

Rapport Bureau d'Étude Programmation Orientée Objet Simon Game



Yanis Chala
Estelle Beneito
4AE-SE1
2021-2022

Introduction

Les objectifs de ce Bureau d'étude sont de réfléchir à une conception orientée objet, utiliser le langage C++, développer une bibliothèque modulaire, imaginer un problème à résoudre ou un nouveau service dans le monde des objets connectés et de l'Internet des Objets, et développer une application attractive pour utiliser notre bibliothèque.

Pour ce faire, nous disposons d'un microcontrôleur ESP8266 programmable en arduino ainsi que différents capteurs et actionneurs.

Pour mettre en pratique les notions vues en cours et en TD nous avons choisi de programmer un Simon game. Le jeu électronique éclaire une des quatre couleurs et produit un son toujours associé à cette couleur. Le joueur doit alors appuyer sur la touche de la couleur qui vient de s'allumer. Le jeu répète la même couleur et le même son, puis ajoute au hasard une nouvelle couleur. Le joueur doit reproduire cette nouvelle séquence. Chaque fois que le joueur reproduit correctement la séquence, le jeu ajoute une nouvelle couleur. Le but étant de faire la plus longue séquence possible.

Conception

Diagramme des cas d'utilisation

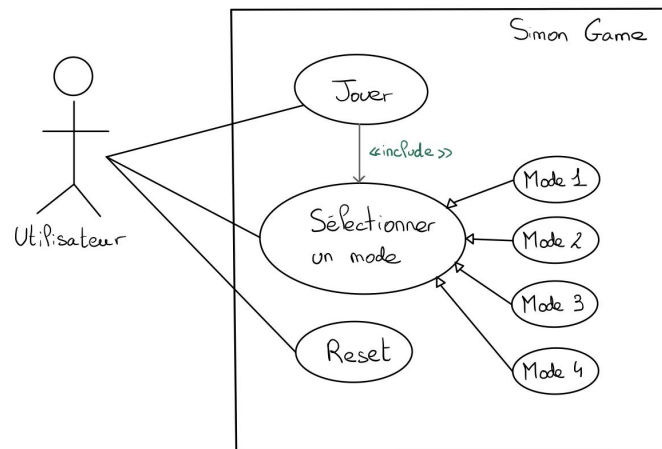


Figure 1 : Diagramme des cas d'utilisation du Simon Game

Diagramme de classe

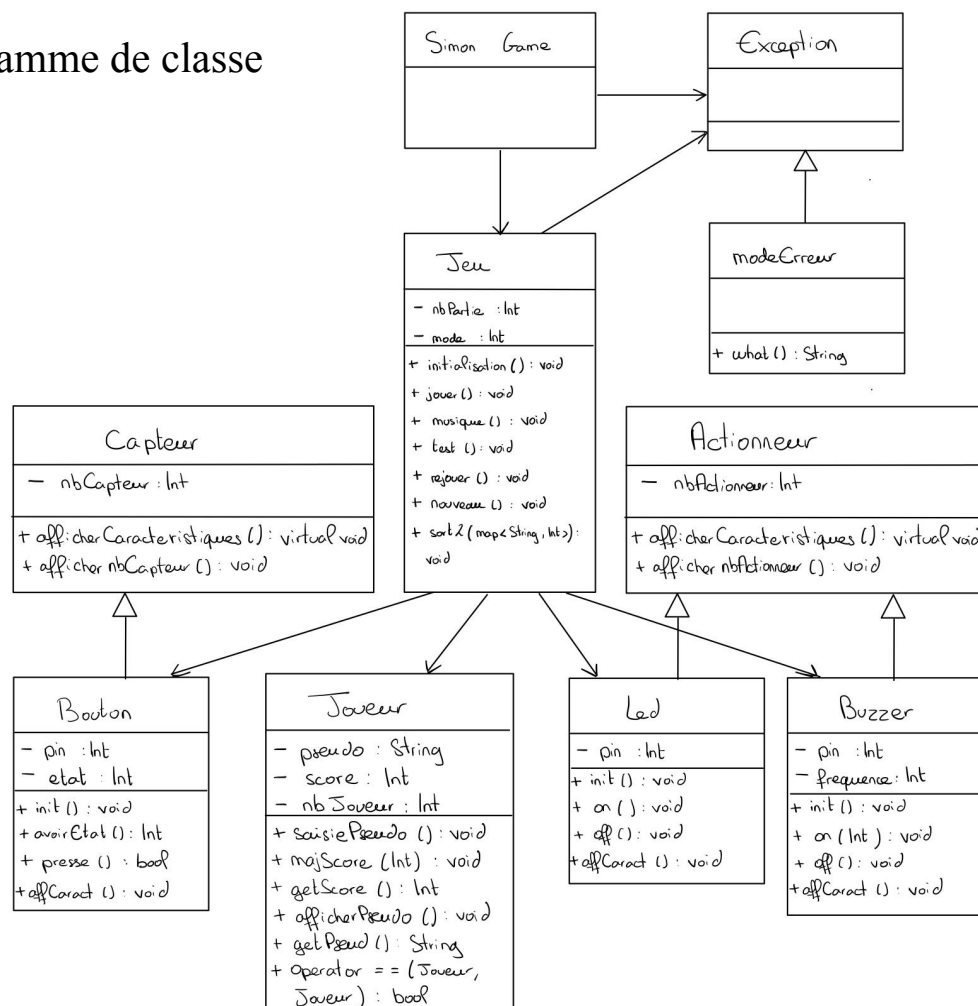


Figure 2 : Diagramme de classe du Simon Game

Nous n'avons pas représenté la classe Joystick car nous n'avons pas eu le temps de la programmer.

Déroulement du projet

Choix des composants

L'ESP8266 est un circuit intégré à microcontrôleur avec connexion Wi-Fi, possédant des interfaces I2C, SPI et UART. La carte ESP8266 offre 16 GPIOs décrit ci-dessous.

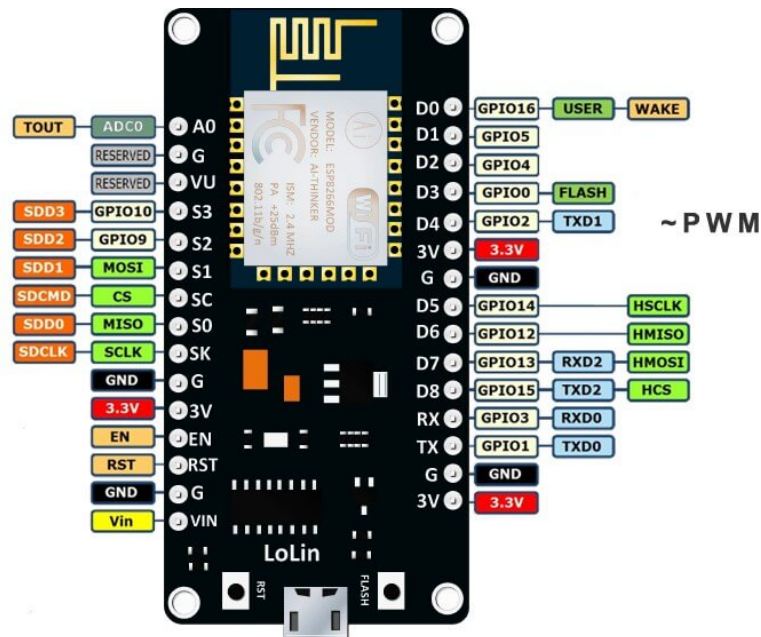


Figure 3 : Pinout ESP8266

Nous avons passé une commande de composant pour obtenir un button pad led 2x2. Malheureusement, les délais pour ce projet étant très court nous ne l'avons pas reçu à temps.

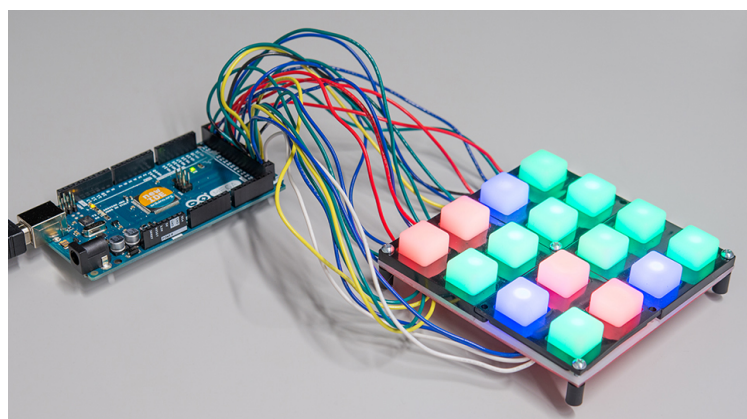


Figure 4 : 4x4 Button Pad avec Arduino Mega 2560

Nous nous sommes finalement contenté de 3 boutons à led de couleur rouge, bleu et orange.

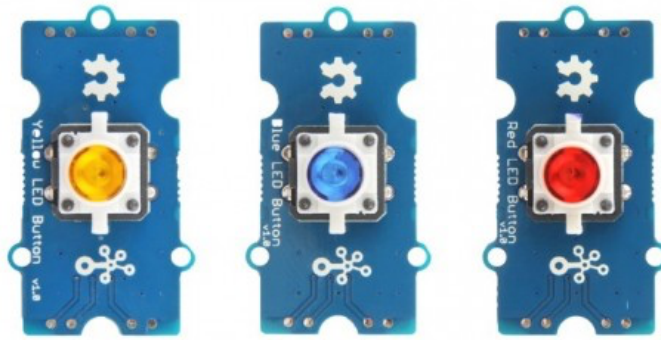


Figure 5 : Bouton poussoir avec led Grove

Programmation

Explication du projet :

