

Kit pour Jeux de Cartes

par André C.

Avant de programmer un jeu de cartes, tel que Solitaire, Spider, Manille, Bridge etc., il faut disposer d'un ensemble de cartes conçues pour un affichage sur Oric. Et ce n'est pas une mince affaire !

Prenons un exemple concret. Dans le cas d'un simple jeu de Solitaire, il est nécessaire de pouvoir afficher 7 colonnes de cartes, ce qui réduit la largeur de chaque carte à 5 octets maximum, soit 1 attribut de couleur plus 4 octets de redéfinition. Ces 4 octets de redéfinition représentent 24 pixels en largeur, ce qui n'est pas si mal (voir ce que ça donne avec les figures 3 et 4 au paragraphe "Jeu de grandes cartes : JEU3.DAT").

Il est évidemment possible de créer de plus petites cartes, bien que la complexité de dessin des cartes à jouer rende très difficile la réduction de la taille des cartes.

Mode HIRES ou TEXT?

En mode TEXT, même en utilisant des cartes très petites (12x19 pixels soit 2x3 caractères, voir la figure 1 page suivante) et en se limitant à 32 cartes, il faudrait pouvoir redéfinir $32 \times 6 = 192$ caractères. L'Oric dispose de 96 caractères dans le jeu n°0 (normal) et 64 caractères dans le jeu n°1, soit 160 caractères redéfinissables en tout.

Et de toute façon, on ne peut pas redéfinir tous ces caractères, car il faut en garder quel-

ques-uns pour l'interface avec le joueur. Conclusion : L'utilisation du mode TEXT est très difficile pour programmer un jeu de cartes.

Il n'y a pas de regrets à avoir car le mode HIRES possède des avantages, notamment de permettre le positionnement des cartes au pixel près.

L'offre du jour

Le kit que je vous propose comporte 3 jeux de 56 cartes (52 cartes, un joker et 3 dos de cartes au choix). Ces 3 jeux correspondent à 3 tailles de cartes : petites, moyenne et grandes. Pour chaque jeu, les data de redéfinition sont regroupés dans un fichier JEU1.DAT, JEU2.DAT ou JEU3.DAT (voir la disquette accompagnant ce CEO-mag).

Le nombre d'octets de définition est fonction de la taille des cartes : 57 octets par carte pour celles de JEU1.DAT (petites cartes), 100 par carte pour celles de JEU2.DAT (cartes moyennes) et 180 par carte pour celles de JEU3.DAT (grandes cartes).

Pour afficher une carte donnée, il est nécessaire de savoir où se trouvent les octets de redéfinition de cette carte. J'ai donc attribué un numéro d'ordre à chaque carte (voir le tableau n°1). Ce numéro, combiné à la taille de carte, permet de calculer facilement l'adresse des octets de définition dans le fichier JEU*.DAT correspondant.

| | n° ordre de chaque carte dans le jeu | | | | | | | | | | | | Tableau n°1 |
|-----------------------|--------------------------------------|----|----|----|----|----|----|----|----|----|----|----|-------------|
| Nom des cartes → | AS | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | V | D | R |
| Carreau | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| Pique | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| Cœur | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| Trèfle | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 |
| Joker / Dos de cartes | 53 | 54 | 55 | 56 | | | | | | | | | |

C'est le même principe que celui utilisé classiquement pour trouver en Ram la position des octets de redéfinition des caractères.

Notez le moyen mnémotechnique utilisé pour se rappeler l'ordre des couleurs: "Le carreau pique le cœur du trèfle".

1) Le jeu de petites cartes : JEU1.DAT

Définition des cartes : 12 pixels de large (2 octets) et 19 pixels en hauteur.

En fait 11 pixels seulement sont utilisés sur les 12 car les dessins symétriques sont plus faciles à optimiser quand le nombre des pixels disponibles est restreint. A ces 2 octets

de large, il faut ajouter un octet pour l'attribut de couleur. Chaque carte est donc définie par 19 lignes HIRES de 3 octets, soit 57 octets.

