

Hoareau Grégory
Passin-Cauneau Barthélemy
PeiP2 Groupe 1
Promotion : 2017-2018

Rapport de Projet

Nom du Projet : Retrolytech



Matière : Électronique avec Arduino

Encadrant du projet :
Pascal Masson et Fabien Ferrero

Sommaire

I) Introduction

Matériel utilisé

Répartition du travail

Calendrier

II) Fonctionnalités présentes

III) Schéma de la borne

IV) Problèmes rencontrés

V) Améliorations possibles

I. Introduction

Notre projet en électronique cette année était de construire une borne d'arcade sur laquelle on a la possibilité de jouer à deux joueurs et que la totalité des jeux présents sur la borne d'arcade soient programmés par nous-même.

Il était initialement prévu qu'il y ait sur la borne d'arcade 4 jeux :

- Deux Pongs (un pour un joueur et l'autre pour deux joueurs)
- Un jeu de labyrinthe en coopération (appelé « Maze »)
- Et un jeu du genre d'Escape Game se déroulant dans le campus de

Polytech

Cependant au vu du temps, nous avons décidé de seulement faire trois jeux (les deux Pongs et le Maze) mais nous pensons continuer de programmer des jeux pour la borne d'arcade pendant nos années d'études à Polytech.

Pourquoi avoir choisi ce sujet ?

Nous avons choisi de faire ce projet car, plus tard, nous aimerions tout deux travailler dans le monde du jeux vidéos. Nous sommes passionnés par la programmation et ce sujet est idéal pour nous améliorer dans ce domaine tout en travaillant sur quelque chose que l'on aime.

Le matériel utilisé :

« *physique* » :

- Une Raspberry Pi3 B
- 2 joysticks
- 8 boutons
- Un encodeur USB
- Un écran Eyoyo [1024*768]
- Des planches de bois

« *informatique* » :

- Processing
- AntiMicro
- Ubuntu Mate

Division du travail sur le projet :

Nous avons choisi de nous diviser le travail sur le projet de la manière suivante :

Barthélemy :

- Installation sur Raspberry des différentes applications utiles
- Câblage des boutons et des Joysticks
- Codage du Pong à deux joueurs
- Codage de l'applet permettant de générer les maps du Maze
- Constructions de la borne d'arcade
- Debug des Pong

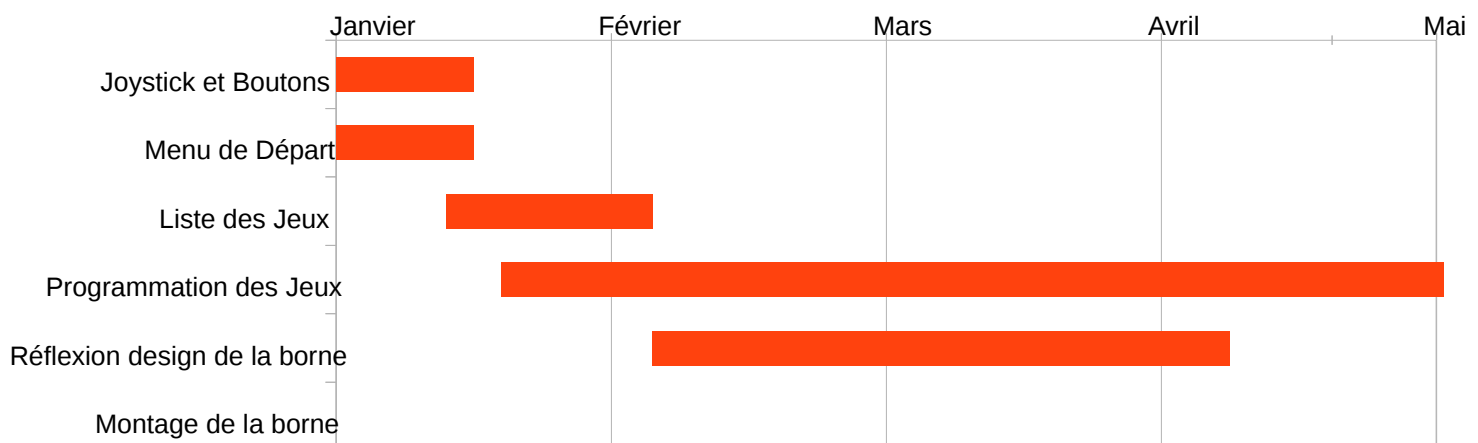
Grégory :

