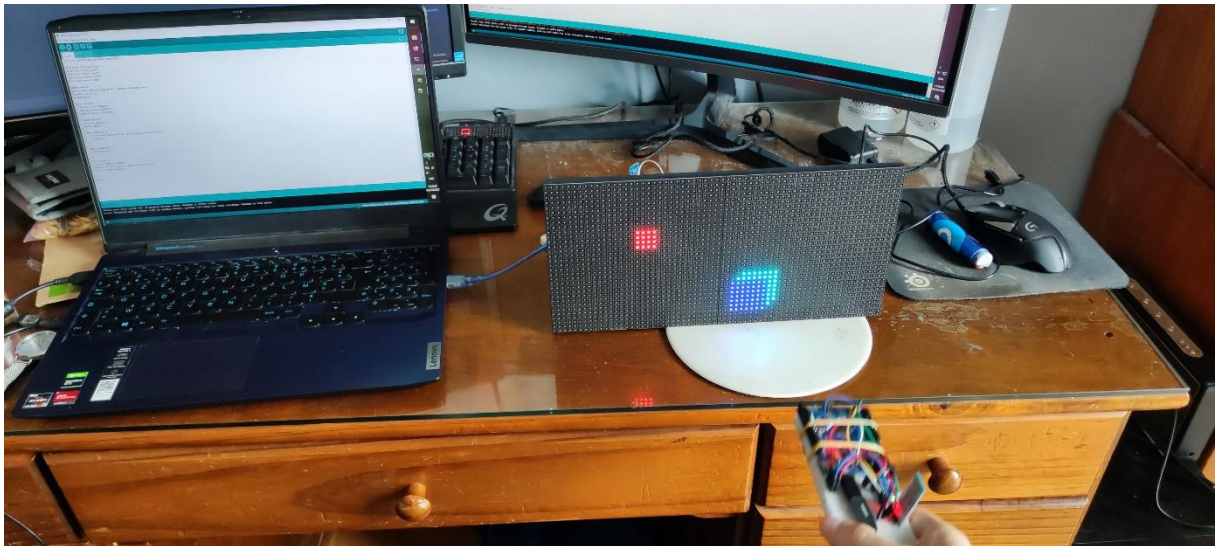


# PROJET ARDUINO : SABERDUINO

ANNÉE 2021-22 PEIP2



PHAN DAMIEN

EDGAR BIZEL

G1

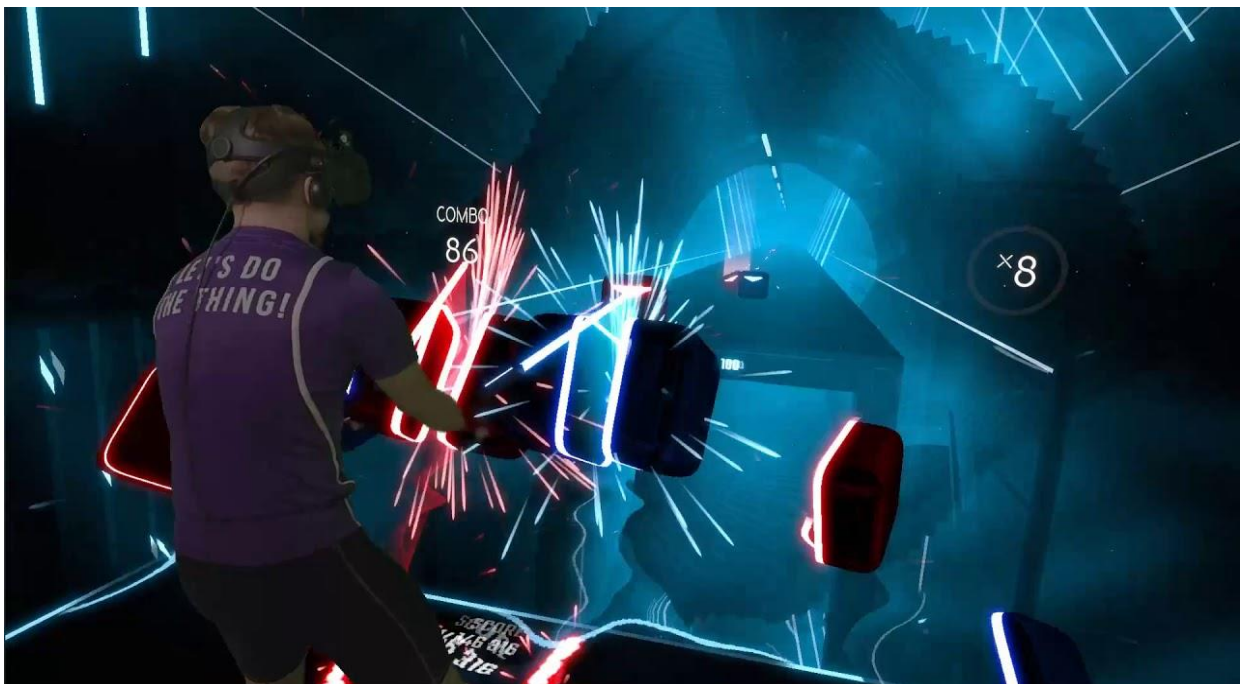
# SOMMAIRE

<b>INTRODUCTION</b>	<b>PAGE 3</b>
<b>PRÉSENTATION DU PROJET</b>	<b>PAGE 4-8</b>
• <b><u>Vision globale</u></b>	
• <b><u>Algorithme</u></b>	
• <b><u>Cahier des charges</u></b>	
• <b><u>Planning</u></b>	
<b>CONCLUSION</b>	<b>PAGE 9</b>
<b>BIBLIOGRAPHIE</b>	<b>PAGE 10</b>

## INTRODUCTION

Notre projet Saberduino a pour but de créer une copie du *Beat Saber*, un jeu existant en VR, avec des moyens plus abordables pour le consommateur.

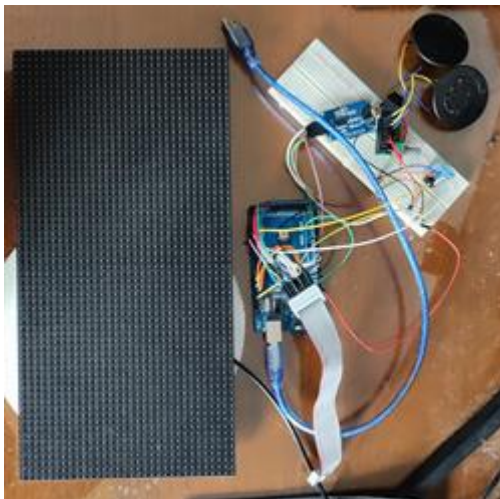
