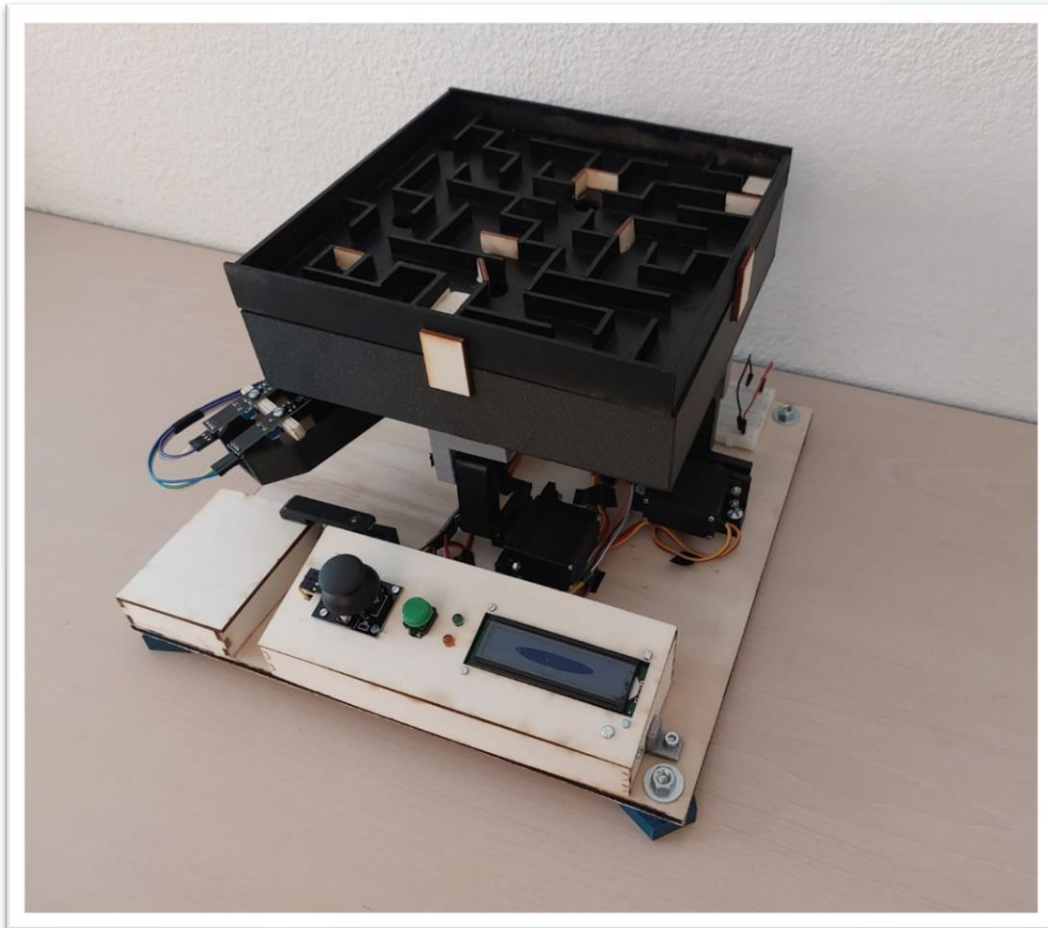


Maze Game

Julius Ortstadt



Sommaire

Introduction.....	2
Schéma électrique du projet	2
Algorithme de fonctionnement	3
Coût du projet.....	4
Planning.....	5
Problèmes et solutions.....	6
Conclusion et perspective	8
Bibliographie.....	8