Nous avons programmé notre système de telle sorte qu'il puisse offrir une interface pour l'utilisateur via le moniteur série. Un menu principal nous propose 4 modes différents :

- Le mode 1 permet de jouer au jeu "classique" : on entre notre pseudo puis le jeu se lance, le but étant de réussir à reproduire la plus longue série de couleurs. Si on arrive à battre le jeu (série de couleurs maximales = 25) ou si l'on se trompe, on pourra faire le choix de rejouer ou non. Dans le cas où l'on ne souhaite pas rejouer, un second joueur peut essayer de battre le meilleur score. Un classement des 2 meilleurs joueurs est affiché à la fin de chaque partie.
- Le mode 2 permet de jouer en 1 vs 1 : un joueur crée une série de couleurs avec un joystick et le second joueur reproduit la suite et ajoute une couleur, ainsi de suite jusqu'à ce que l'un des 2 joueurs se trompe.
- Le mode 3 permet de jouer à un jeu de rapidité jusqu'à 3 joueurs : chaque joueur choisit une couleur de bouton (bleu, rouge ou orange). Lorsque la led du milieu s'allume en vert, le plus rapide à appuyer sur son bouton gagne un point (attention la led peut feinter en s'allumant d'une autre couleur). Un joueur qui appuie au mauvais moment perd un point. Le premier joueur à atteindre 10 points remporte la victoire.
- Le mode 4 permet d'afficher les caractéristiques du système : on peut observer sur le moniteur série le nom et la pin des différents capteurs et actionneurs du système.

Respect du cahier des charges en terme d'utilisation du C++ :

Nous avons programmé notre projet en fonction du cahier des charges demandé :

- Création de plusieurs classes : nous avons créé au total 9 classes.
- Utilisation du mécanisme d'héritage : nous avons définis 2 classes mère (Capteur et Actionneur). Le mode 4 de notre système utilise les mécanismes d'héritage pour afficher les caractéristiques des différents capteurs et actionneurs.
- Redéfinition d'opérateur : nous avons redéfini l'opérateur == pour vérifier si le pseudo du joueur1 est différent de celui du joueur2.
- Utilisation de la STL : nous utilisons les library map et vector pour classer les joueurs en fonction de leur score.
- Utilisation des exceptions : nous avons créé une exception modeErreur. Lorsqu'un mauvais mode de jeu est sélectionné, on lève une exception.

Fonctionnement du système

Étapes à suivre pour mettre en marche notre système :

1. Téléchargez toutes les library et placez-les dans votre dossier library Arduino : Capteur(.h et .cpp) Actionneur(.h et .cpp) Bouton(.h et .cpp) Led(.h et .cpp) Buzzer(.h et .cpp) Joueur(.h et .cpp) Jeu(.h et .cpp) Joystick(.h et .cpp)
2. Ouvrez le main du projet avec Arduino : SimonGame.ino
3. Branchez le système à votre ordinateur (en considérant les capteurs et actionneurs déjà branchés)
4. Téléversez le programme et ouvrez le moniteur série sur 115200 baud.
5. Il ne vous reste plus qu'à suivre les instructions à l'écran.

Conclusion

En conclusion, notre système actuel est capable d'assurer les fonctions d'un Simon Game. C'est-à-dire permettre à l'utilisateur de reproduire une suite de couleurs aléatoire. Nous sommes allés encore plus loin en ajoutant des fonctionnalités du type l'affichage du classement des joueurs et des caractéristiques du système. Nous vous proposons une vidéo de fonctionnement du système à l'adresse suivante :

https://drive.google.com/file/d/1H-vvDBlUnBwCengaV_WWa5ZwuBkG6dpT/view?usp=sharing

Ce projet, pluridisciplinaire, a permis à l'ensemble de l'équipe de découvrir de nouveaux outils, de nouvelles technologies (la STL par exemple) mais aussi des nouveaux composants intéressants comme la carte ESP8266. Il nous a permis d'acquérir de nombreux atouts pour la suite de notre parcours, en nous permettant de maîtriser la programmation orientée objet.

Nous avons adoré travailler en équipe sur ce projet très libre où notre imagination est notre seule limite.

Difficultés rencontrées

Malheureusement, nous avons été rattrapés par le temps et nous n'avons pas terminé de programmer le mode 2 et 3 de notre système.

De plus, nous avons eu des difficultés sur l'utilisation de la STL car on s'est rendu compte qu'il était impossible de trier une map en fonction de sa valeur. Nous sommes donc passé par la déclaration d'un vecteur de paires.

Améliorations

Plusieurs pistes d'améliorations sont possibles pour notre système. Dans un premier temps, on pourrait programmer le mode 2 et 3 qui sont des modes multijoueur. Par la suite, on peut implémenter le button pad à la place des 3 boutons à led ce qui permettrait d'avoir une structure se rapprochant du jeu initial. Pour finir, on pourrait ajouter un écran à la place d'utiliser le moniteur série pour que le système devienne un peu plus autonome.