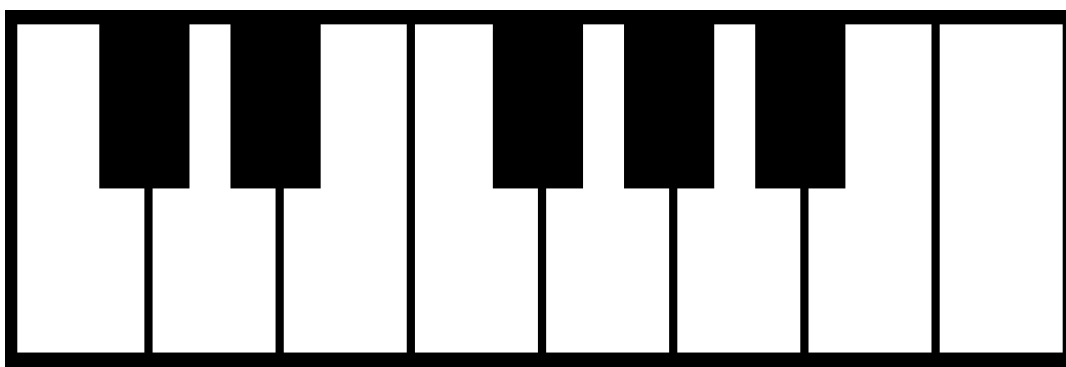


Projet en groupe d'architecture 1ARC

Play-o-matic

Asc.1



Documentation technique

VMware vs VirtualBox :

Nous avons fait le choix d'utiliser VMware car il s'avère que le son n'est pas émulé sans réglage préalable sous VirtualBox.

L'affichage du piano :

Il y a d'abord l'affichage des touches blanches et ensuite seulement les touches noires.

Pour l'affichage des touches blanches, il faut d'abord définir le mode de vidéo, la couleur (1111b = blanc et 0000b = noir), et utiliser l'interruption système 10h qui permet l'affichage.

Ensuite, on va utiliser CX et DX pour placer nos pixels. Il faut imaginer qu'il s'agit d'un plateau à deux dimensions ; CX = l'axe des abscisses et DX = l'axe des ordonnées.

On incrémente DX pour faire une ligne verticale, puis on réinitialise la valeur de DX et on incrémente CX, cela permet de refaire une ligne verticale mais décalé d'un pixel vers la droite.

Il faut aussi rajouter des comparaisons qui empêcheront le programme de « faire des lignes verticales » tous les 30 pixels (par exemple). Cela permet de faire définir les touches visuellement.

Pour les touches noires, c'est le même principe, à la différence qu'il faut définir CX et DX pour chaque touche noire. Sinon, le système de comparaison reste le même et la réinitialisation de la ligne est la même.

Les sons :

En assembleur, pour faire un piano, ou plus généralement des sons, il y a certaines choses à prendre en compte.

Pour jouer une note (ou un son), il faut ouvrir le port 61h, c'est lui qui gère le haut-parleur de l'ordinateur.

Une fois cela fait, il faut accorder la note. Pour ça, il nous faut la fréquence en Hertz de la note et effectuer un calcul avec.

Prenons par exemple la note « La », avec une fréquence de 440 Hz. Pour accorder la note en assembleur, il nous faut faire $1193180 / 440$. Cela nous permet de trouver un nombre, qu'il faudra utiliser dans le code afin d'obtenir la note « parfaite ».

Pourquoi 1193180 ? Il s'agit de la fréquence de base du haut-parleur sur un ordinateur, ce chiffre est donné par l'oscillation de la puce.

Les jumps :

Maintenant que l'on sait jouer des notes en assembleur, cela paraît aisé de faire un piano contenant toutes les octaves, mais il n'en est rien. On est très vite confronté à un problème de taille : les sauts (jump).

En assembleur, il y a deux types de saut : les sauts conditionnels et les sauts inconditionnels.

Les sauts inconditionnels ont la capacité de pouvoir traverser le code de part en part. Mais ce n'est pas le cas des sauts conditionnels.

Du coup il faut réorganiser son code afin de ne pas avoir un saut conditionnel qui doit faire tout le code.

La compilation du programme :

Une fois notre code fini, il faut le « transformer » en disquette bootable. Pour cela, nous avons utilisé une machine virtuelle linux (Debian) et installé le logiciel NASM (aptitude install nasm). Ensuite, il faut faire : `nasm -f bin -o LeNomDuFichier LeNomDuFichier.asm`. Ensuite, il faut la convertir en disquette bootable : `cat LeNomDuFichier /dev/zero | dd of=NomFloppy bs=512 count=2880`.

Enfin, il suffit de faire une machine virtuelle, qui démarre sur cette disquette.