

SEDORIC? DO IT YOURSELF!  
(13)

Aujourd'hui **LOVE !**

Une des plus belles bogues de Sédoric concerne la routine "**Prendre un caractère au clavier**" et plus particulièrement le sous-programme "Cherche la 1<sup>ère</sup> lettre du mot-clé Sédoric de n° d'ordre X" (#D907 à #D927). Nous allons y mettre bon ordre.

### La routine "Prendre un caractère au clavier" de Sédoric (#EB78 / RAM OVERLAY) :

Cette routine examine si une touche a été pressée. Si ce n'est pas le cas, elle termine avec N = 0. Si une touche ordinaire a été pressée, elle met le code ASCII correspondant dans A, force N à 1 et termine avec le b7 de #C049 à 0 (ce qui signifie "pas de touche de fonction en cours"). Si une touche de fonction a été pressée (combinaison FUNCT+touche ou FUNCT+SHIFT+touche, voir le manuel Sédoric page 54, 55 et 102), elle recherche la chaîne de caractères qui correspond au code de fonction associé. Puis elle termine avec le 1<sup>er</sup> caractère dans A, N à 1, et b7 de #C049 à 1 (ce qui signifie "il y a d'autres caractères à saisir"). Lors du prochain appel à la routine #D845, Sédoric examine l'état du b7 de #C049, le trouvant à 1, il ira directement lire le caractère suivant et ainsi de suite jusqu'au dernier. Lorsqu'il arrive au dernier caractère de la chaîne en cours de traitement, il remet le b7 de #C049 à 0 et retourne avec ce dernier caractère dans A et N à 1.

### La table "KEYDEF" (#C800 à #C87F) :

Comme l'indique le manuel, il y a 256 codes de fonction différents (par exemple, le code 87 correspond à MOVE). La table "KEYDEF" mémorise la correspondance entre chaque "FUNCT+touche" ou "FUNCT+SHIFT+touche" et l'un de ces 256 codes de fonction.

La lecture de cette table montre qu'il y règne la plus grande anarchie. Certains codes y existent en plusieurs exemplaires : #00, #09, #0C et #1C. Le code #00 associé à la chaîne "PRINTHEX\$(DEEK(#) s'y trouve 28 fois ! D'autres codes sont complètement absents : **il n'y a aucun code de #20 à #7F (mots-clés Sédoric)** et il manque 9 des 16 commandes prédéfinies. Le sous-programme traitant des codes correspondants aux mots-clés Sédoric étant complètement bogué, il est clair qu'au dernier moment les codes Sédoric ont été remplacés par des #00.

L'attribution des diverses fonctions aux 256 codes de fonctions est elle même anarchique. D'une part, 126 codes (de #80 à #FD) ont été alloués aux 119 mots-clés BASIC (token de #80 à #F6) les 7 derniers codes de fonctions (de #F7 à #FD) sont donc inutilisés. D'autre part, la table des mots-clés Sédoric comporte 104 entrées (dont certaines en double) or de #20 à #7F il n'y a que 96 codes, résultat : les dernières commandes Sédoric ne sont pas accessibles (pas de code).

L'utilisateur ne peut en principe intervenir qu'au moyen des commandes KEYUSE (permettant de redéfinir les 16 premières commandes de code #00 à #0F) et KEYDEF (permettant de ré-attribuer les "256" codes de commandes disponibles aux diverses combinaisons "FUNCT+touche" et "FUNCT+SHIFT+touche"). Il est évident qu'il faudrait **revoir les commandes prédéfinies**. Certains codes (HIRES, RUN, etc.), qui sont accessibles par ailleurs, devront être remplacés par les 5 commandes Sédoric UNPROT, VUSER, WIDTH, WINDOW et RESTORE qui sont complètement inaccessibles. Nous pensons revoir tout cela dans la version 3 de Sédoric !

### Mise en évidence de la bogue :

Nous allons attribuer le code 87 (commande Sédoric MOVE), à la combinaison "FUNCT+1" (un des multiples #00). Pour cela tapez KEYDEF 87.␣ puis FUNCT+1. Maintenant testez si la combinaison FUNCT+1 vous affiche bien MOVE à l'écran. Hélas non, on obtient LOVE ! (voir illustration de gauche à la fin de cet article).

