

Compte-rendu : BE de C++

Introduction

Ce rapport a pour but de documenter et d'expliquer nos choix de conception pour le bureau d'étude de conception orientée objet. Ce projet est un peu particulier car nous n'avons eu accès physiquement aux capteurs et actionneurs qu'à la dernière séance, dû à des commandes tardives. Nous avons voulu créer un jeu de société intéressant et innovant, où l'on utilise pour seul matériel un dé. Nous avons profité de la liberté créative que ce bureau d'étude nous autorise pour inventer un jeu un peu farfelu.

Présentation du jeu

Notre système est un dé connecté implémentant un jeu de bizkit. Il est constitué d'une carte ESP8266, cœur de notre programme, possédant une interface WIFI, de sept leds branchées en série, d'un capteur de luminosité et d'un écran LCD.

En début de partie, il faut paramétrer le jeu : un affichage sur le téléphone, connecté à la carte ESP8266 utilisée via WIFI, demande d'entrer le nombre de joueurs et le nom des joueurs. La partie se lance ensuite.

Lorsqu'on passe la main au-dessus du capteur de luminosité, celui-ci lance le dé, et le score du dé est affiché par des leds série. Chaque joueur lance deux dés et doit exécuter des actions selon les scores obtenus (les règles s'affichent en continu sur l'écran du téléphone). Par exemple, si le score des deux dés fait 7, c'est le bizkit, tout le monde doit crier "bizkit" et le dernier à la faire boit. Si le score fait 6, alors le joueur précédant le joueur en train de jouer boit. Au contraire, si le score fait 8, c'est le joueur suivant le joueur en train de jouer qui boit. Lors d'un tour, le nom du joueur en train de jouer et le score de ses dés s'affichent sur l'écran LCD, ce qui permet à tout le monde de savoir où on en est dans la partie. Lorsque le tour d'un joueur est fini, on passe au joueur suivant. La partie se termine lorsque la connexion téléphone est coupée.

Une règle particulière est lorsqu'un joueur fait un double 6 : il doit alors ajouter une nouvelle règle au jeu. Pour cela, il a deux options : il peut demander à implémenter une règle déjà programmée dans une base de donnée, ou bien il peut créer sa propre règle. Lorsqu'il choisit d'implémenter une règle, l'algorithme explore une base de règles inactives et choisit d'en activer une aléatoirement. Lorsqu'il choisit de créer une nouvelle règle, un affichage sur le téléphone apparaît. Il doit d'abord choisir entre trois types de règles possibles : les règles simples (lorsque la condition de la règle est atteinte, le joueur boit un certain nombre de gorgées), les règles avec rôle (lorsque la condition est atteinte, le joueur devient un certain rôle défini, et celui qui avait ce rôle précédemment le perd) et les règles

avec rôle récurrent (même chose que pour les règles avec rôle, sauf que lorsqu'une seconde condition est atteinte durant la partie, le joueur boit un certain nombre de gorgées même si ce n'est pas son tour). Il doit ensuite saisir les différents paramètres de sa règle (conditions de déclenchement, nom de la règle, nombre de gorgées données, condition sur le déclenchement du rôle récurrent), puis celle-ci est implémentée dans le jeu et est immédiatement active.

La figure 1 présente le diagramme de séquence d'un fonctionnement normal de l'application.

