AirStrike

Yassine LOUDAD

Yassine LOUDAD
28/05/2010

Sommaire

Introduction

L’assembleur est le langage le plus proche du langage machine, ainsi il présente différents avantages et inconvénients qui sont un critère important de faisabilité du programme.

Mon choix s’est porté sur le développement d’un jeu en 2D, ceci nous permettra d’explorer toutes les possibilités offertes par l’assembleur 68k et son émulateur Easy68k qui propose des fonctions Trap fort intéressantes, comme l’affichage graphique, le jeu de sons, l’accès aux fichiers et aux périphériques d’entrée.

Mon jeu est un jeu d’arcade en 2d du genre « shoot'em up», c’est un avion qu’on peut déplacer verticalement pour tirer sur les avions ennemis qui viennent de la droite, en évitant les avions de ligne qui peuvent se cacher derrière des nuages.

I- Structure du programme

Comme tout programme avec interface graphique, mon programme est muni d'une boucle principale infinie qui exécute les tâches suivantes :

- Intercepter les signaux du clavier et les traiter.
- Créer de nouveaux avions et nuages en fonction du score.
- Vérifier d'éventuelles collisions entre les unités (missiles compris).
    - Appeler la fonction GameOver au cas où le joueur a perdu.
- Déplacer les unités (Changer leurs coordonnées).
- Effacer la zone de jeu.
- Afficher les unités.
    - Appeler les fonctions d'affichage de chacune des unités.
- Afficher le score et le nombre de bavures restantes.

II- La mise en mémoire des unités

Le nombre d'unités, que ce soit des nuages, des avions ennemis ou civils, croit avec le score. Ainsi, j'ai décidé de mettre les unités sur des tableaux. Au début on commence par allouer une zone mémoire qui peut contenir jusqu'à 100 unités, et on déclare une variable pour stocker le nombre d'unités qui doivent s'afficher :

```
Nuages     DS.w 300   (vitesse, Y, X)
NombreNuages   Dc.l 0

Civils     DS.w 300   (vitesse, Y, X)
NombreCivils   Dc.l 0

Mechants   DS.w 300   (vitesse, Y, X)
NombreMechants Dc.l 0

Tirs DS.l 30    (existence, Y, X)
NombreTirs     Dc.l 10
DernierTir     Dc.l 0
```

III- Le jeu de sons

Easy68k peut jouer des sons en même temps, ce qui nous permet d'avoir une musique de fond ainsi que des sons de tirs et d'explosions.

On commence par charger les fichiers de musiques qui sont au format wav avant la boucle infinie :

```
lea     SonMusique,a1  Charger la musique de fond
move    #0,d1
move    #74,d0
trap    #15


lea     SonGameOver,a1 Charger la musique du game over
move    #1,d1
move    #74,d0
trap     #15
```

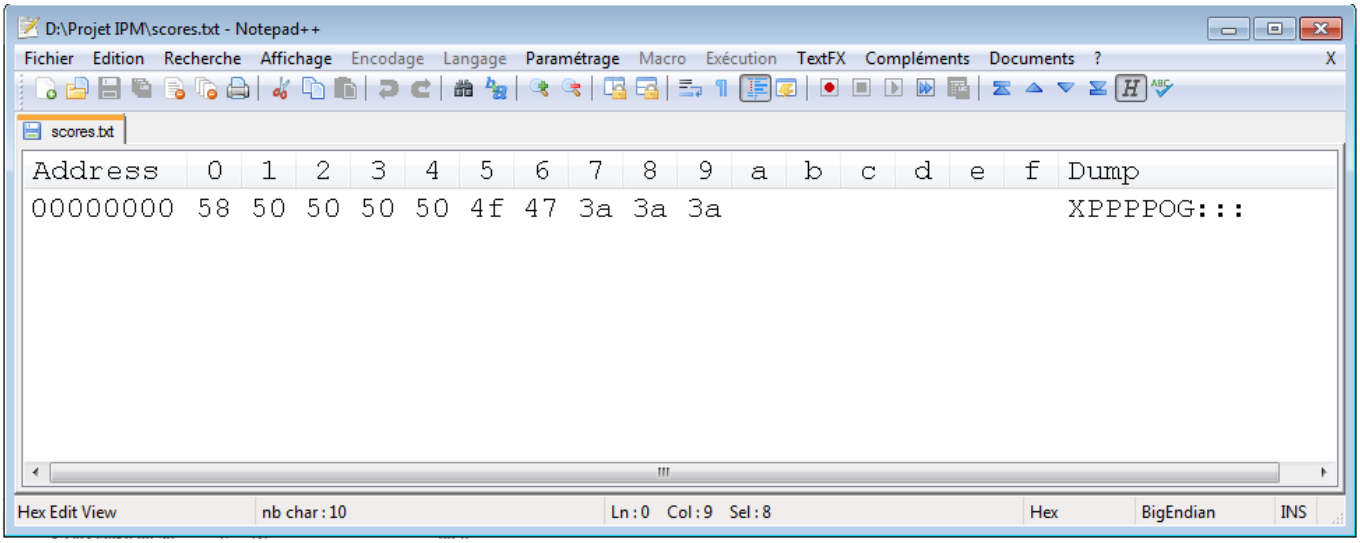
Et on lance une ou l'autre selon la phase du jeu :

```
move.l  #0,D1           Arreter la musique de fond
move.l  #2,D2
move    #77,d0
trap    #15

move.l  #1,D1           Jouer la musique du gameover
move.l  #0,D2
move    #77,d0
trap    #15
```