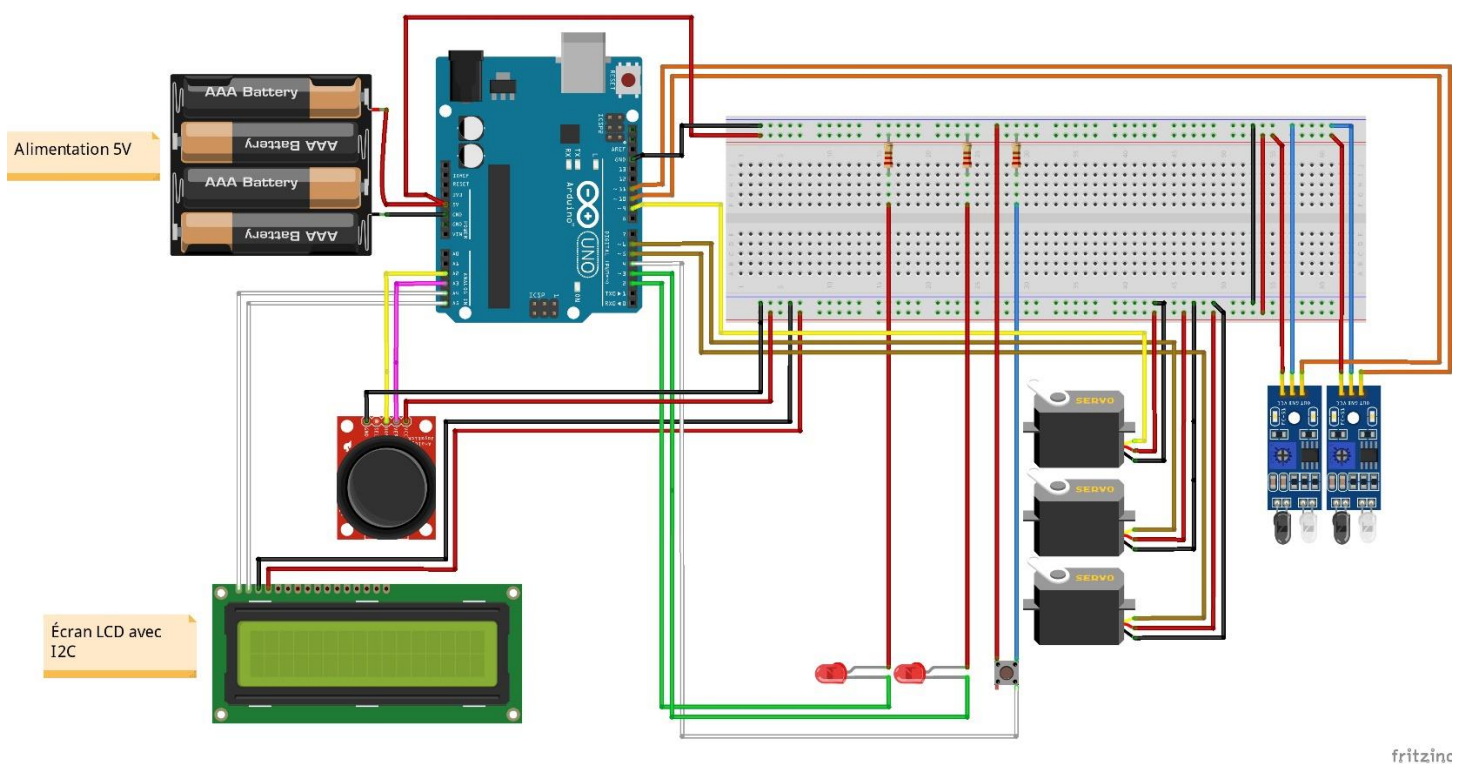
Introduction

Le projet Maze Game est un mini jeu de labyrinthe comme on peut déjà le connaître. Le principe est simple : on a une bille située sur un plateau de jeu et cette bille doit être naviguée par le joueur du point de départ au point d'arrivée.

