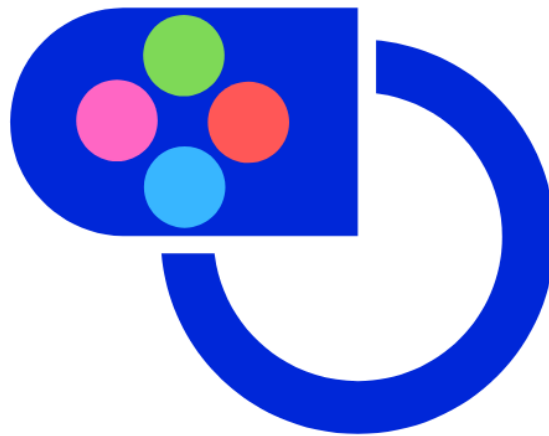


# Manuel d'utilisateur

UnijeV



Rédigé par : Groupe FA01

Steven Blouin  
Samuel Labonté  
Christophe Michaud-Lavoie  
Mouhamadou Lamine Sakho  
Manal Teto  
Vincent Bichard

# Sommaire

Il est possible de résumer l'ensemble de notre travail en quelques lignes. Dans un premier temps nous avons eu un entretien avec le client durant lequel nous avons empathisé avec lui dans le but de comprendre son besoin le plus clairement possible. Ensuite nous avons interprété son besoin afin de de créer un énoncé du problème clair et concis : Réaliser une manette permettant de jouer à tous les jeux de la PS4 avec une seule main. Par la suite nous avons, après une phase d'étalonnage proposé différentes solutions répondant au problème énoncé plus tôt. Enfin nous avons hiérarchisé les solutions grâce à une matrice décisionnelle afin d'identifier la plus efficace. Après nous avons développé la solution choisie plus précisément en développant plusieurs prototypes (trois exactement) avant d'arriver à notre prototype final qui répond clairement au problème du client. Dans la suite de ce manuel nous allons détailler ce processus de conception.

# Table des matières

Sommaire	i
Table des matières	ii
Listes de figures	iii
Introduction	1
Attributs du produit	2
1 - Mécanique	2
Mécanique électronique	2
Coquille extérieure	4
2 - Électronique et logiciel	6
3 - Électrique	11
Conclusion et recommandations	14
Appendices	15
Appendices A - Logiciels employés	15

# Listes de figures

Figure 1 - Diagramme des plaques de boutons et des fosses imprimées en 3-D	3
Figure 2 - Diagramme du mécanisme des boutons sur le manche de la manette	6
Figure 3 - Joystick analogique et sorties électriques	8
Figure 4 - Senseur de pression analogique et sorties électriques	9
Figure 5 - Diagramme de circuit analogique pour deux boutons numériques	9
Figure 6 - Schéma technique de la plaque contenant tous les circuits et modules	11
Figure 7 - Diagramme des connexions entre circuits et plaques de boutons	11
Figure 8 - Schéma de la nomenclature de chaque bouton correspondant sur la manette	12

# Introduction

Lorsqu'un produit est conçu, il est souvent conçu pour un marché en particulier. Des problèmes surviennent lorsqu'un différent marché est intéressé à ce produit mais ne peut l'utiliser à cause de contrainte majeure. Il est facile de justifier de telle action puisqu'un produit est conçu pour le plus gros marché possible mais ceci est injuste pour ceux qui n'ont pas la chance d'utiliser le produit. Qu'arrive-t-il lorsqu'une certaine personne change de marché du jour au lendemain et n'est plus capable d'utiliser un produit qu'elle utilisait tous les jours avant? Notre produit a pour but de permettre d'inclure plusieurs marchés pour que personne ne soit forcée d'arrêter une activité jadis adorée.

Les jeux vidéo sont conçus pour une clientèle ayant une bonne dextérité avec les deux mains. Donc une personne ayant seulement une main n'est plus considérée comme étant un client potentiel pour ce produit. Le UnijeV pour PS4 est conçu pour permettre aux personnes ayant perdu l'habileté d'utiliser une des deux mains. L'UnijeV inclut un plus gros marché pour permettre à tout le monde de jouer au jeu vidéo. La manette UnijeV pour PS4 est la seule solution sur le marché qui permet aux personnes avec seulement une main de jouer aux jeux vidéo et qui comprend tous les boutons d'une manette ordinaire de PS4.