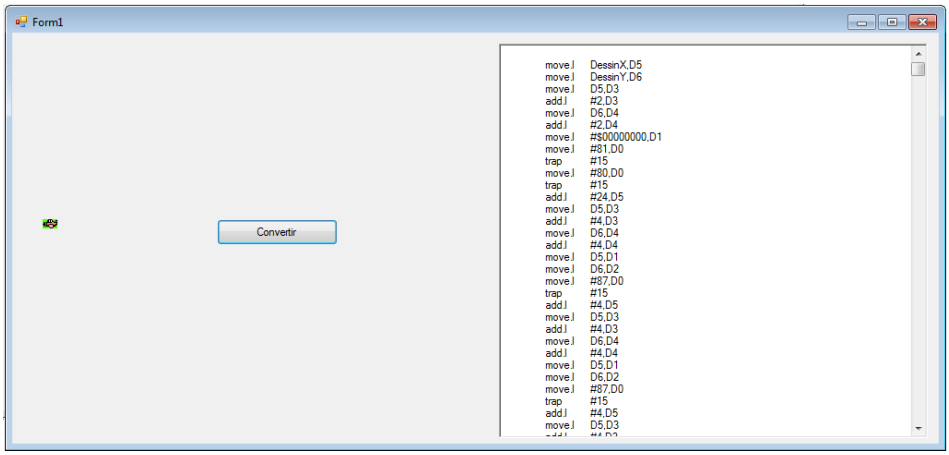
IV- L'enregistrement du score

A la fin d'une partie, il faudra afficher les meilleurs scores et enregistrer le score du joueur s'il est parmi les dix premiers. Les scores sont enregistrés dans le fichier scores.txt, 10 octets pour 10 scores. Ainsi, on commence par lire les scores un par un, et les afficher tant qu'ils sont supérieurs au score du joueur, sinon, on fait translater les scores restants à droite et on enregistre le score en écrasant celui de l'itération en cours, et on continue l'affichage :



V- Affichage des avions, nuages, et des missiles

Bien qu'Easy68k soit muni de plusieurs fonctions graphiques qui permettent de faire des dessins simples, afficher une image comme celle d'un avion ou un missile reste une tâche compliquée à réaliser en assembleur 68k. Pour nous la faciliter, j'ai développé un petit convertisseur en C# qui donne le code assembleur qui permet de dessiner une image donnée



rien qu'avec les fonctions simples dont dispose l'émulateur 68k. Ainsi on a créé pour toute image une fonction, et pour dessiner cette image il suffit de mettre les coordonnées dans (DessinX, DessinY) et faire appel à sa fonction :

```
move.l      D2,DessinY
move.l      D3,DessinX

BSR DessinerMissile
```

Conclusion

Le but de mon projet était de profiter de toutes les fonctionnalités que dispose l'assembleur 68k, que ce soit au niveau graphique, au niveau du jeu de sons ou de l'IO, et je pense l'avoir atteint avec mon jeu AirStrike. Il aurait été possible d'ajouter des améliorations au jeu, notamment la saisie du nom de joueur lors de l'enregistrement du score, ou ajouter aux avions ennemis la possibilité de tirer des missiles.