

The Amazing Project – Console de Jeu

Compte-rendu final

Dans ce compte-rendu nous allons parler du déroulement du projet. On va vous faire comprendre ce que c'est comme console, comment elle fonctionne et le déroulement de sa création. On va diviser ce rapport en 3 parties importantes qu'on abordera en profondeur. Pour commencer nous allons parler de son fonctionnement. Puis nous allons voir les étapes principales de son montage et ses différents éléments. Et finalement, nous allons expliquer quels sont les problèmes rencontrés au cours de ce projet et comment on les a surmontés.

1. Le code et son fonctionnement

A partir du début de l'année toutes les manipulations ont été faites avec la carte Arduino. La carte Arduino possède son propre langage avec les différentes fonctions qu'on peut utiliser pour des nombreuses composantes. Vu que la carte Arduino n'est pas pratique pour ce qui est le graphisme, nous nous sommes penchés vers la carte Raspberry Pi 3. Elle-même n'a pas son propre langage unique comme la carte Arduino. Avec la carte Raspberry il est possible de choisir le langage de programmation préféré. Donc le choix du Processing a été vite fait. Il est très pratique pour créer/coder ses propres jeux (faciles ou complexes) qu'on peut ensuite utiliser dans le projet.

Nous allons maintenant voir comment est construit le code pour notre console de jeu. Nous n'allons pas nous intéresser à la partie qui concerne le codage du jeu lui-même mais plutôt à celle qui est liée à l'électronique. Dans notre cas, les extraits du code qui touchent à l'électronique sont ceux qui s'occupent des boutons poussoirs.

Exemple :

```
//Code :  
void setup() {  
  
//PORTS POUR LES BOUTONS : //touches directionnelles :  
GPIO.pinMode(4, GPIO.INPUT); //haut --> donc inutile ici  
GPIO.pinMode(5, GPIO.INPUT); //gauche  
GPIO.pinMode(6, GPIO.INPUT); //bas --> donc inutile ici  
GPIO.pinMode(7, GPIO.INPUT); //droite //autres boutons :  
GPIO.pinMode(8, GPIO.INPUT); //a  
GPIO.pinMode(9, GPIO.INPUT); //b  
GPIO.pinMode(10, GPIO.INPUT); //x  
GPIO.pinMode(11, GPIO.INPUT); //y  
//Fin du code
```

On peut vite voir certaines ressemblances du code écrit avec Processing (pour la Raspberry Pi 3) avec celui écrit pour la carte Arduino. Tout comme pour l'Arduino, on doit faire un « setup » où on précise si un GPIO (anglais : general purpose input/output) agit comme une entrée ou une sortie en donnant aussi le numéro correspondant.

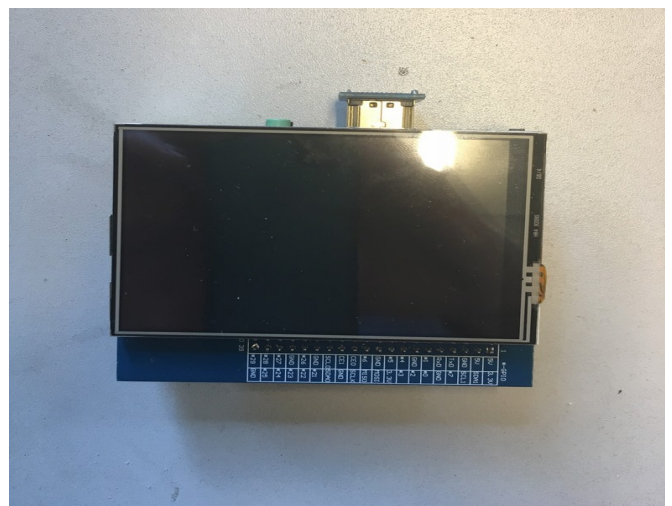
```
//Code :  
//BOUTONS RASPBERRY :  
if (GPIO.digitalRead(5) == GPIO.HIGH) { //gauche  
xRaq=xRaq-speedRaq;  
}  
if (GPIO.digitalRead(7) == GPIO.HIGH) { //droite  
xRaq=xRaq+speedRaq;  
}  
//Fin du code
```

Ici, de même qu'avec la carte Arduino on lit sur chaque GPIO utilisé son état correspondant (HIGH ou LOW). Nous pouvons voir que si le GPIO (exemple : 5) respecte la condition imposée, alors la variable xRaq prend une certaine valeur qui est directement utilisée dans le code.