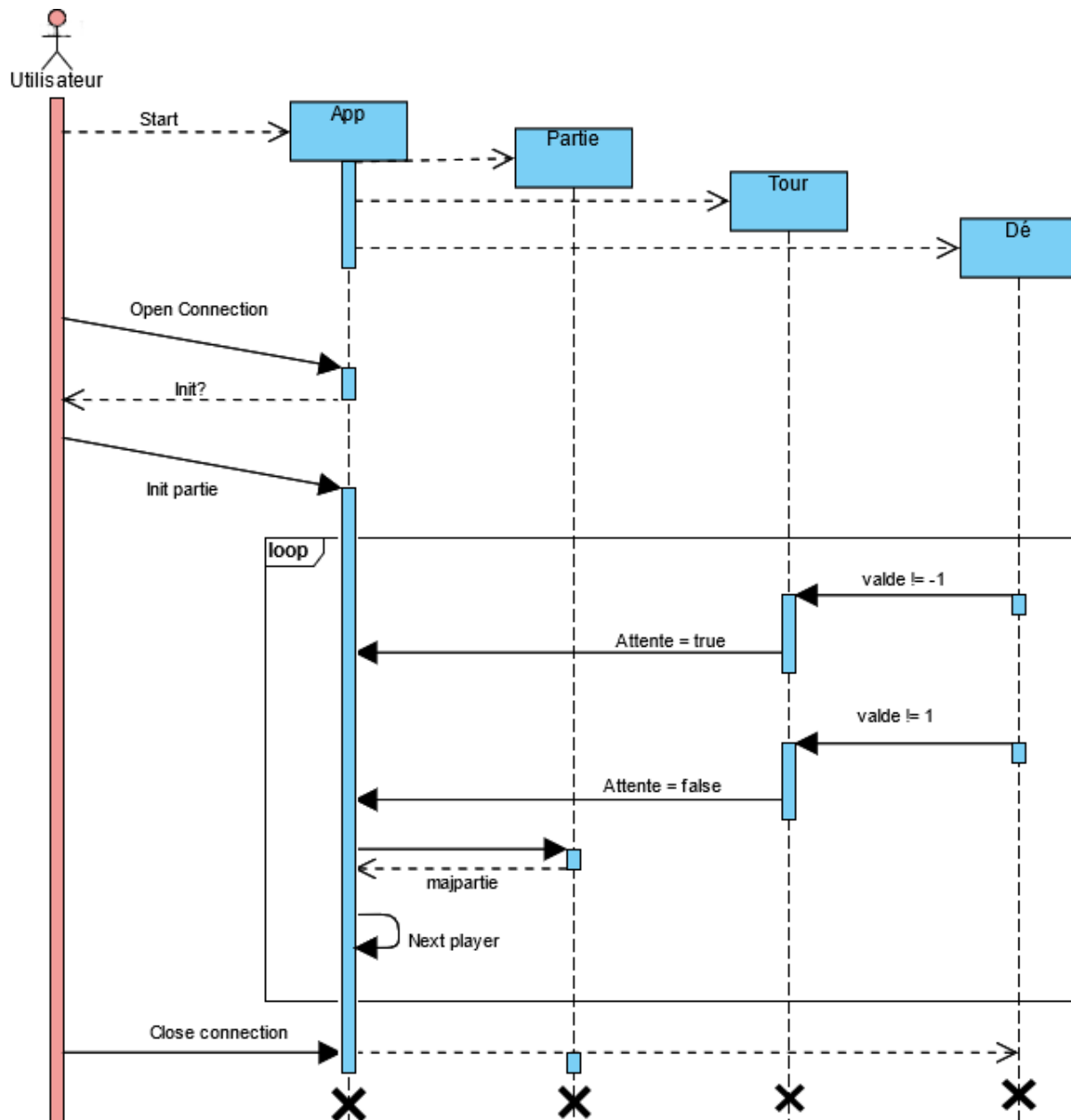


Figure 1 : Diagramme de séquence de l'application

Présentation des capteurs et actionneurs

Pour notre projet, nous devons choisir des capteurs et actionneurs adaptés.

Pour le capteur, nous nous sommes dirigés vers un capteur de lumière “Grove - Light Sensor v1.0” qui utilise une photorésistance pour détecter la luminosité dans l’environnement. Branché sur l’output A0, il donne la valeur analogique de la luminosité, qui sera comparée pour déclencher ou pas le lancer du dé.

Pour modéliser le dé en électronique, nous avons choisi sept LEDS “Grove-Chainable RGB LED” qui se connectent en série sur la pin D7/D8. Chaque LED est contrôlable individuellement et en fonction du score du dé, certaines LEDs s’allument.

Pour afficher les scores du dé en temps réel, nous avons opté pour un écran LCD “Grove - OLED Display 1.12” “ qui est branché sur la pin I2C. Il affiche le score des deux dés lancés par chaque joueur.

Liens vers les composants :

- Capteur de lumière : <https://www.seeedstudio.com/Grove-Light-Sensor-p-746.html>
- LED série : https://wiki.seeedstudio.com/Grove-Chainable_RGB_LED/
- Ecran LCD : https://wiki.seeedstudio.com/Grove-OLED_Display_1.12inch/

La partie affichage sur le téléphone

Pour afficher les règles et saisir les données nécessaires au bon fonctionnement du jeu, il est nécessaire de mettre en place un affichage sur le téléphone d'un des joueurs, qui sera le "maître du jeu". Nous avons pour cela utilisé la fonctionnalité WIFI du ESP8266. Dans notre programme, nous créons différentes pages HTML, qui sont ensuite envoyées sur l'adresse IP du ESP et permettent soit d'afficher les valeurs voulues, soit de les saisir. Le diagramme de séquence de la figure 2 résume le fonctionnement de ses pages.

