

Histoires de claviers

Par André C.

ETAT DU PROBLÈME

Tant que vous utilisez un Oric véritable, il n'y a pas de problème. Les choses se compliquent sérieusement avec un émulateur. En effet, la reconnaissance de telle ou telle touche repose sur deux choses : la Rom et le clavier. Je ne traite ici que de l'Oric-1 et de l'Atmos, le cas du Telestrat est un peu différent (mais guère). La difficulté vient de ce que l'Oric utilise un clavier QWERTY simplifié, alors que le PC (ou autre ordinateur hôte) utilise un clavier national, par exemple AZERTY en France. Les claviers nationaux sont plus compliqués car ils comportent des tas de lettres particulières (en France : à, é, è, ù, ç, etc.). Et la situation est analogue pour nos amis nordiques, germaniques, espagnols etc. D'où la nécessité de multiplier le nombre de touches de ces claviers et d'attribuer plusieurs caractères à une même touche. Avec l'Oric, nous avons déjà la classique touche SHIFT, héritée des machines à écrire, permettant en gros de basculer entre majuscules et minuscules. Sur les PC, une touche supplémentaire AltGr, permet d'ajouter des caractères nouveaux : ~, !, \, etc.) ou de recaser des caractères utilisés moins souvent : #, {, [, ^, @, etc.). Certains touches sont rémanentes et permettent d'associer un accent avec n'importe quelle lettre : â, ë, î, ò, ÿ, etc.). Tout cela n'a pas grande utilité pour jouer avec un émulateur Oric, mais cela encombre le clavier et certaines touches ont complètement changé de place.

LES RÉPONSES

Sedoric avait apporté une première réponse à ce problème, hélas trop partielle : La commande AZERTY simule l'utilisation d'un clavier AZERTY en transformant les appuis sur les touches : Q, W, A, Z, M, ; en A, Z, Q, W, ;, M. Tandis que la commande QWERTY rétablit la situation native. De même la commande ACCENT SET transforme les appuis sur les touches : @, \, {, !, }, ■ en à, ç, é, ù, è, ê. Tandis que la commande ACCENT OFF rétablit la situation native. Mais c'est vraiment une illusion de se limiter à ces pauvres corrections.

Pour pallier plus efficacement à cette gêne, Fabrice (et d'autres personnes) on introduit des Roms modifiées pour tenir compte des claviers nationaux. Ce sont les Bas11_**.rom (avec par exemple DE, ES, FR, SE, UK à la place des

). Pour ma part, j'ai fait une Rom Basic10_fr.rom (calquée sur Bas11_fr) afin de jouer avec Euphoric en configuration Oric-1. D'autres Roms ont également été élaborées, notamment les Bas12_.rom, BAS121_**.rom et Bas122_**.rom (avec DE, ES, FR, SE, UK à la place des **), basées sur les géniales Roms de Pascal Leclerc (voir par exemple l'article de Yann L. dans le Ceo-Mag n°295, pages 28 à 31).

LA RECONNAISSANCE DES TOUCHES PRESSÉES

Ouf, c'est un beau paquet d'épines retirées de nos petits petons ! Mais cela ne suffit pas. L'Oric utilise plusieurs procédures en Rom pour reconnaître quelle(s) touche(s) a(ont) été pressée(s). Il est possible de se greffer sur ces procédures, notamment dans les programmes en langage machine. Mais heureusement, l'Oric met aussi à notre disposition des commandes Basic pour exploiter les entrées au clavier. Il existe donc toute une panoplie de manières de reconnaître les touches pressées, basées sur les codes Ascii, les chaînes de caractères, la valeur des chiffres ou les scan-codes du clavier.

Les commandes GET R\$ et GET R attendent qu'une touche soit pressée et en retournent le caractère dans la variable R\$ ou la valeur (seulement pour les chiffres) (attention, ce n'est pas le code Ascii) dans la variable R (ici R\$ et R sont des exemples de noms de variable). La variable "chaîne de caractère" ne contient alors qu'un seul caractère. Idem la variable numérique ne contient qu'un seul chiffre. Pour recueillir une chaîne plus longue ou un nombre plus important, il faut utiliser la commande INPUT et terminer par ENTER.

La commande KEY\$ est similaire à GET R\$, mais n'attend pas la réponse. Elle travaille en quelque sorte "à la volée" et nécessite généralement une boucle pour saisir quelque chose. On remarque aussi une différence de syntaxe (R\$=KEY\$ eu lieu de GET R\$).