Dans notre console nous utilisons le jeu le plus basique qui soit, Pong. Donc si la première condition ci-dessus est respectée, le bouton connecté à GPIO 5 est appuyé, alors la barre sur l'écran se déplace vers la gauche. De même, si le bouton 7 est appuyé, alors la barre va vers la droite.

2. Composition et montage

L'élément essentiel de notre console de jeu est l'écran tactile 3.5 pouces que monsieur Masson nous a commandé après notre demande. L'avantage de cet écran est bien évidemment sa taille car notre était d'avoir une console portable de plutôt petites dimensions. De plus, ce qui nous a attiré dans cet écran est le fait qu'il se pose directement sur les GPIO de la carte Raspberry. Cela nous évite d'avoir la nappe et le module qui connectent l'écran avec la carte. Donc par conséquent, nous diminuons l'épaisseur et la largeur de la console. Pour le cas où tous les GPIOs sont pris par l'écran sont prévus les GPIO mis en parallèle sur la carte de l'écran. Ceux deniers peuvent s'utiliser exactement comme les entrées/sorties de la carte Raspberry.

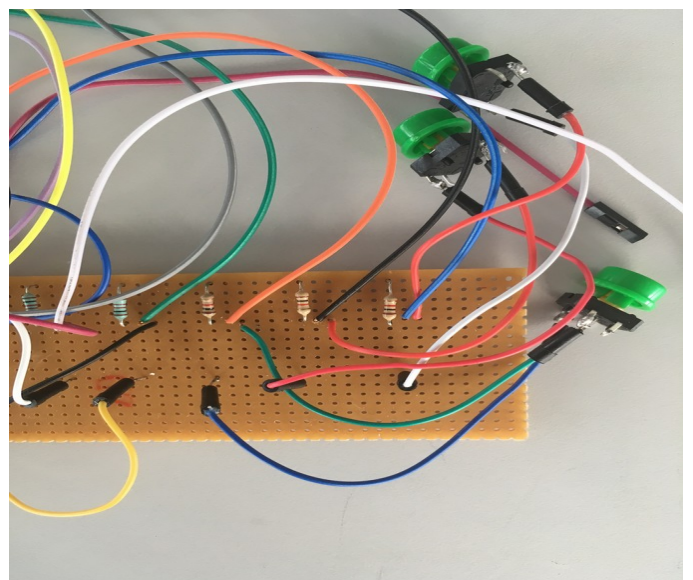
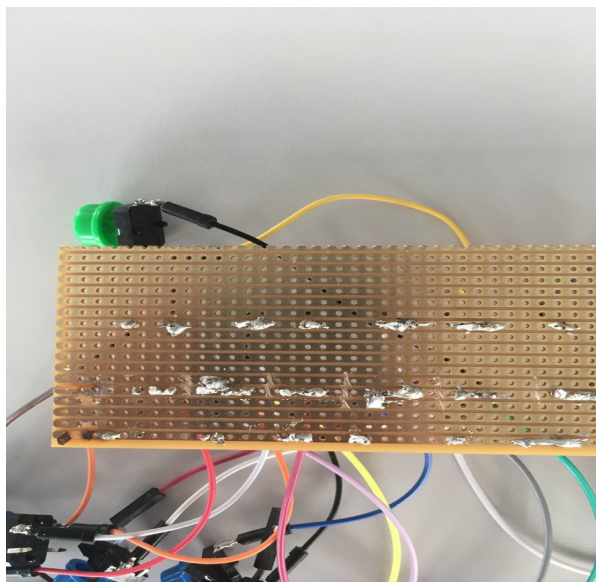


Comme les boutons poussoirs sont d'assez petite taille pour servir de touches dans la console portable, nous avons dû commander les autres