Pour afficher une carte donnée, il suffit de connaître son numéro d'ordre et d'appliquer la formule suivante : $57 * (\text{numéro_d'ordre} - 1)$ qui donne l'offset des 57 octets de définition de la carte dans le fichier.

Par exemple, pour afficher la dame de cœur, le tableau n°1 de la page précédente indique que son numéro d'ordre est 38. La formule donne $57 * (38 - 1) = 2109$, soit #083D. C'est la valeur que l'on trouve dans le tableau n°2 ci-dessous.

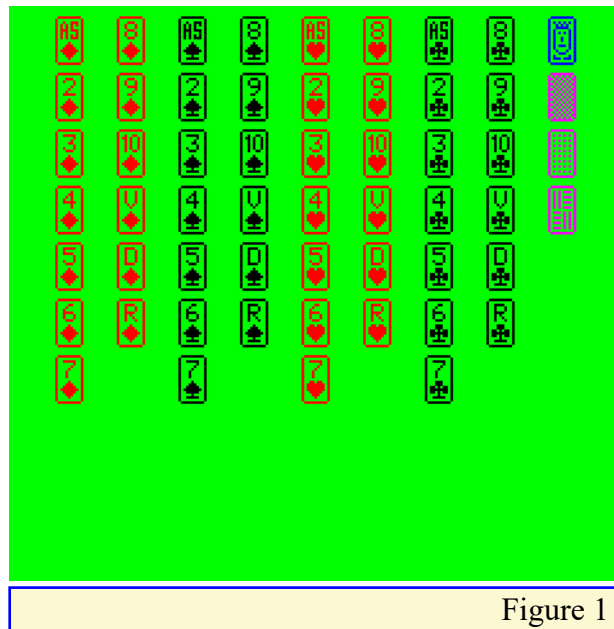


Figure 1

Offset de chaque carte dans JEU1.DAT

| Nom des cartes | AS | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | V | D | R |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Carreau | #0000 | #0039 | #0072 | #00AB | #00E4 | #011D | #0156 | #018F | #01C8 | #0201 | #023A | #0273 | #02AC |
| Pique | #02E5 | #031E | #0357 | #0390 | #03C9 | #0402 | #043B | #0474 | #04AD | #04E6 | #051F | #0558 | #0591 |
| Cœur | #05CA | #0603 | #063C | #0675 | #06AE | #06E7 | #0720 | #0759 | #0792 | #07CB | #0804 | #083D | #0876 |
| Trèfle | #08AF | #08E8 | #0921 | #095A | #0993 | #09CC | #0A05 | #0A3E | #0A77 | #0AB0 | #0AE9 | #0B22 | #0B5B |
| Joker/Dos | #0B94 | #0BCD | #0C06 | #0C3F | #0C78 | (eof) | | | | | | | |

Tableau n°2 : Adresses des cartes de JEU1.DAT

Le jeu comporte 56 petites cartes de 11x19 pixels, chaque carte est définie par 19 lignes HIRES de 3 octets, soit 57 octets. Ce tableau indique la position depuis le début du fichier (offset) de chacune des cartes dans le fichier JEU1.DAT (normalement localisé de #5000 à #5C78). Ces offsets restent valables si JEU1.DAT est déplacé en mémoire. Il suffit de les ajouter à l'adresse de chargement du fichier JEU1.DAT pour avoir les adresses réelles des cartes.