## Principe du débogage :

Le sous-programme "Cherche la 1<sup>ère</sup> lettre du mot-clé Sédoric de n° d'ordre X", situé de #D907 à D927 est complètement bogué. Il est possible de le remettre en état, mais la place disponible localement est trop courte de 2 octets. Nous allons donc procéder comme nous avons fait pour LINPUT : avec un JSR dans l'ancienne commande EXT.

## Mise en pratique :

Bootez avec vos disquettes "DO IT 1" en A et "DO IT 2" en B. Tapez HIMEM#13FF↵, NOYAU↵ et B-MONAC1↵. Le listing ci-dessous reproduit la version déboguée complète (en 2 parties #2907 à #2927 et 3A30 à #3A35) du sous-programme (adresses en RAM).

Dumpez en #290A vous devez obtenir les octets suivants : 86 F2 A2 3F etc. Assemblez en #290A les mnémoniques indiqués ci-dessous en gras (1<sup>ère</sup> partie du listing). Puis dumppez en #3A30. Vous devez obtenir les octets EA EA EA EA EA EA 86 etc. Assemblez en #3A30 les mnémoniques indiqués ci-dessous en gras (2<sup>ème</sup> partie du listing).

2907	A5 F2	LDA \$F2	291B	8E 4B C0	<b>STX \$C04B</b>
2909	48	PHA	291E	A6 F2	<b>LDX \$F2</b>
<u>290A</u>	20 31 EA	<b>JSR \$EA30</b>	2920	CA	<b>DEX</b>
290D	A0 00	<b>LDY #\$00</b>	2921	68	PLA
290F	B9 BD CB	<b>LDA \$CBBD,Y</b>	2922	85 F2	STA \$F2
2912	E8	<b>INX</b>	2924	A9 DD	LDA #\$DD
2913	C8	<b>INY</b>	2926	A0 C9	LDY #\$C9
2914	C8	<b>INY</b>	puis...		
2915	C8	<b>INY</b>	<u>3A30</u>	E8	<b>INX</b>
2916	C8	<b>INY</b>	3A31	86 F2	<b>STX \$F2</b>
2917	C5 F2	<b>CMP \$F2</b>	3A33	A2 3F	<b>LDX #\$3F</b>
2919	90 F4	<b>BCC \$290F</b>	3A35	60	<b>RTS</b>

Vérifiez soigneusement votre travail. Pendant que vous y êtes, allez voir si le code de MOVE (#57) est bien là en #1828. Sauvez avec SAVEU"B-NOYAU",A#1400,E#4FFF↵. Régénérez votre disquette "DO IT 1" avec B-MISEAJOUR↵ et rebootez.

## Et maintenant on teste !

C'est simple : FUNCT+1 doit cette fois vous afficher MOVE et non plus LOVE... un de perdu 10 de retrouvés ! Ce résultat est illustré ci-dessous avec la recopie d'écran située à droite.

```
SEDORIC V1.0                                CAPS
@ SEDORIC DO IT YOURSELF!

Drive A (Do It!) Do It 1 (Euphoric)
NOYAU .COM 61 BANQUE1 .COM 5
BANQUE2 .COM 5 BANQUE3 .COM 5
BANQUE4 .COM 5 BANQUE5 .COM 5
BANQUE6 .COM 5 BANQUE7 .COM 5

*607 free sectors (S/42/17), 8 Files
Ready
AVEC FUNCT+1 A LA PLACE DE MOVE
ON OBTIENT :
LOVE■
```

```
SEDORIC V1.0                                CAPS
@ SEDORIC DO IT YOURSELF!

Drive A (Do It!) Do It 1 (Euphoric)
NOYAU .COM 61 BANQUE1 .COM 5
BANQUE2 .COM 5 BANQUE3 .COM 5
BANQUE4 .COM 5 BANQUE5 .COM 5
BANQUE6 .COM 5 BANQUE7 .COM 5

*607 free sectors (S/42/17), 8 Files
Ready
AVEC FUNCT+1 ON OBTIENT MAINTENANT :
MOVE■
```