Le but de ce jeu est de pouvoir découper des blocs arrivant vers nous, suivant un rythme dépendant de la musique qui est jouée.



# PRÉSENTATION DU PROJET

- **Vision globale**

Notre projet se compose de deux modules. La console, responsable de l'affichage du jeu.  
Et le “sabre”, la manette servant à détecter les mouvements.

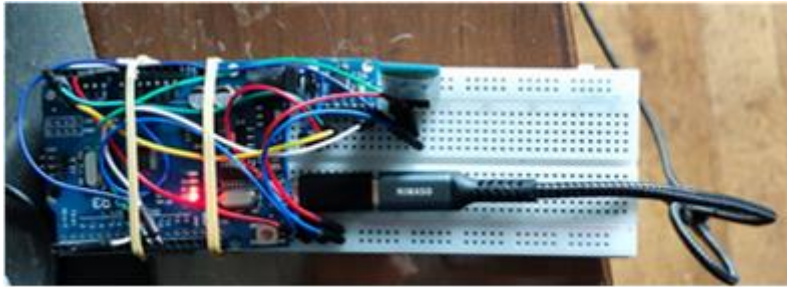


## Console

La console se présente sous la forme d'une matrice de LEDs, reliée à un Arduino Mega.

Nous pouvons également voir les haut-parleurs et le module Bluetooth.

La console est responsable de l'affichage du jeu. C'est donc cette puce Arduino Mega qui contient le plus de code C++.