Dans les 2 cas, pour obtenir le code Ascii, il faut compléter avec la commande ASC(R\$). Ne pas confondre avec la commande VAL(N\$) qui retourne la valeur numérique du nombre exprimé dans une variable chaîne. Rappelons que CHR\$(valeur) effectue l'opération inverse de ASC(R\$): on fournit le code ASCII et on récupère le caractère.

Enfin, pour les utilisateurs de Sedoric, la

commande KEYIF (valeur) GOTO... ou KEYIF(valeur) THEN... ELSE... exploite le scan-code du clavier. Avantage : l'exécution est plus rapide que si on utilisait un code Ascii ou un caractère. Inconvénient : Les scan-codes sont pour le moins abscons (voir "Manuel d'utilisation du Microdisc Oric et du Sedoric", annexe 7, page 104). Par exemple le scan-code de la touche **A** est #AE. En fait, il ne faut pas dire "le scan-code de **A** est", mais "le scan-code de la 2e touche noire du 3e rang est" #AE, car c'est la position de la touche qui compte et non son étiquette. Après une commande AZERTY, cette touche a toujours le scan-code #AE, mais affiche un **Q**.

Enfin, pour les amateurs de langage machine, les adresses mémoires #0208, #0209 et #2DF peuvent rendre de grand services, notamment en terme de rapidité. Les utilisateurs avertis du langage Basic pourront faire de même avec des PEEK(#208), PEEK(#0209) et PEEK(#2DF). Voyons le détail :

- L'adresse mémoire #0209 ne concerne que les touches de fonctions (CTRL, FUNCT, SHIFT gauche, SHIFT droit) et contient le scan-code de la touche pressée (voir ci-dessus) ou #38 si aucune touche n'a été pressée. Voici les 4 scan-codes en question : #A2 pour CTRL, #A5 pour FUNCT, #A4 pour SHIFT gauche et #A7 pour SHIFT droit).
- L'adresse mémoire #0208 ne concerne que les autres touches et contient le scan-code de la touche pressée (voir ci-dessus) ou #38 si aucune touche n'a été pressée. Mais ces scan-codes ne sont pas très facile à mémoriser !
- L'adresse mémoire #2DF est plus classique, car elle gère le code Ascii. Elle concerne toutes les touches ayant un code Ascii. Mais il faut savoir comment ça marche pour bien l'utiliser. Si aucune touche n'a été pressée, le bit 7 de l'octet en #2DF est à zéro. Lorsqu'il passe à un, c'est qu'une touche a été pressée. Les bits 6 à 0 contiennent alors le code Ascii de la touche pressée.

CE QUE ÇA DONNE EN PRATIQUE.

Tout ça c'est bien beau, mais voyons ce qu'on récolte en utilisant ces différentes approches, aussi bien avec un Oric réel qu'avec un Oric émulé sur PC.

L'Atmos comporte 58 touches, dont la majorité donne accès à 2 caractères avec SHIFT. Mais seulement 26 de ces touches, soit moins de la moitié sont présentes au même endroit et avec la même fonction sur un clavier AZERTY de PC. Certaines ont gardé le même rôle mais ont changé de place, par exemple la touche **Qq**. Certaines touches ont changé de place et de fonction. Par

exemple, pour obtenir les 2 caractères de la touche **£-** (Oric, à droite du zéro) il faudra presser sur PC soit **£\$** soit **6-** ! Mais c'est l'apparition de la touche AltGr, qui a créé le plus de bouleversements en donnant 3 rôles à de nombreuses touches. Et inévitablement cela a entraîné une redistribution considérable. Par exemple les 3 caractères de la touche PC **5([** sont obtenus avec les 3 touches **%5**, **(9** et **{[** sur un clavier Oric !

Les variations étant presque infinies, pour ne pas nous lancer dans une étude encyclopédique, nous ne testerons que quelques caractères, chacun représentant un ensemble ayant les mêmes caractéristiques. La première rangée de touches en haut du clavier sera tout particulièrement explorée.

- **Q** (comme échantillon pour l'ensemble des touches **Q, W, A, Z, M, ;** et **@, \, {, |, }, ■**).
- **CTRL** (comme échantillon pour l'ensemble des 4 touches de contrôle : CTRL, FUNCT, SHIFT gauche et SHIFT droit) .
- **Flèche gauche** (comme échantillon pour l'ensemble des autres touches rouges, soit ESC, les 4 flèches, RETURN et DEL).
- **SHIFT1** (SHIFT gauche+**1**, comme échantillon pour l'ensemble des touches "SHIFTkeukchose" générant sur l'Oric un caractère auquel on peut accéder sur PC par frappe directe sur une touche située ailleurs, soit **§!, £\$, ^, 1&, µ*, 5([, °]** etc.
- **SHIFT5** (SHIFT gauche+**5**, comme échantillon pour l'ensemble des touches "SHIFTkeukchose" générant sur l'Oric un caractère auquel on peut accéder sur PC par une touche "SHIFTkeukchose" située ailleurs, par exemple **%ù, £\$** etc.
- **SHIFT#** (SHIFT gauche+**#**, comme échantillon des touches "SHIFTkeukchose" ayant un nouvel accès avec AltGr sur PC, soit **0à@, 3"#, 6-l**, etc.
- **SHIFT+** (SHIFT gauche+**+**, comme échantillon des touches "SHIFTkeukchose" ayant gardé la même place et la même fonction.