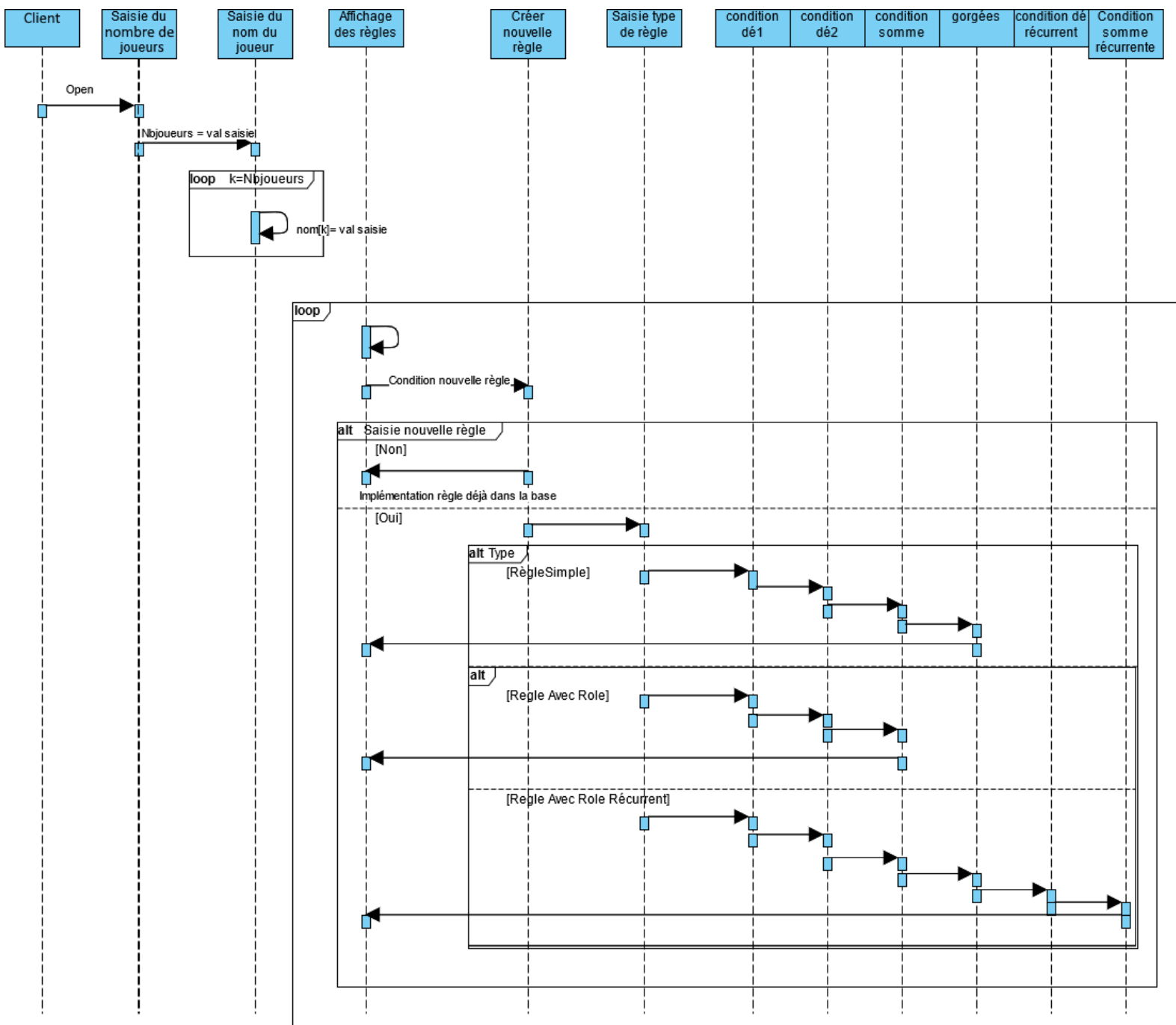


Figure 2 : Diagramme de séquence de l'évolution de l'affichage portable

A l'ouverture du wifi, la première page qui apparaît est celle de la sélection du nombre de joueurs. Cette valeur saisie, une nouvelle page permettant de saisir le nom de chaque joueur apparaît, et se réactualise tant que le nom de chaque joueur n'est pas donné.