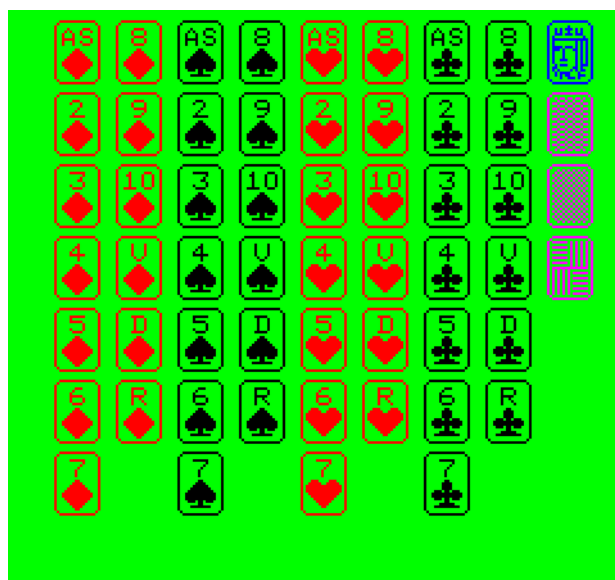


Figure 2

Si vous n'avez pas déplacé le fichier JEU1.DAT en mémoire, celui-ci a été chargé en #5000 et les 57 octets de data de la dame de cœur se trouvent à partir de #5000 + #083D = #583D.

Sur la disquette CARDS.DSK, le programme SHOWJEU1.BAS permet d'afficher l'ensemble des 56 cartes. La figure 1 de la page précédente montre ce que cela donne.

2) Le jeu de cartes moyennes : JEU2.DAT

Définition des cartes : 18 pixels de large (3 octets) et 25 pixels en hauteur. A ces 3 octets de large, il faut ajouter un octet pour l'attribut de couleur. Chaque carte est donc définie par 25 lignes HIRES de 4 octets, soit 100 octets.

Comme précédemment, pour afficher une carte donnée, il suffit de connaître son numéro d'ordre et d'appliquer la formule suivante : $100 * (\text{numéro_d'ordre} - 1)$ qui donne l'offset de la carte dans le fichier.

Par exemple, pour afficher le sept de carreau, le premier tableau indique que son numéro d'ordre est 7. La formule donne $100 * (7 - 1) = 600$, soit #0258. C'est la valeur que l'on trouve dans le tableau n°3 ci-dessous. Si vous n'avez pas déplacé le fichier JEU2.DAT en mémoire, celui-ci a été chargé en #5000 et les 100 octets de data du 7 de carreau se trouvent à partir de #5000 + #0258 = #5258.

Offset de chaque carte dans JEU2.DAT

| Nom des cartes | AS | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | V | D | R |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Carreau | #0000 | #0064 | #00C8 | #012C | #0190 | #01F4 | #0258 | #02BC | #0320 | #0384 | #03E8 | #044C | #04B0 |
| Pique | #0514 | #0578 | #05DC | #0640 | #06A4 | #0708 | #076C | #07D0 | #0834 | #0898 | #08FC | #0960 | #09C4 |
| Cœur | #0A28 | #0A8C | #0AF0 | #0B54 | #0BB8 | #0C1C | #0C80 | #0CE4 | #0D48 | #0DAC | #0E10 | #0E74 | #0ED8 |
| Trèfle | #0F3C | #0FA0 | #1004 | #1068 | #10CC | #1130 | #1194 | #11F8 | #125C | #12C0 | #1324 | #1388 | #13EC |
| Joker/Dos | #1450 | #14B4 | #1518 | #157C | #15E0 | (eof) | | | | | | | |

Tableau n°3 : Adresses des cartes de JEU2.DAT

Le jeu comporte 56 cartes de 18x25 pixels, chaque carte est définie par 25 lignes HIRES de 4 octets, soit 100 octets. Ce tableau indique la position depuis le début du fichier (offset) de chacune des cartes dans le fichier JEU2.DAT (normalement localisé de #5000 à #65E0). Ces offsets restent valables si JEU2.DAT est déplacé en mémoire. Il suffit de les ajouter à l'adresse de chargement du fichier JEU2.DAT pour avoir les adresses réelles des cartes.

Sur la disquette CARDS.DSK, le programme SHOWJEU2.BAS permet d'afficher l'ensemble des 56 cartes. La figure 2 de la page précédente montre ce que cela donne.