Dans ce manuel, toutes les spécifications techniques du produit, ses fonctionnalités et ses capacités seront expliquées à fond, à l'aide d'un nombre d'explications et de schémas. Des recommandations sur de meilleures techniques de conception et de gestion de temps découleront également de ces explications.

# Attributs du produit

Ce produit est une intégration de plusieurs systèmes électriques, électroniques, mécaniques et logiciels. Tous ces systèmes ont pour but de créer une expérience tactile et réactive aux commandes de l'utilisateur. Cette section sera donc divisée en plusieurs parties, chacune touchant l'un de ces systèmes, son fonctionnement et son intégration avec les autres systèmes comportant le tout du produit.

La manette est conçue de façon à ce qu'elle soit entièrement utilisable à l'étendu maximal d'une manette de jeux vidéo standard avec une seule main. Ceci est accompli en employant non seulement les pouces et index de l'utilisateur, comme le font une manette typique à deux mains, mais aussi le poignet et le doigt du milieu de la main utilisable. Lors de la conception de la manette, ces concepts se trouvaient au cœur du développement de solutions et de spécifications techniques.

Ainsi, tous les boutons d'une manette de PS4 régulière sont présents, bien que leur format soit légèrement différent dans certains cas.

## 1 - Mécanique

La portion physique mécanique de la manette consiste en deux parties distinctes : la mécanique liée à l'électronique, et la coquille de la manette. La première permet de directement manipuler les signaux électroniques se dirigeant au module Arduino Micro, tel que les joysticks et les boutons. La seconde renferme tous les fils et les modules électroniques en un produit assemblé.

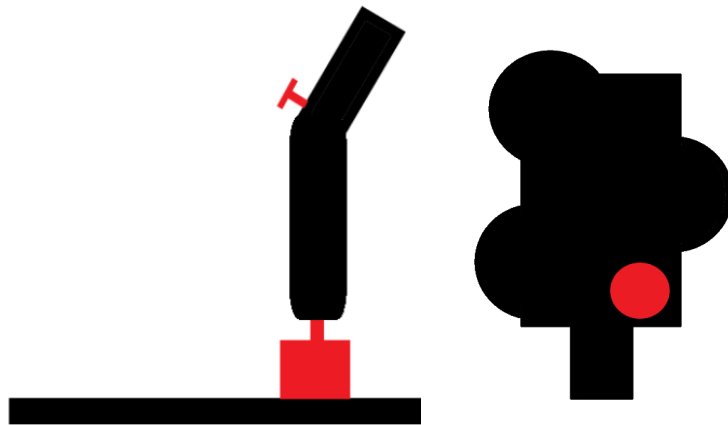
### Mécanique électronique

La majorité des composantes avec lesquelles l'utilisateur interagit ne sont pas conçues par notre équipe mais sont procurées d'une tierce partie et sont intégrées aux circuits de notre produit.

La manette finale comporte deux joysticks : l'un pour le contrôle de l'angle de la caméra et l'autre pour le mouvement du personnage, dans la majorité des jeux de la PlayStation 4. Ce premier se trouve sous le pouce de l'utilisateur, tandis que l'autre est manipulé par le poignet, et toute la manette est montée sur ce joystick. Les diagrammes suivants illustrent l'emplacement exact de ces joysticks en rouge :

À gauche : vue de côté de la manette

À droite : vue de haut de la manette, où le joystick inférieur figurant sur l'illustration de gauche se trouve sous la manette.