Pour rendre le jeu plus dynamique et intéressant, Maze Game est automatisé. C'est-à-dire que pour pouvoir jouer, le joueur se laisse guidé par le menu et effectue les instructions qui lui sont demandées. Ensuite, la bille se libère et le joueur la place au point de départ du labyrinthe. Il l'amène au point d'arrivée grâce au joystick qui fait bouger le labyrinthe et ainsi la bille. Cependant il ne doit pas tomber dans un des trous perdants présent dans ce dernier. Une fois que le joueur a réussi à arriver au point d'arrivée, la bille tombera dans le trou situé à cet endroit et il aura gagné. La bille est bloquée et le jeu se termine et l'écran affiche le score de la partie, donc le temps mis pour gagner. Finalement, le joueur peut rejouer s'il le souhaite.

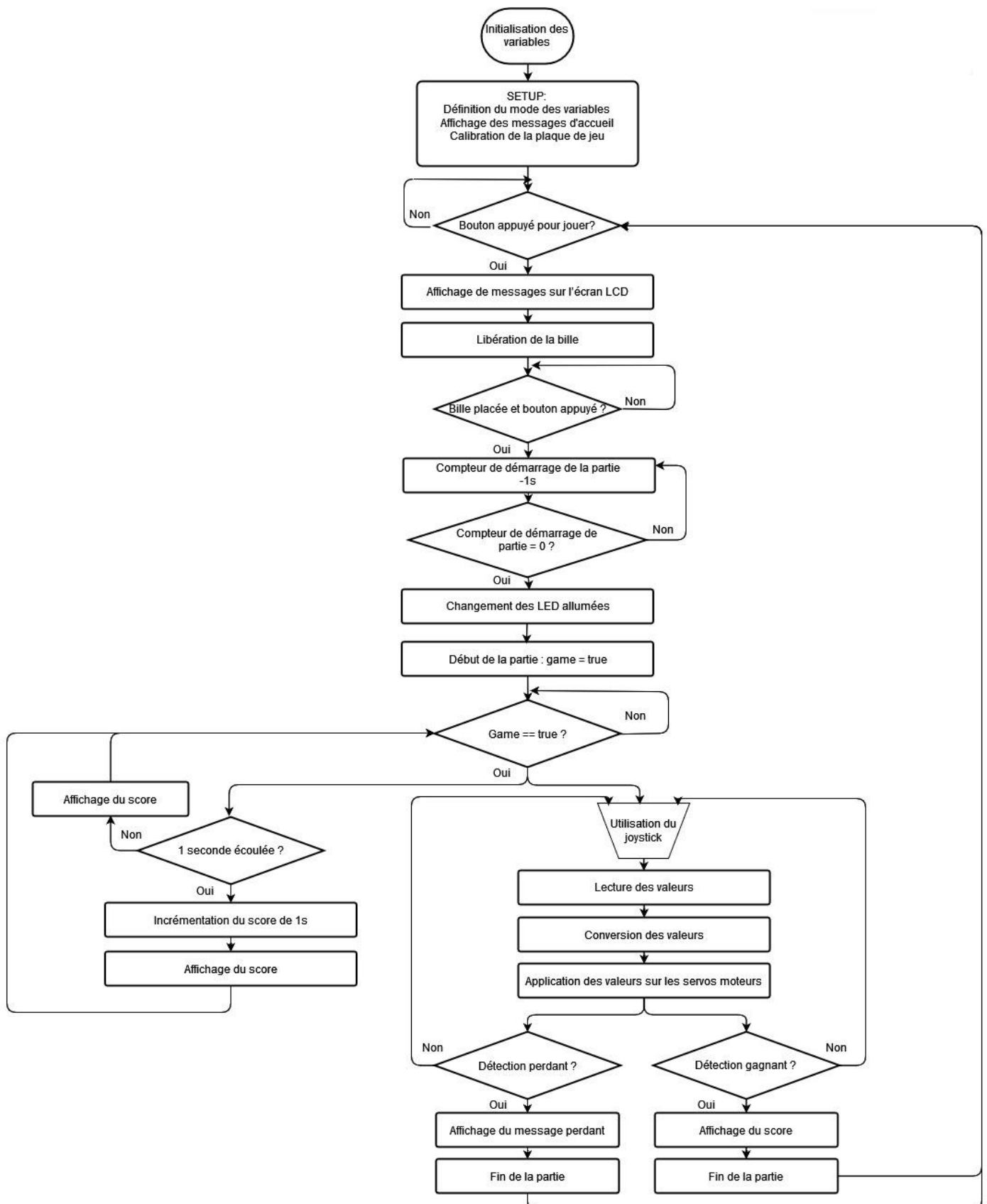
Schéma électrique du projet

Ci-contre est le schéma électrique du projet Maze Game. Ce dernier indique tous les branchements effectués pour que le jeu fonctionne.



Algorithme de fonctionnement

L'algorithme de fonctionnement pour le projet Maze Game explique le déroulement des étapes au sein du code. Il permet de comprendre le fonctionnement du code qui fait marcher le projet.



Coût du projet

Pour calculer le coût du projet Maze Game il est nécessaire de prendre en compte d'une part le coût matériel et d'autre part le coût ingénieur.

En ce qui concerne le coût matériel, le tableau ci-dessous résume le matériel utilisé ainsi que le prix estimé.

Matériel	Quantité	Prix total
Servo moteurs MG996R Tower Pro	2	18€
Servo moteur SG90	1	2€
Joystick KY-023	1	4€
Arduino UNO	1	20€
Breadboard	1	1€
LED	2	0,5€
Bouton poussoir	1	0,5€
Résistance	3	0,5€
Écran LCD avec Module I2C	1	10€
Alimentation 5V	1	8€
Bille en métal	1	2€
Capteurs infra-rouges	2	4€
Câbles	--	10€
Pièces en impression 3D	--	20€
Bois	--	5€
Coût total : 105,5€		

Ensuite pour le coût ingénieur :

Il y a eu 24 heures de travail réparties sur 8 séances de cours. De plus, il y a eu 30 heures de travail en dehors des cours, soit un total de 54 heures.

Ainsi avec un salaire brut de 38 000€ pour 1600h de travail, le coût final pour les heures de travail s'élève à 1.282,5 €.