3) Le jeu de grandes cartes : JEU3.DAT

Définition des cartes : 24 pixels de large (4 octets) et 36 pixels de hauteur. A ces 4 octets de large, il faut ajouter un octet pour l'attribut de couleur. Chaque carte est donc définie par 36 lignes HIRES de 5 octets, soit 180 octets.

Comme pour les deux jeux précédents, pour afficher une carte donnée, il suffit de connaître son numéro d'ordre et d'appliquer la formule : $180 * (\text{numéro_d'ordre} - 1)$ qui donne

l'offset de la carte dans le fichier.

Par exemple, pour afficher le joker, le premier tableau indique que son numéro d'ordre est 53. La formule donne $180 * (53 - 1) = 9360$, soit #2490. C'est la valeur que l'on trouve dans le tableau n°4 ci-dessous.

Si vous n'avez pas déplacé le fichier JEU3.DAT en mémoire, celui-ci a été chargé de #5000 à #7760 et les 180 octets de data du joker se trouvent à partir de $\#5000 + \#2490 = \#7490$.

Sur la disquette CARDS.DSK, le programme SHOWJEU3.BAS permet d'afficher les 56

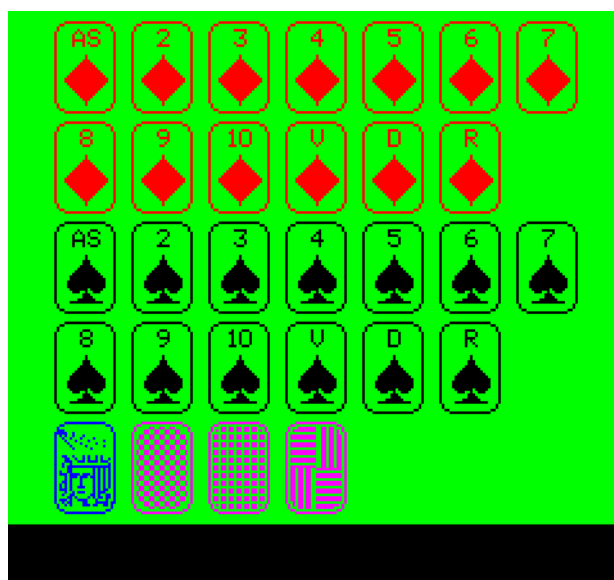


Figure 3

Offset de chaque carte dans JEU3.DAT

| Nom des cartes | AS | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | V | D | R |
|----------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Carreau | #0000 | #00B4 | #0168 | #021C | #02D0 | #0384 | #0438 | #04EC | #05A0 | #0654 | #0708 | #07BC | #0870 |
| Pique | #0924 | #09D8 | #0A8C | #0B40 | #0BF4 | #0CA8 | #0D5C | #0E10 | #0EC4 | #0F78 | #102C | #10E0 | #1194 |
| Cœur | #1248 | #12FC | #13B0 | #1464 | #1518 | #15CC | #1680 | #1734 | #17E8 | #189C | #1950 | #1A04 | #1AB8 |
| Trèfle | #1B6C | #1C20 | #1CD4 | #1D88 | #1E3C | #1EF0 | #1FA4 | #2058 | #210C | #21C0 | #2274 | #2328 | #23DC |
| Joker/Dos | #2490 | #2544 | #25F8 | #26AC | #2760 | (eof) | | | | | | | |

Tableau n°4 : Adresses des cartes de JEU3.DAT

Ce tableau indique la position depuis le début du fichier (offset) de chacune des cartes dans le fichier JEU3.DAT. Il faut donc ajouter ces offsets à l'adresse de chargement du fichier JEU3.DAT pour avoir les adresses réelles des cartes. Le jeu comporte 56 cartes de 24x36 pixels, chaque carte est définie par 36 lignes HIRES de 5 octets, soit 180 octets. Ces offsets restent valables si JEU3.DAT est déplacé en mémoire (il est normalement localisé de #5000 à #7760).