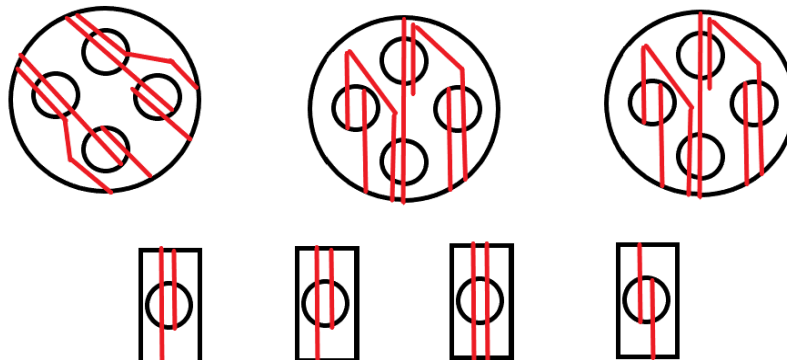
Ces deux joysticks sont accompagnés par 17 boutons qui, lorsqu'ils sont appuyés, complètent un circuit en rejoignant deux fils conducteurs par une plaque conductive imbriquée dans un coussin en silicone. Ce coussin est flexible, ayant une certaine résistance à la force d'appui du doigt sur le bouton, complétant l'expérience tactile satisfaisante d'appuyer sur un bouton.

Il est à noter qu'une manette typique de PS4 contient 18 boutons et deux joysticks, dont deux des boutons sont imbriqués dans les joysticks, et lorsqu'on enfonce ces joysticks, les boutons sont activés. Ceci n'est cependant pas possible dans notre cas pour l'analogique inférieur manipulé par le poignet, car seulement le poids du bras de l'utilisateur pourrait activer ce bouton. Ainsi, un bouton supplémentaire est ajouté à la manette afin de remplir ce rôle.

Il n'y a que 13 des boutons d'une manette de PS4 régulière qui demeurent compatibles mécaniquement avec ce produit. Les cinq autres doivent être fabriqués et incorporés dans la coquille de la manette, qui est l'objet de la prochaine partie.

Afin que les boutons permettent de compléter un circuit, de minces plaques en plastique ont été imprimées en 3-D avec des fosses d'un millimètre de profondeur ou seraient imbriqués les fils conducteurs. Le schéma suivant illustre leur emplacement :

Figure 1 - Diagramme des plaques de boutons et des fosses imprimées en 3-D

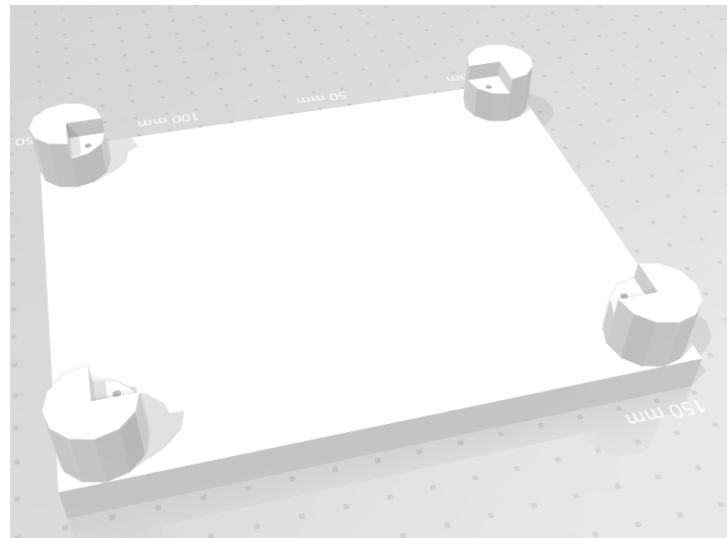


Les lignes rouges montrent l'emplacement des fosses où seraient placés les fils, et les cercles noirs sont où se trouvent les boutons sur lesquels l'utilisateur appuie afin d'envoyer le signal électrique. Les deux boutons absents sur ce schéma, qui n'en contient que 16, n'opère pas avec le même système. L'un se trouve dans le joystick supérieur, tandis que l'autre utilise un capteur de pression pour une plus grande gamme de fonctionnalités.