Finalement le coût total du projet s'élève à 1.388 €.

Planning

Au niveau du planning il y a eu quelques variations entre le travail prévu pour la séance et le travail finalement réalisé.

Ci-contre le planning initial et les différences avec le planning final :

Séance n°	Planning initial	Planning final
Séance 1 – 12/12/2022	<ul style="list-style-type: none"> Esquisse du projet Liste du matériel Détermination des dimensions du projet 	<ul style="list-style-type: none"> Esquisse du projet Liste du matériel Détermination des dimensions du projet
Séance 2 – 19/12/2022	<ul style="list-style-type: none"> Début de la conception de la planche de jeu (design) Début de la modélisation des pièces 	<ul style="list-style-type: none"> Début de la conception de la planche de jeu (design) Début de la modélisation des pièces Début de la modélisation de la plaque de base et des pièces associées pour supporter le projet
Séance 3 – 09/01/2023	<ul style="list-style-type: none"> Schéma du montage électronique Test de certaines fonctionnalités 	<ul style="list-style-type: none"> Test de certaines fonctionnalités
Séance 4 – 16/01/2023	<ul style="list-style-type: none"> Codage des fonctionnalités de base (contrôle de la planche) 	<ul style="list-style-type: none"> Continuation du codage et test de certaines fonctionnalités du projet Conception des pièces pour le joint triple, la fixation des servo moteurs et la transmission du mouvement des moteurs à la plaque de jeu
Séance 5 – 02/02/2023	<ul style="list-style-type: none"> Codage des fonctionnalités de détection de début de jeu 	<ul style="list-style-type: none"> Conception du labyrinthe et découpage des pièces pour ce dernier Conception et découpage de la plaque recevant le mouvement des servo moteurs Codage et test de l'écran LCD Codage et test du joystick
Séance 6 – 09/02/2023	<ul style="list-style-type: none"> Production des pièces Début de l'assemblage 	<ul style="list-style-type: none"> Modélisation des pièces pour la transmission du mouvement des servos Conception et découpage du bloc menu