J'ai donc retenu ces 7 touches comme petit échantillon de test. Je me propose de voir ce que ça donne quand on presse ces touches sur :

- Ce qui s'affiche à l'écran (résultat direct)
- Ce qui est imprimé par GETR\$:PRINTR\$;
- Ce qui est imprimé par R\$=KEY\$:PRINTR\$;
- Ce qui est imprimé par PRINTEX\$(PEEK(#208))
- Ce qui est imprimé par PRINTEX\$(PEEK(#209))
- Ce qui est imprimé par PRINTEX\$(PEEK(#2DF))

Et ceci sur un Atmos réel, sur Sedoric avec la Rom Atmos standard Basic11b.rom ou sur Sedoric avec la Rom française correspondante Bas11_fr.rom.



VOICI CE QUE ÇA DONNE :

A) Ce qui s'affiche à l'écran (résultat direct)

Observation : On a l'impression que Atmos reel et Sedoric Basic11b donnent la même chose, mais comme le révèle le A obtenu à la place du Q, il existe une douzaine de touches qui ne correspondent pas (au point qu'il est très difficile de taper sur un clavier AZERTY avec une Rom normale). Avec Sedoric Bas11_fr on obtient ce qui correspond aux étiquettes des touches du clavier AZERTY. Voir premier tableau ci-contre.

B) Ce qui est affiché par GETR\$:PRINTR\$:

Mêmes résultats. Voir premier tableau ci-contre.

C) Ce qui est affiché par R\$=KEY\$:PRINTR\$:

Mêmes résultats. Voir premier tableau ci-contre.

D) Ce qui est imprimé par PRINTHEX\$(PEEK(#208))

Observation : L'adresse mémoire #208 ne concerne pas les touches de contrôle (#38 est retourné comme quand aucune touche n'a été pressée), mais uniquement les autres touches (et retourne le scan-code de la touche). Le scan-code de Q est bien #B1, alors que celui de A est #AE. Au premier abord, on ne comprend pas que Sedoric Bas11_fr se laisse tromper et le retourne le scan-code de A alors qu'il affiche Q dans les autres tests. En fait, les scan-codes n'ont rien à voir avec les étiquettes des touches, mais uniquement avec la place de la touche dans le clavier. On devrait

| Touche | Atmos | SedoBasic11 | SedoBas11fr |
|---------|------------|-------------|-------------|
| Q | Q | A | Q |
| CTRL | rien | rien | rien |
| FlècheG | recule à G | recule à G | recule à G |
| SHIFT1 | ! | ! | & |
| SHIFT5 | % | % | (|
| SHIFT# | # | # | " |
| SHIFT+ | + | + | = |

| Touche | Atmos | SedoBasic11 | SedoBas11fr |
|---------|-------|-------------|-------------|
| Aucune | #38 | #38 | #38 |
| Q | #B1 | #AE | #AE |
| CTRL | #38 | #38 | #38 |
| FlècheG | #AC | #AC | #AC |
| SHIFT1 | #A8 | #A8 | #A8 |
| SHIFT5 | #90 | #90 | #90 |
| SHIFT# | #B8 | #B8 | #B8 |
| SHIFT+ | #BF | #BF | #BF |

dire "le scan-code de la touche" et pas "le scan-code de tel caractère". Les touches SHIFT1, SHIFT5, SHIFT# et SHIFT+ obtiennent toujours le bon scan-code, car elles n'ont pas bougé de place entre clavier Oric et clavier PC.

E) Ce qui est imprimé par PRINTHEX\$(PEEK(#209))

Observation : L'adresse mémoire #209 ne concerne que les 4 touches de contrôle, quelle que soit la touche pressée conjointement. Si aucune touche n'est pressée ou si une touche "normale" est pressée, cela donne la même chose : #38 = pas de touche contrôle pressée. #A2 et #A4 sont bien les scan-codes de CTRL et de SHIFT gauche respectivement et ces 2 scan-codes n'ont pas été inversés alors que la position de ces deux touches a été permutée entre le clavier Oric et le clavier PC. La règle n'est donc pas absolue, mais a été raisonnablement adaptée.