- Codage du menu de la borne d'arcade
- Codage du Pong à un joueur
- Codage du Maze
- Codage de l'envoi de mail pour le Pong (à un et deux joueurs)
- Récupération des ressources graphique nécessaire au Maze

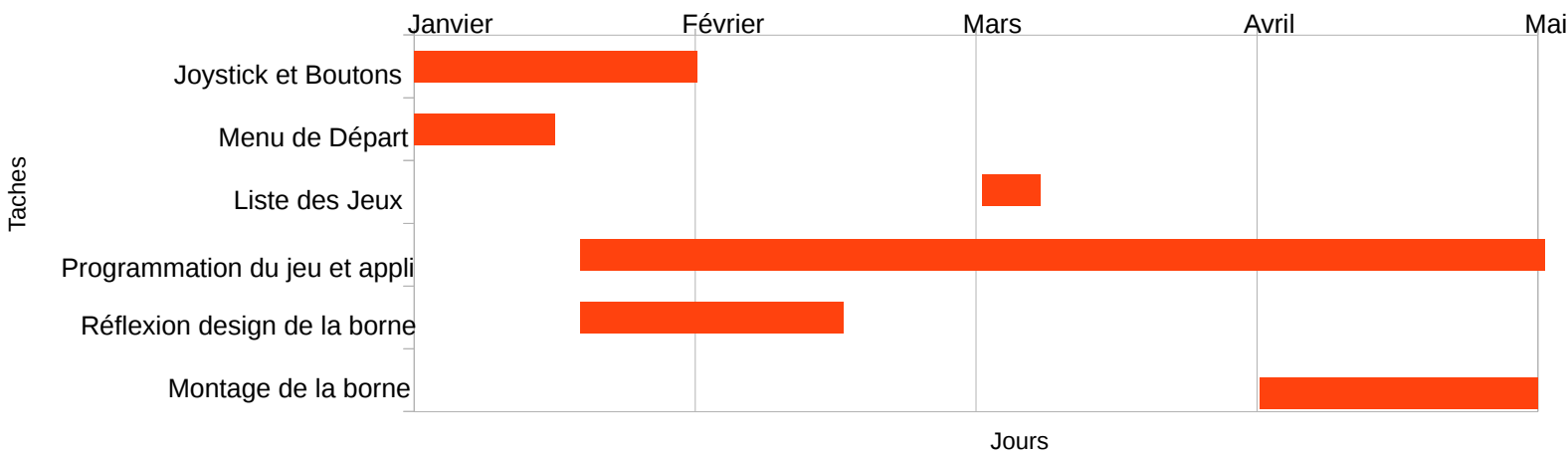
Diviser le travail de cette manière nous permettait de garder une indépendance entre nous. Chacun pouvait avancer de son côté à son rythme pour qu'à la fin, nous mettions en commun notre travail.