Séance 7 – 16/02/2023	<ul style="list-style-type: none"> • Intégration code-jeu • Test des fonctionnalités 	<ul style="list-style-type: none"> • Finalisation de l'assemblage du labyrinthe • Finalisation de l'assemblage du bloc de menu • Finalisation de l'assemblage du mécanisme de mouvement pour la plaque de jeu • Modélisation et impression du système de recollection pour la bille lorsque cette dernière tombe dans un trou
Séance 8 – 02/03/2023	<ul style="list-style-type: none"> • Finalisation du code • Finalisation de l'assemblage 	<ul style="list-style-type: none"> • Impression du bras de collection de la bille • Impression du système de fixation pour le servo du couvercle et du couvercle lui-même • Impression des dernières pièces • Soudure des composants • Fixations des composants • Continuation du développement du code
Séance 9 – 09/03/2023	<ul style="list-style-type: none"> • Soutenance 	<ul style="list-style-type: none"> • Soutenance

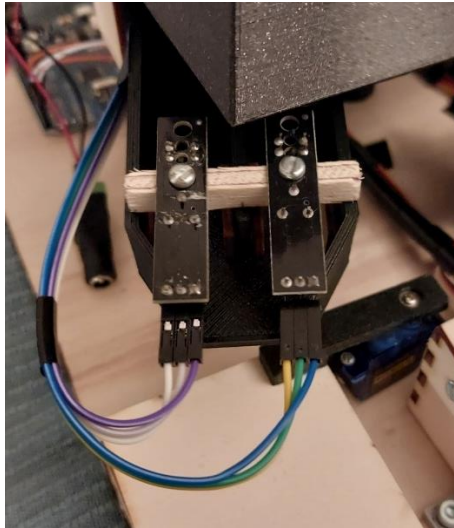
En ce qui concerne les différences, on peut voir que majoritairement les parties concernant la conception, la construction et l'assemblage ont pris un certain retard au cours des séances. Cela était notamment dû à certaines difficultés rencontrées au cours de ces tâches. Elles sont explicitées en détails dans la partie « Problèmes et solutions » de ce document. Principalement il s'agissait d'itérations sur le design de certaines pièces qui m'ont obligé à en changer d'autres. De plus, certains systèmes ne fonctionnaient pas comme prévu, il était donc nécessaire de faire certains ajustements.

Pour la partie code, il y avait certains obstacles qui ont aussi retarder le développement de ce dernier, ce qui c'est ainsi répercuter sur le reste du planning.

Problèmes et solutions

Lors de la réalisation du projet j'ai rencontré trois difficultés majeures.

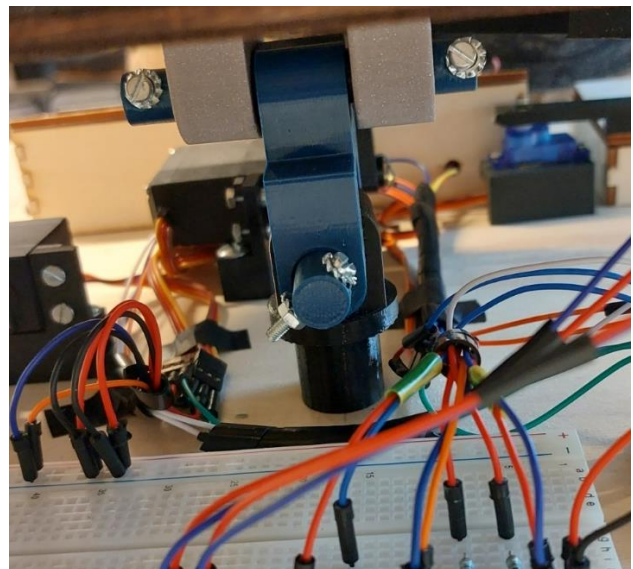
1/ La principale était de trouver une solution pour le système de détection et d'acheminement de la bille. Initialement j'avais prévu d'utiliser des capteurs à pas de course et de les fixés sur la plaque de base. Cependant le problème était d'amener la bille sur ces capteurs avec assez de puissance pour le déclencher. J'ai donc changé de concept et opté pour des capteurs infra-rouge fixés sur la plaque de base mais il était cependant plus avantageux de les fixés sur la plaque en mouvement pour détecter plus facilement la bille lorsqu'elle tombe dans un trou comme ci-contre :



