MICHEL Hugo
MASSON Alexandre

Compte rendu de notre bureau d'étude : Application du C++ au domaine des objets connectés

Nous avons décidé lors de ce bureau d'étude de mettre en application nos connaissances en C++ lors de la réalisation d'un **Simon** : un célèbre jeu sorti pour la première fois en 1978. Le concept est simple : le jeu compte quatre touches associées chacune à une couleur et un son distincts. Le Simon joue une première touche que le joueur doit reproduire dans un court délai. Si le joueur réussit, alors le jeu ajoute une touche aléatoire à sa séquence. Le joueur doit ainsi reproduire la séquence depuis le début, et ainsi de suite jusqu'à ce que le joueur se trompe.

"Simon" original

Notre version du Simon est articulé autour de 5 éléments principaux : un buzzer qui émet les sons, trois boutons avec LED intégrée (un bleu, un rouge et un jaune) et un afficheur 4 digits à 7 segments.

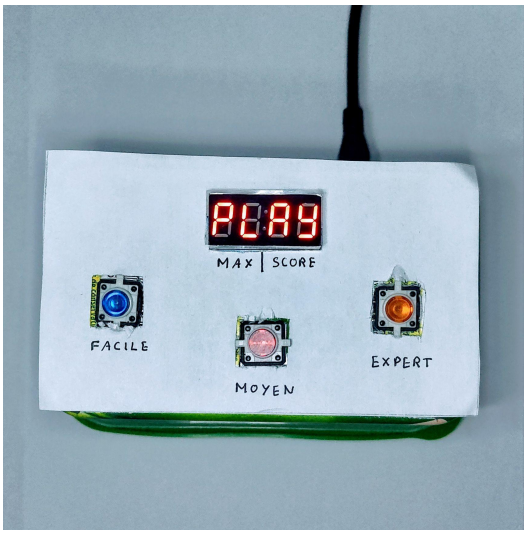
Une fois le Simon lancé, il vous faudra faire un premier choix, celui du mode de jeu. Comme indiqué sur l'afficheur, presser le bouton bleu lancera le mode "Facile" (jeu du Simon classique), presser le bouton rouge lancera le mode "Modéré" (même jeu mais en plus rapide) et presser le bouton jaune lancera le mode "Expert" (mode de jeu sans indications visuelles).

Une fois ce choix fait, les touches vont s'allumer successivement, indiquant le début du jeu, puis une séquence s' amorcera, qu'il faudra alors reproduire plus ou moins vite selon le mode choisi. Dans le cas du mode "Expert", une première séquence indiquant l'association son et lumière de chaque touche sera jouée, afin de permettre au joueur de repérer les touches avant de jouer à l'aveugle.

Lorsque l'on se trompe, le jeu est terminé, il faut à nouveau choisir un mode de jeu à l'aide des boutons puis recommencer.

L'afficheur indique dans un premier temps quel bouton permet de choisir quel mode, puis le niveau choisi et enfin en plein jeu, son score en temps réel (2 digits du cadran de droite de l'afficheur) et le score maximum atteint sur le mode de jeu en cours (2 digits du cadran de gauche).

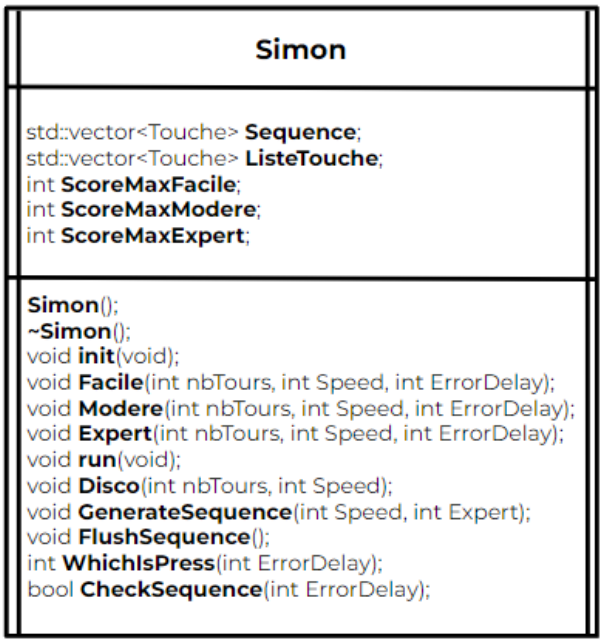
MICHEL Hugo
MASSON Alexandre

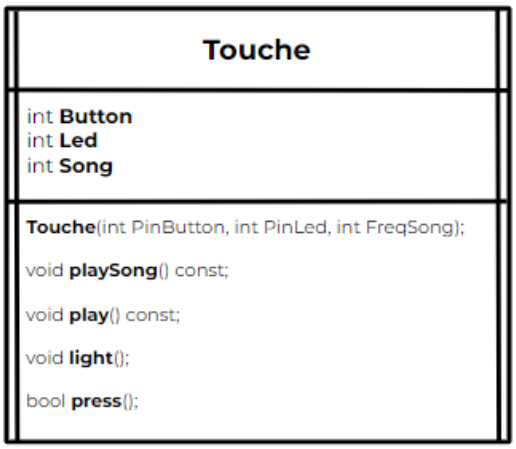


Maquette de notre Simon

Pour la réalisation de notre jeu, nous avons organisé notre programme en plusieurs classes :

- Une classe **Touche** qui représente les attributs et méthodes associés à une touche du jeu, chaque instance permet de créer un lien entre bouton, couleur et son. Cette disposition nous permet d'avoir autant de touches que voulu.
- Une classe **Simon** qui représente les attributs et les méthodes associés au fonctionnement interne de l'application, relatif à la séquence, et aux différents modes de jeu.





L'afficheur est quant à lui piloté par des fonctions de sa propre bibliothèque "TM1637.h".

Cette organisation en classe nous permet de n'avoir qu'une méthode **Simon::init()** qui fait le lien entre les PIN sur lesquelles sont branchés les composants et le programme en créant des instances de nos touches et en initialisant le buzzer. On trouve aussi une méthode **Simon::run()** qui lance un mode de jeu et fait tourner notre programme en boucle sur le microcontrôleur via notre fichier **Simon.ino** .

Il est intéressant de noter que notre système ne possédant pas de mémoire, le score maximum atteint par mode de jeu est remis à zéro à chaque mise hors tension du système. Nous n'avons également pas réussi à initialiser l'afficheur dans la partie init() de part la construction de notre programme. Enfin nous avons constaté qu'un temps d'attente trop long dans une boucle forcait le programme à redémarrer, nous obligeant à adapter notre code en fonction.

Notre version du jeu n'est d'ailleurs pas figée dans le temps puisqu'un ajout de boutons d'autre couleur est tout à fait envisageable (nous n'avions que ces trois de disponible au moment de la réalisation du projet), tout comme l'ajout d'un speaker, qui pourrait diffuser d'autres sons que les bip du buzzer. Nous avions aussi pensé à d'autre mode de jeu qui serait facilement implémentable au vue de l'architecture du système.

Ce projet était assez intéressant à mettre en œuvre, surtout pour ce qui est de la réflexion orientée objet pas toujours facile à adopter ou à mettre en place. Cependant au bout d'un certain temps passé à remodeler et repenser notre programme, nous avons réussi à aboutir à une version fonctionnelle et fun de notre Simon.