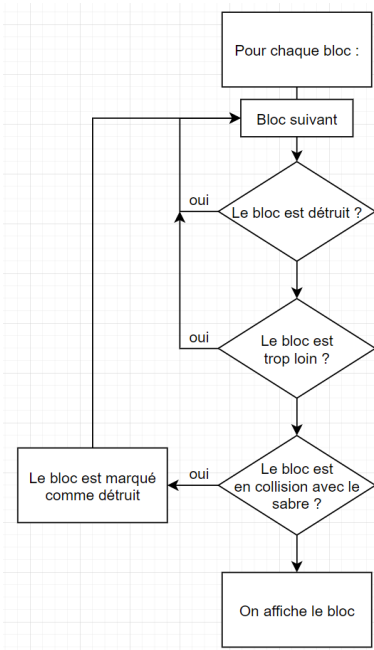


## Sabre

Le sabre est composé d'un Arduino Uno, d'un module Bluetooth et d'un accéléromètre-gyromètre GY-521. Il est responsable de la détection des mouvements du joueur.

### • Algorithme

En raison de la complexité du code, il nous est impossible de faire un algorithme complet. Voici donc la boucle principale, qui est répétée aussi souvent que possible :



Le principe est simple :

- Pour chaque bloc dans la liste d'attente, on regarde s'il est détruit, s'il va être détruit ou s'il est trop loin.

- Si l'une de ces conditions est remplie, le bloc est ignoré.

Sinon, on affiche le bloc.

## • Cahier des charges

Le cahier des charges original incluait la liste suivante :

### Pour le sabre :

- Communiquer par Bluetooth
- Signaler son inclinaison
- Signaler la direction du mouvement (gauche vers droite, haut vers bas...)
- Vibration lors du choc avec un bloc

### Pour la console :

- Afficher le jeu : des cubes, marqués par une flèche indiquant la direction de tranchage
- Recevoir les informations du sabre et les traiter
- Émettre de la musique

Nous avons réussi à respecter la majorité des conditions, mais nous avons dû effectuer quelques concessions.

La concession principale concerne les blocs : nous avons décidé de supprimer la direction de tranchage.

En effet, blocs étaient censés être marqués par une flèche.

Une flèche vers le haut, par exemple, aurait symbolisé que l'on devait détruire ce bloc en partant du bas et en arrivant en haut.

Cette fonctionnalité a été supprimée pour plusieurs raisons :

- Tout d'abord, l'accéléromètre que nous avons utilisé ne permet pas une précision suffisante.

Nous espérons pouvoir détecter sa position précise, mais cela n'est pas possible.



Pour cela, il faudrait utiliser un système de caméra optique ou infrarouge, comme l'a fait la console Nintendo Wii. Ce système est également utilisé par les casques de réalité virtuelle Windows Mixed Reality.

Nous avons donc décidé de quadriller l'espace de jeu en 9 carrés. Grâce à des calculs d'angle, nous sommes en mesure de dire dans quel carré la manette se trouve.

Ce système est bien moins précis que ce que nous souhaitions, mais il est fonctionnel.

- L'autre raison pour laquelle nous n'avons pas pu afficher la direction de tranchage est une raison graphique. En effet, l'écran que nous utilisons dispose uniquement de 32 par 64 pixels. C'est très faible, en comparaison à un écran moderne. En divisant la zone de jeu, cela revient environ à 12 pixels par 12 pixels pour chaque carré. C'est très insuffisant pour afficher des détails.

L'autre fonctionnalité que nous avons supprimée concerne le moteur à vibrations. La manette était censée vibrer lors de la collision avec un bloc. Nous n'avons pas eu le temps de l'implémenter, car nous avons passé trop de séances afin de faire fonctionner l'accéléromètre.