2/ Ensuite, une autre difficulté était le joint pour le mouvement de la plaque. Initialement, je voulais utiliser un joint rotule imprimer en 3D. Cependant pour des raisons d'assemblage et de fixation, un joint triple comme ici, était plus adapté.



Joint à rotule



Joint triple utilisé

3/ Un dernier défi était la calibration des servos moteurs et du joystick. Notamment l'utilisation de `map()` pour convertir les valeurs. Le premier joystick ayant eu un défaut, a renvoyé des valeurs non constantes. Cependant après changement, les valeurs étaient constantes et il ne manquait plus qu'à calibrer les moteurs et adapter les valeurs pour que le labyrinthe soit bien horizontal.

Conclusion et perspective

Le projet Maze Game est un jeu de labyrinthe avec un but simple mais une conception un peu plus compliquée.

Afin de réaliser ce projet il était nécessaire de concevoir beaucoup de pièces parfois plus compliquées que d'autre. De plus, des itérations sur le design de ces pièces à amener à des changements dans le fonctionnement du jeu. Cela a eu pour but de rendre le jeu plus fluide et agréable à utiliser.

Toute la partie construction et assemblage a constitué une grande partie du projet. Avec un design assez compact il fallait que tout s'emboîte assez bien.

Finalement, le code qui fait fonctionner tout le projet a dû être fait et tester. Celui-ci a aussi nécessiter plusieurs changement afin que tout fonctionne comme il le fallait.

Toutes les fonctionnalités annoncées pour ce projet fonctionnent comme il le faut. Donc le contrôle du labyrinthe, le menu, la navigation à travers ce dernier, le système de détection et de capture de la bille et les diverses informations affichées par l'écran et les LED.

De plus, tous les codes et pièces pour ce projet, peuvent être retrouvé sur mon GitHub dont le lien se trouve dans la partie « Bibliographie ».

Avec 9 séances supplémentaires, j'ajouterai certaines fonctionnalités au projet dont :

- Un mode de jeu avec plusieurs vies.
- Un leaderboard pour comparer son score avec celui d'autres parties.
- Un couvercle en plexiglass pour le labyrinthe afin d'éviter la triche.
- Différents labyrinthes que l'on peut changer rapidement afin de diversifier le jeu.
- Recouvrir les câbles et l'électronique pour donner une meilleure apparence au projet.

Bibliographie

- Description de la bibliothèque Wire.h pour le module I2C :
<https://reference.arduino.cc/reference/en/language/functions/communication/wire/>
- Branchement de l'écran LCD avec le module I2C sur Arduino :
<https://arduino-france.site/lcd-1602/#3>
- Aide pour réaliser un compteur en utilisant la fonction millis() :
<https://home.et.utwente.nl/slootenvanf/2020/06/08/countdown-timers-tasks-in-parallel/>
- Description de la fonction map() :
<https://www.arduino.cc/reference/en/language/functions/math/map/>
- Lien vers le GitHub du projet Maze Game :
https://github.com/JuliusOrtstadt/Maze_Game