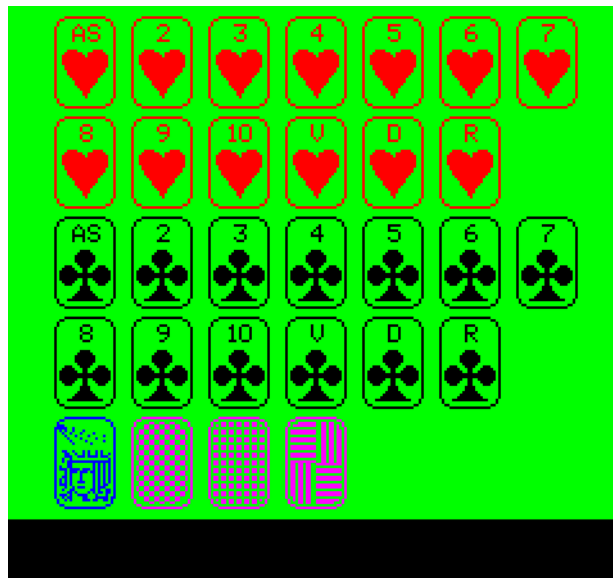


Figure 4

grandes cartes, mais doit utiliser 2 écrans HIRES. Les figures 3 et 4 montrent ce que cela donne.

Note sur l'affichage des cartes

Si vous avez lancé un des programmes Basic SHOWJEU*.BAS, vous avez pu constater l'extrême lenteur de l'affichage.

Même la compilation du programme Basic ne suffirait pas à accélérer suffisamment cet affichage pour qu'il puisse être utilisé dans un jeu. Il est donc nécessaire de mettre en place une petite routine en langage machine (LM) pour l'affichage d'une carte donnée.

A titre d'exemple, vous trouverez dans le fichier zip 3 listings source en LM : AFFCARTE1.ASM, AFFCARTE2.ASM et AFFCARTE3.ASM (un pour chacun des 3 types de cartes).

Il suffit de poker en page zéro les adresses source (adresse des octets de définition de la carte dans JEU*.DAT) et cible (adresse de la carte dans l'écran HIRES) et d'effectuer un CALL#9801. L'affichage de la carte est instantané !

Sur la disquette se trouvent 3 mini-demos illustrant la mise en œuvre de ces routines LM.

Contenu de la disquette

1. JEU1DAT, JEU2DAT et JEU3DAT qui contiennent les données de redéfinitions des 56 cartes des 3 jeux proposés.

2. SHOWJEU1.BAS, SHOWJEU2.BAS et SHOWJEU3.BAS qui affiche ces données et donc les 3 jeux de cartes complets.
 3. BONJOUR.COM qui, au boot, affiche les écrans correspondants (ce sont les figures 1 à 4 de cet article), à savoir JEU1.HRS, JEU2.HRS, JEU3A.HRS et JEU3B.HRS.
 4. AFFCARTE1.BAS et AFFCARTE1.BIN petite démo qui affiche la dame de cœur (en utilisant le langage machine).
 5. AFFCARTE2.BAS et AFFCARTE2.BIN petite démo qui affiche le sept de carreau (idem).
 6. AFFCARTE3.BAS et AFFCARTE3.BIN petite démo qui affiche le joker (idem).
- Soit 17 fichiers en tout.

Conclusion et clin d'œil !

Ce kit est en libre diffusion. Vous pouvez l'utiliser, le modifier etc. à votre guise. Tout ce que je souhaite, c'est qu'il vous inspire et que vous preniez plaisir à nous programmer un logiciel bien addictif !

Et pour finir, vous pouvez admirer, page suivante, les plus belles cartes du jeu dont je me suis inspiré. Toute ressemblance avec les cartes originales est évidemment voulue, mais le résultat est un vrai coup de bol !

Ces cartes font partie du célèbre "Loriot Rommé" dessiné par Vicco von Bülow, un humoriste, dessinateur, réalisateur, acteur, écrivain, scénariste et même artiste de cabaret...

