'exploiter les commandes affichées et ceci en repassant dessus à l'aide de CTRL+A. Si l'on choisit la réponse "O", ces commandes seront à nouveau proposées plus tard. En tout état de cause, même si l'on sort du programme (accidentellement par exemple), rien n'est perdu : un nouveau "RUN" permet de reprendre sans problème le travail en cours.

c) Déplacement de tous les caractères des jeux n°0 (codes 32 à 127) et n°1 (codes 32 à 95).

Cette seconde partie du programme est également auto-informative et fonctionne un peu comme la première partie, sauf que cette fois, il s'agit de déplacer des caractères. Les 8 octets définissant le caractère source sont copiés à la place des 8 octets du caractère cible. En fait, il s'agit plus d'une duplication que d'un déplacement.

Il faut choisir le jeu de départ ("0" ou "1") ou "ESC" pour voir les jeux, puis le caractère source (touche de ce caractère au clavier ou "ESC" pour entrer son code Ascii). Idem pour le caractère cible, "P" dans l'exemple illustré (même jeu) (Figure 23 gauche). Après avoir validé, on voit immédiatement le résultat en haut de l'écran : dans le premier exemple utilisé, le "P" de "CAPS" est identique au A (même redéfinition) (Figure 23 milieu). Le se-



Fig. 21 : Redéfinition du caractère "A" et affichage des 8 octets pour les DATA d'un programme Basic

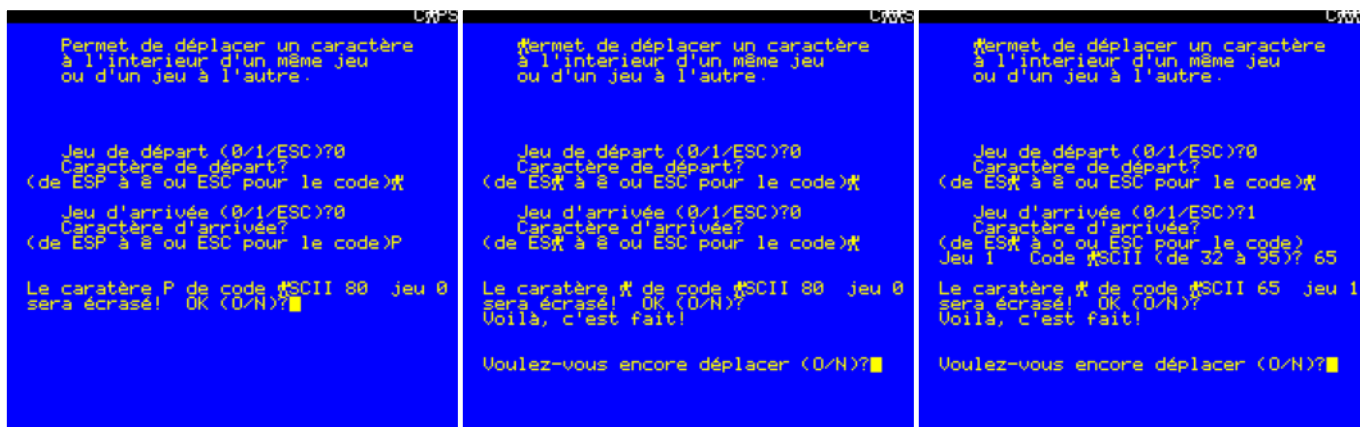


Fig. 23 : Déplacement d'un caractère à l'intérieur d'un même jeu et d'un jeu à l'autre

cond exemple montre la copie du "A" redéfini dans le caractère de code Ascii 65 du jeu alterné (Figure 23 droite). Encore un petit défaut : pour visualiser le résultat, il faut faire comme si on voulait encore déplacer et répondre "ESC" à la première question. Le programme n'a pas prévu de détecter si on cherche à déplacer (copier) un caractère sur lui-même et donc pour sortir de cette boucle, il suffit d'indiquer la même cible que la source (ça non plus, ce n'est pas indiqué !) (Figure 24 gauche).

Cette partie reboucle sur elle-même tant qu'on a des caractères à déplacer. Une sauvegarde des jeux de caractères est proposée à la fin. A la question "Voulez-vous encore déplacer" (Figure 24 gauche), il faut répondre "N". Un écran de sauvegarde s'affiche alors (Figure 24 milieu). Pour utiliser les informations affichées, il faut encore répondre "N". Ce n'est que lorsque le "Ready" est affiché que l'on peut utiliser le CTRL+A pour valider les options affichées à l'écran (sauvegarde de jeu standard, sauvegarde de jeu semi-graphique et régénération des caractères Oric d'origine).

La ligne permettant de remettre les caractères dans leur état d'origine (reset des caractères) touche tous les caractères des 2 jeux. Tant qu'on n'utilise pas cette possibilité, les caractères redéfinis restent en l'état, même après avoir quitté le programme.

Notez enfin, que ce système de "déplacement" de caractères peut être utilisé dans plusieurs buts, selon les besoins. On peut par exemple "sauvegarder" les caractères du jeu n°0 qui vont être redéfinis dans le jeu n°1 ou copier un caractère dans un autre que l'on va éditer à son tour et ainsi de suite, en vue de faire une mini animation, comme avec Caractor de Bertrand Brocard.

c) Composition du pack Redefmovc.

La disquette Redefmovc.dsk contient 4 fichiers : REDEFMOVC.COM est le programme Basic proprement dit (#0501 à #15F9).

REDEFMOVC.HLP est un écran d'aide, appelé par le programme (#BB80 à #BFDf).

REDEFMOVC.CHS est un jeu de caractères redéfinis dans le jeu n°1 (graphique) pour construire des cadres ou des tableaux (Figure 24 droite). Le jeu n°1 Oric comporte 64 caractères de Ascii 32 à Ascii 95, soit 512 octets (64x8=512) occupant donc #200 octets de #B900 à #BB00. Dans le cas présent, ce sont les 44 caractères Ascii 33 à 76 qui ont été redéfinis, il reste donc 20 caractères disponibles pour redéfinir des caractères complémentaires.

REDEFMOVC.DEM est une démo affichant un exemple d'utilisation des caractères redéfinis ci-dessus. Ce fichier comporte 3 parties mergées : Une partie Basic, le jeu de caractères Redefmovc.chs et un écran texte tout prêt. La démo affiche deux fois le même "dessin", dans la moitié supérieure de l'écran et dans la moitié inférieure de l'écran. Mais l'affichage du haut se fait avec le jeu n°1 (graphique) et l'affichage du bas avec le jeu n°0 (normal).

Conclusion : Comme indiqué au cours de cet article, le programme Redefmovc souffre de quelques imperfections. Malgré tout, sa simplicité d'utilisation et la possibilité de "déplacer" des caractères couvrent toutes les demandes usuelles en pratique. Pour le reste (symétries, inversion vidéo, etc.) il faut faire appel à des programmes plus à suivre...



Fig. 24 : ESC, sauvegardes et jeu