Calendrier :

Prévisionnel :



Réal :



II. Fonctionnalités présentes

Notre borne d'arcade contient finalement :

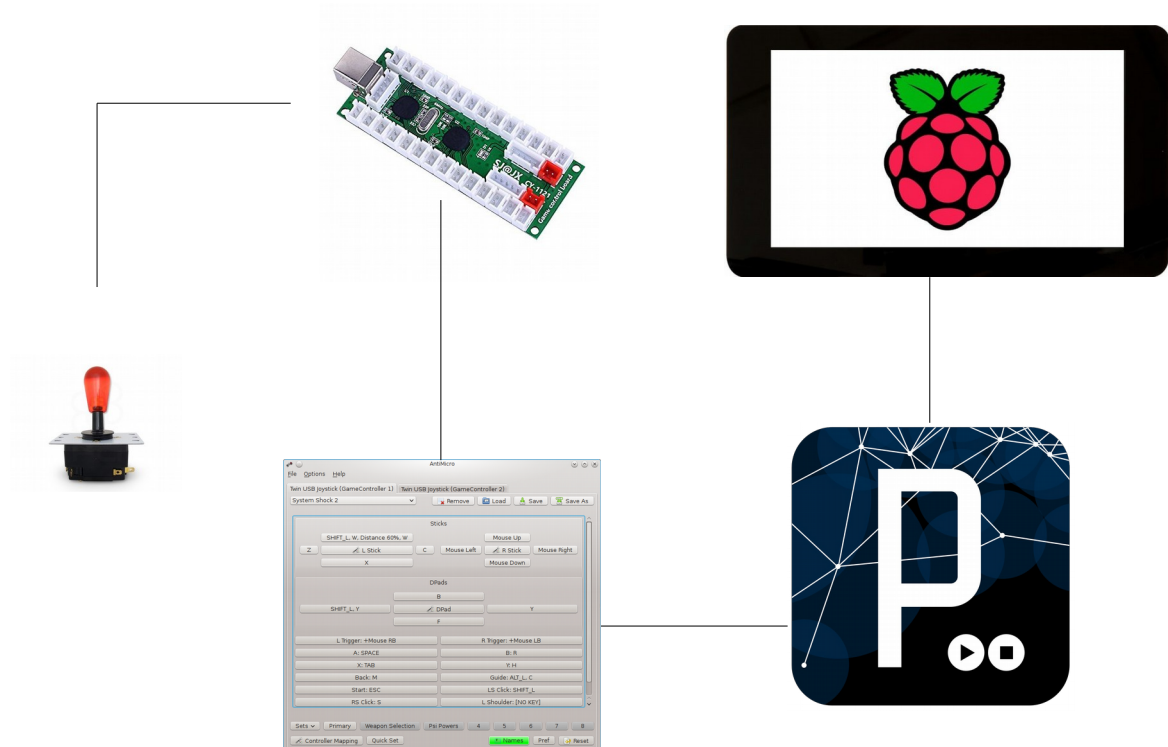
- Un menu dans lequel on peut choisir le jeu auquel on veut jouer.

- Trois jeux : les 2 Pong et le Maze, mais à l'heure de l'écriture de ce compte-rendu ce dernier n'a qu'un seul niveau.

- Il y a également la possibilité sur le Pong (un joueur et deux joueurs), que la borne d'arcade envoie à un joueur le score fait durant la partie.

Le joueur n'a qu'à rentrer son adresse mail, et la borne lui envoie un mail avec le récapitulatif de la partie.

III. Schéma de la borne



Lorsque l'on appuie sur un joystick (ou un bouton), ce dernier envoie un signal à l'encodeur USB. Ce dernier reçoit ce signal et le transmet à la Raspberry en lui associant le numéro sur lequel l'élément est connecté, puis c'est au tour d'Antimicro de jouer : ce logiciel affiche tout les emplacements de l'encodeur USB puis associe à chacun une touche du clavier. Ainsi, lorsque l'on joue au Pong, pousser le joystick du Joueur 1 vers le haut va être traduit par la touche « z », touche qui dans le programme, permet de monter la raquette du Joueur 1.

IV. Problèmes rencontrés

Dès le début, il était évident pour nous que nous n'allions pas utiliser une Arduino pour notre projet : Il nous fallait quelque chose de plus puissant, qui avait une interface graphique et un OS.