F) Ce qui est imprimé par PRINTHEX\$(PEEK(#2DF))

Observation : L'adresse mémoire #209 contient #00 si aucune touche n'a été pressée. Son bit 7 est à un si une touche a été pressée et les autres bits contiennent le code ASCII de la touche pressée. A la valeur hexadécimale obtenue, il faut donc retirer #80 (1000 0000 en binaire) par avoir le code ASCII proprement dit. La valeur entre parenthèses indique le code ASCII et le caractère correspondant suit. On voit que c'est cohérent avec ce qu'on a en direct à l'écran, ce que retourne GET\$ et ce que retourne KEY\$.

| Touche | Atmos | SedoBasic11 | SedoBas11fr |
|---------|-------|-------------|-------------|
| Aucune | #38 | #38 | #38 |
| Q | #38 | #38 | #38 |
| CTRL | #A2 | #A2 | #A2 |
| FlècheG | #38 | #38 | #38 |
| SHIFT1 | #A4 | #A4 | #A4 |
| SHIFT5 | #A4 | #A4 | #A4 |
| SHIFT# | #A4 | #A4 | #A4 |
| SHIFT+ | #A4 | #A4 | #A4 |

| Touche | Atmos | SedoBasic11 | SedoBas11fr |
|---------|-------------|-------------|-------------|
| Aucune | #00 | #00 | #00 |
| Q | #D1 (#51) Q | #C1 (#41) A | #D1 (#51) Q |
| CTRL | rien | rien | rien |
| FlècheG | #88 (#08) | #88 (#08) | #88 (#08) |
| SHIFT1 | #A1 (#21) ! | #A1 (#21) ! | #A6 (#26) & |
| SHIFT5 | #A5 (#25) % | #A5 (#25) % | #A8 (#28) (|
| SHIFT# | #A3 (#23) # | #A3 (#23) # | #A2 (#22) " |
| SHIFT+ | #AB (#2B) + | #AB (#2B) + | #BD (#3D) = |

EN CONCLUSION

Si vous avez un Oric réel devant vous, pas de problème, c'est bon dans tous les cas. Si vous utilisez un émulateur avec clavier AZERTY, Sedoric doit être configuré avec Bas11_fr.rom (ou leurs dérivés tels que Basic10_fr.rom et Bas12x_fr.rom). Sinon vous allez au-devant de bien des difficultés car l'interface d'entrée ne sera pas adaptée.

Par contre, si vous voulez programmer ou modifier un jeu ou une application qui fonctionne aussi bien sur un clavier Oric que sur un clavier AZERTY et qui repose sur la position des touches à presser, vous devez impérativement utiliser les scan-codes. C'est le cas des jeux utilisant autre chose que les 4 flèches pour les déplacements. Par exemple l'utilisation des fameuses touches IJKM (respectivement pour haut, gauche, droite, bas avec un clavier Oric) ne marchera pas avec un clavier PC (sauf si les lettres sont reconnues comme telles, mais alors la position relative des touches est perdue).

La situation est pire, si le programme utilise SHIFT combiné à un chiffre de 0 à 9. En effet, au lieu de presser les touches SHIFT plus 0 à 9, ce qui donne en réalité les caractères), !, @, #, \$, %, ^, &, *, (il faudra presser les touches ou combinaisons de touches qui donnent ces mêmes

caractères, ce qui peut devenir compliqué, car c'est le désordre complet (je vous renvoie à la comparaison visuelle des claviers Oric et PC à la page précédente). SAUF si le programme a été bien fait et gère les scan-codes au lieu des caractères ASCII...

MORALITÉ

Les Roms nationales ne sont pas des adaptations absolues (c'est impossible), mais permettent d'utiliser raisonnablement un émulateur avec un clavier non QWERTY. Je n'ai pas abordé le CTRL+T car cette fonction n'arrange rien. C'est un mode bâtarde, dont le seul intérêt est de pouvoir accéder aux lettres minuscules.

Le seul point gênant restant, mais impossible à résoudre, est que sur un Oric réel dans son état natif (QWERTY et CAPS), la frappe des touches produit des chiffres, des lettres majuscules et les caractères des rangées supérieures (comme si une touche CapLock invisible était bloquée), alors que devant un PC dans son état usuel, la frappe des touches produit des lettres minuscules et les caractères des rangées inférieures. C'est surtout gênant avec les caractères que l'on utilise moins souvent, car nos doigts ont leurs habitudes, que nos yeux peinent à corriger...