Voici donc le cahier des charges final :

#### Pour le sabre :

- Communiquer par Bluetooth
- Signaler son inclinaison

#### Pour la console :

- Afficher le jeu : des cubes, arrivant au bon moment
- Recevoir les informations du sabre et les traiter
- Émettre de la musique

## • Planning

Damien Phan	Séance 1	Séance 2	Séance 3	Séance 4 (ORAL)	Séance 5	Séance 6	Séance 7	Séance 8
Prise en main + compréhension de l'accéléromètre								
Programmation du module GY-521								
Bluetooth								
Montage Bluetooth + GY-521								
Finalisation								

Edgar Bizel	Séance 1	Séance 2	Séance 3	Séance 4 (ORAL)	Séance 5	Séance 6	Séance 7	Séance 8
Programmation : Base								
Programmation : Cubes								
Programmation : Affichage								
Mise en place de l'écran								
Programmation : Son								
Lecture des niveaux Beat Saber								
Bluetooth								
Finalisation								

Damien Phan	Séance 1	Séance 2	Séance 3	Séance 4 (ORAL)	Séance 5	Séance 6	Séance 7	Séance 8
Programmation : Base								
Prise en main + compréhension de l'accéléromètre								
Ajout de la batterie								
Moteur de vibration								
Montage du sabre								
Communication avec le jeu								
Finalisation								

Edgar Bizel	Séance 1	Séance 2	Séance 3	Séance 4 (ORAL)	Séance 5	Séance 6	Séance 7	Séance 8
Programmation : Base								
Programmation : Affichage								
Mise en place de l'écran								
Programmation : Son								
Ajout du module de son à l'arduino								
Finalisation								

### Damien Phan

Je n'ai pas pu respecter mon planning. Comme expliqué dans la partie cahier des charges, je n'ai pas pu ajouter le moteur à vibrations.

De plus, je n'ai pas pu procéder au montage du sabre. J'ai passé énormément de temps à appréhender l'accéléromètre.

En effet, la difficulté a été de trouver un bon exemple pour récupérer les données qui m'intéressaient.

Le montage électronique était simple. Mais tous les programmes que je trouvais donnaient des résultats non concluants.

C'est seulement à la fin que je suis parvenu à obtenir des données correctes et plus utilisables.

### Edgar Bizel

Je suis parvenu à respecter la majorité des attentes, mais mon planning a été quelque peu bouleversé.

Le planning original n'était pas très intelligent : il prévoyait de connecter l'écran une fois la programmation de l'affichage faite. Mais il est difficile de programmer sans pouvoir tester le résultat. J'ai donc préféré connecter l'écran dès que possible.

Il y avait également certains éléments que je n'avais pas envisagés. Par exemple, la lecture de niveaux Beat Saber. En effet, pour que le jeu fonctionne, il fallait des niveaux. J'ai préféré réutiliser ceux de Beat Saber plutôt que d'en créer, mais cela a pris du temps.



## CONCLUSION

Pour conclure, notre projet nous a permis d'appréhender le monde des ingénieurs qui nous attends.

Nous avons pu au travers de ce projet apprendre le travail en groupe et le respect de délais ainsi que l'appréhension des problèmes.

Nous tenons à remercier aussi les professeurs qui nous ont aidé durant toute la durée du projet sans quoi nous nous serions perdus et n'aurions pas pu terminer à temps.

## BIBLIOGRAPHIE

Demande d'aide pour relier la Teensy à une matrice de LEDs :

<https://forum.pjrc.com/threads/68968-Teensy-4-SmartLED-SD-Speaker-Bluetooth?p=295468#post295468>

Connexion de la matrice de LEDs :

<https://learn.adafruit.com/32x16-32x32-rgb-led-matrix/overview>

Format de niveau utilisé par Beat Saber :

<https://bsmg.wiki/mapping/map-format.html>

Format JSON :

<http://json.org/json-fr.html>

Guide HC-05 :

<https://arduino.stackexchange.com/questions/50974/how-to-solve-problem-atinq-error-1f-atinit-error17-on-bluetooth-module-hc/55264#55264>

Utilisation du Gy-521 :

<https://www.youtube.com/watch?v=0XxF2NyqmQo&t=78s>

<http://gilles.thebault.free.fr/spip.php?article32>

<http://users.polytech.unice.fr/~pmasson/Enseignement/Elements%20de%20Robotique%20avec%20arduino%20-%20Accelerometre%20-%20Projection%20-%20MASSON.pdf>