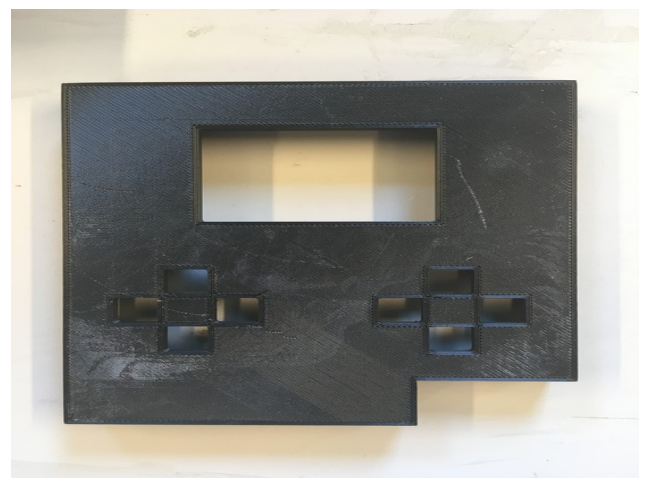
boutons poussoirs qui sont plus grands et possèdent des capuchons de 2 couleurs (vert et bleu). Même si le jeu codé, Pong, ne nécessite que 2 boutons pour aller à gauche ou à droite, nous avons à notre disposition 8 boutons. Ils sont disposés de façon suivante : 4 boutons à gauche et 4 boutons à droite. On peut voir en détail leur disposition sur l'image ci-dessous.



Pour ce qui concerne le montage des boutons, nous avons utilisé une platine d'essai afin de souder les fils (qui connectent la platine avec la carte Raspberry et avec les boutons poussoirs), ainsi que les résistances d'environ 100 Ohm.



Ensuite l'emballage/boîte dans laquelle on cache toutes les composantes qui font fonctionner la console a été modélisée grâce à un logiciel en ligne. Ensuite cette boîte a été imprimée avec l'imprimante 3D au Fablab. Elle est composée de 2 pièces symétriques. La seule différence sont les trous laissés pour les boutons, l'écran et l'accès à la batterie pour qu'on puisse alimenter la carte Raspberry qui se trouvent sur la partie supérieure et les côtés. Les images de ces 2 pièces on peut voir ci-dessous.



Enfin il ne reste plus qu'à assembler toutes les pièces de la console à l'aide de la colle et la patafix afin d'éviter le déplacement indésirable des pièces à l'intérieur de la console.

3. Problèmes rencontrés

Comme on l'a dit dans l'introduction il y a eu des problèmes auxquels nous avons été confronté lors de l'avancement du projet.

Tout d'abord, le plus grand d'entre eux était lié à l'environnement de la carte que nous avons utilisé pour ce projet. Une fois le système

opérationnel installé, nous avons eu constamment des bugs lorsqu'on voulait faire des modifications là-dessus. Par exemple, tout au début l'écran ne réagissait pas au toucher car cette fonctionnalité n'était pas

active. En ayant activé cette-dernière, tout semble bien marcher jusqu'à ce que la barre de menu qui est très utile pour faire des réglages disparaisse. A sa place on a retrouvé la partie noire de l'écran.

En pensant que le problème (absence de la barre de menu) consiste en le tactile, nous avons essayé de refaire tout à nouveau sans activer le tactile de l'écran. Cela s'est produit encore une fois. Suite à plusieurs essais de réinstaller le système sur la carte, le problème survenait tout le temps. Il était impossible de trouver l'information utile à propos de cela.

Puis comme l'écran est d'assez petite taille, la résolution par défaut n'était bien adaptée pour voir tout parfaitement en détail. Les autres résolutions proposées sur la carte ne l'étaient pas non plus.

Finalement, l'envie de construire une console qui ne soit pas de très grandes dimensions nous a posé beaucoup de contraintes. La plus importante d'entre elles est bien évidemment relativement petit espace à l'intérieur de la boîte. Cela ne nous a pas permis d'utiliser des différents modules qui auraient pu diversifier les fonctionnalités de la console de jeu.

Donc, en étant confronté à tous ces problèmes, nous n'avons pas pu avancer avec la vitesse prévue au début du projet (expliquée dans le planning). Jusqu'au dernier moment nous avons eu des problèmes venant de la carte Raspberry ce qui nous empêchait de bien travailler. Cela s'est traduit par le fait que les derniers tests et l'assemblage de la console avec tout le matériel nécessaire a été fait pendant la dernière semaine.

Conclusion :

Ce projet a été très intéressant pour nous en terme de travail en binôme. Nous avons appris de travailler de façon autonome chez nous et en TD. Malgré les difficultés que nous avons eu lors du projet on ne retient que du positif. Nous trouvons que le projet en électronique est une excellente idée. Cette expérience va nous servir beaucoup lors des nombreux projets qu'on aura pendant le cycle d'ingénieur peu importe la spécialité choisie. Nous sommes vraiment reconnaissant de nous avoir permis de le réaliser.