Nous nous sommes mis d'accord que nous allions travailler avec une Raspberry. Une fois nos Raspberry et un écran (temporaire) en main, les soucis commencent...

Un début difficile

-Durant les premières semaines du projet nous avons rencontré quelques problèmes.

Au début nos Raspberry ne s'allumaient pas. Puis après une réinitialisation de la carte SD, l'OS précédemment installé ne pouvait pas faire tourner Processing.

No Entiendo

-Une fois l'OS et Processing installés sur la Raspberry, on s'est rendu compte que les boutons et les joysticks, lors de leurs utilisation, renvoyaient des chaînes de caractère incompréhensible pour Processing. Après plusieurs recherches, nous avons trouvé une application qui permet d'affecter, à chaque boutons et aux directions des joysticks, des touches du clavier : Antimicro. Mais cette application ne pouvait pas être utilisée avec l'OS que l'on utilisait, nous avons donc du changer d'OS.

Ce n'est qu'au bout de nombreuses installations/désinstallations, que nous avons trouvé un OS capable de faire tourner Processing et AntiMicro : Ubuntu Mate.

Deux joueurs ?

-Notre projet est une borne d'arcade pour 2 joueurs, cependant nous nous sommes rendu compte que Processing ne pouvait pas, « naturellement », prendre en compte deux déplacements simultanés. En effet, lorsqu'une touche du clavier (pour nous, un joystick/bouton) est appuyée, Processing stocke la valeur de la touche dans une variable appelée « key ». Cependant, si on appuie sur une autre touche, l'ancienne n'est plus considérée comme appuyée et la nouvelle prend sa place. Ainsi en jouant au Pong à 2 joueurs, tant que l'un déplace sa raquette, l'autre ne peut pas bouger la sienne.

Ce problème est apparu alors qu'il ne nous restait que 2 semaines avant la fin des projet... Nous avons vite cherché à régler ce problème en procédant à un « switch/case » dans la fonction keyPressed du jeu (cf. Code).

La Balle Fantôme

-Lors du Pong (un joueur), la balle accélère dès qu'elle touche la raquette et le mur de gauche. Cependant à un moment la vitesse de la balle sera telle que à un instant t , la balle sera à gauche de la raquette mais à l'instant suivant la balle se retrouvera de l'autre côté de la raquette sans que la fonction de rebond ne s'applique.

Ce problème ne peut malheureusement pas être corrigé, car la Raspberry n'est pas assez puissante pour calculer aussi vite. C'est pour cela que lorsque l'on joue au Pong, la balle peut très bien rebondir comme elle peut traverser la raquette, à vitesse lente ou rapide...

V. Améliorations possible

Les améliorations présentes dans cette section sont des idées (qui n'étaient pas présentes dans nos objectifs initiaux) que nous avons eu soit trop tard (et par conséquent nous n'avons pas eu le temps de les implémenter) ou qu'elles semblaient nous prendre trop de temps et qu'elles n'étaient pas nécessaires au fonctionnement de la borne.

Nous avons, comme cité plus haut, l'intention de continuer de coder des jeux pour la borne (et donc les ajouter à cette dernière) mais également de compléter le jeu « Maze » en y ajoutant d'autres niveaux dans lesquels il y aurait les boutons (qui permettent d'ouvrir des portes situées dans le labyrinthe de son allié).

Le fait d'avoir simplifié l'ajout de nouvelles cartes au jeu nous permettra à l'avenir de rajouter du contenu plus facilement.

Nous avons pensé que nous pourrions ajouter du sons sur les jeux (par exemple : pour les Pongs rajouter des bruits lors des rebonds de la balle ou une musique de fond pour le Maze).

L'idée d'ajouter des bandes de LED qui, au moment de nouveaux High Score, de victoire au Pong, de fin d'un jeu, se mettraient à clignoter comme ce qui pouvait se trouver sur les anciennes borne d'arcade.