On passe ensuite au jeu : une page d'affichage des règles est mise à l'écran, et se réactualise automatiquement toutes les secondes, pour mettre à jour la règle en cours.

Dans le cas où un joueur fait un double 6, une nouvelle page apparaît et il a deux choix : soit il crée une nouvelle règle, soit il implémente une règle déjà créée. Si il choisit d'implémenter une règle déjà créée, cette règle est activée et on revient à l'affichage des règles. S' il choisit de créer une nouvelle règle, on passe à l'affichage suivant, qui donne le choix du type de règle qu'il veut créer. Enfin, en fonction du type de règle qu'il a choisi, des pages permettant de saisir les paramètres nécessaires à définir apparaissent consécutivement, puis on revient à l'affichage des règles.

Implémentation des bibliothèques C++

Le diagramme de classe de la figure 3 présente les différentes bibliothèques de classe implémentées.

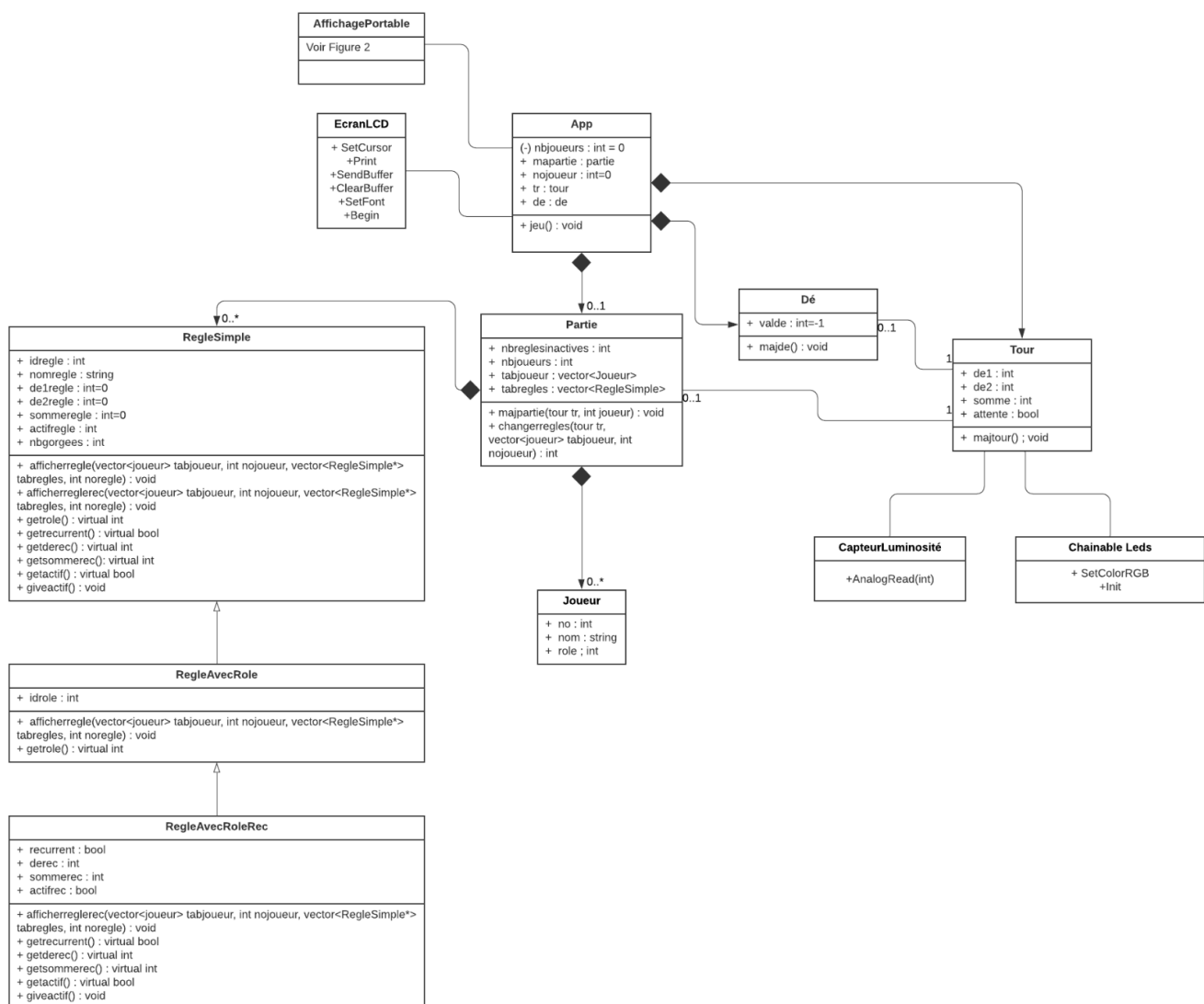


Figure 3 : Diagramme de classe du système complet

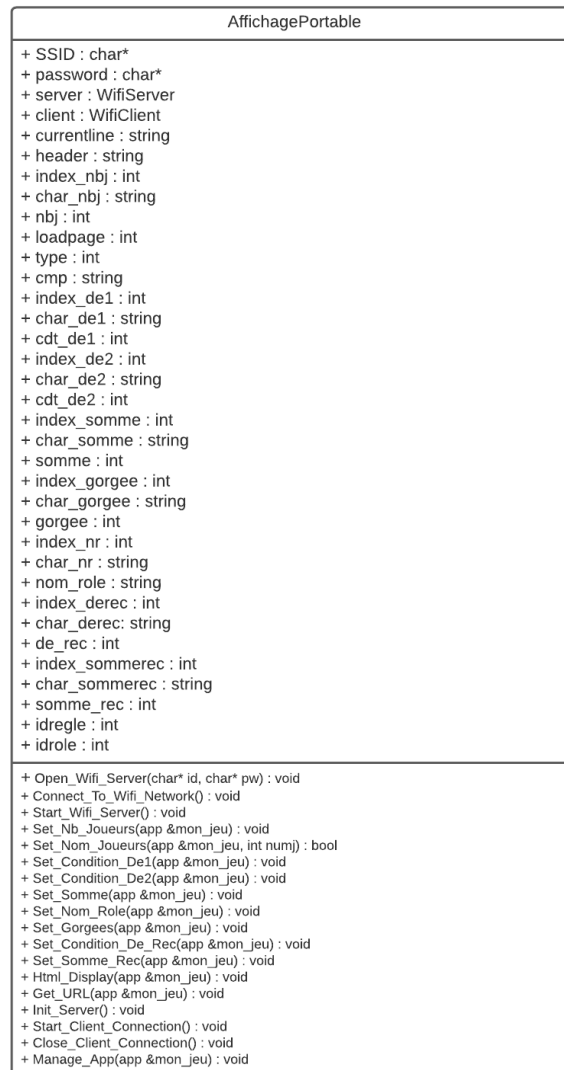


Figure 4 : Détail de la classe AffichagePortable

La classe RegleAvecRoleRec est une classe fille de RegleAvecRole, elle-même classe fille de RegleSimple. Ainsi les fonctions virtuelles d’affichage et de récupération des attributs sont redéfinies dans les classes filles si nécessaire.

Nous avons aussi utilisé des interfaces déjà implémentées dans Arduino pour utiliser le capteur, les LEDs et l’écran LCD. Ces classes n’existent donc pas réellement dans le programme, mais sont utilisées comme des bibliothèques

Conclusion

Malgré les conditions sanitaires difficiles et le retard des commandes des composants, nous avons pu mettre en place une application orientée objet, basée sur Arduino. Ce projet nous a donc permis de consolider nos compétences en C++ et en conception orientée objet. Au niveau de notre application, nous sommes globalement satisfaits, car le jeu est fonctionnel et tous les capteurs réagissent bien comme convenu.

Néanmoins, nous aurions pu optimiser un peu plus l'affichage sur l'écran LCD pour mettre plus d'informations. Autre perspective d'amélioration : permettre aux joueurs de décider du nombre de tours à l'initialisation de la partie, afin de fixer la durée de la partie. Une troisième perspective d'amélioration ambitieuse serait de rajouter d'autres jeux dans l'application.

Nous tenons à remercier fortement les encadrants de travaux pratiques pour leur disponibilité et réactivité tout au long du bureau d'étude, malgré le contexte actuel.