## Coquille extérieure

La coquille extérieure a été construite entièrement à partir d'impressions en 3-D. La coquille de la manette comporte cinq parties majeures : la base de la manette, la partie de gauche et de droite de la prise et la partie supérieure et inférieure de la tête de la manette. La prise et la manette ont été imprimées séparément afin d'assurer l'intégrité structurelle de l'impression, ainsi que pour faciliter l'assemblage de l'électronique à l'intérieur de la manette. Ces moitiés peuvent être liées ensemble par des supports s'insérant dans un trou dans la paroi de l'autre moitié. Les images suivantes ont été prises des modèles en 3-D des pièces majeures imprimées.

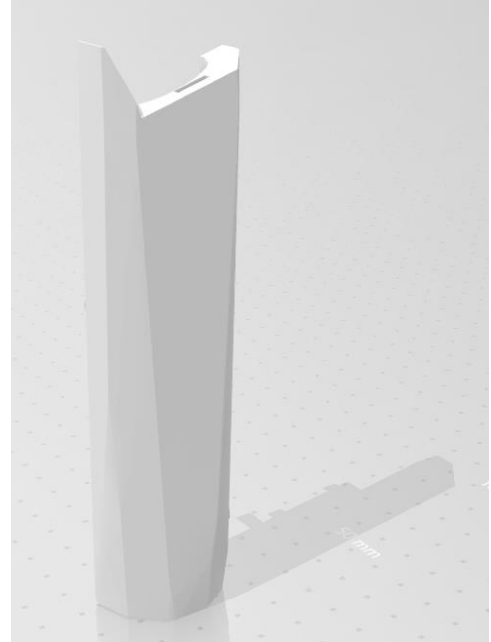
*À droite: la base de la manette ou repose le circuit électrique, le module arduino et le joystick inférieur.*



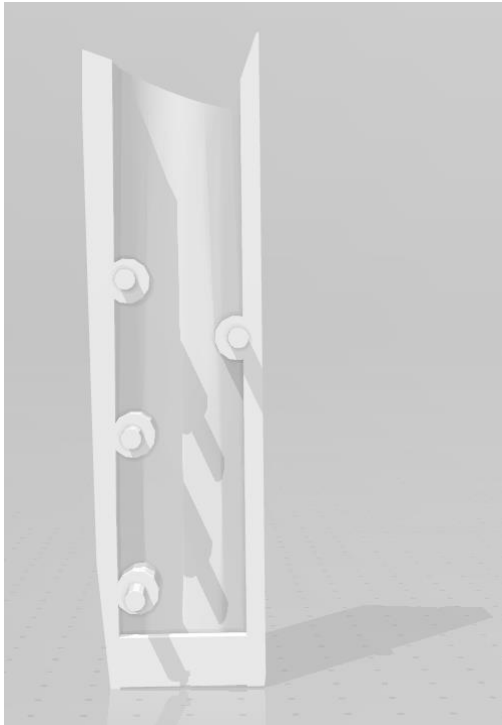
*À gauche : une vue d'intérieur de la portion de droite de la prise de la manette, contenant quatre trous pour lier ensemble les deux moitiés de la prise.*



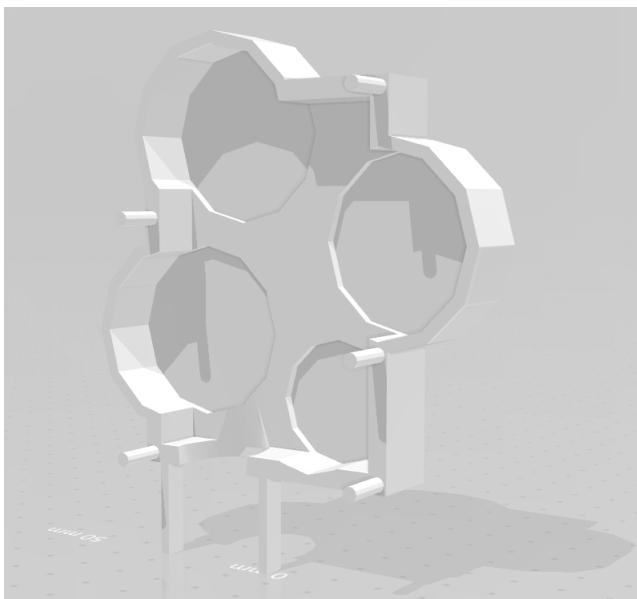
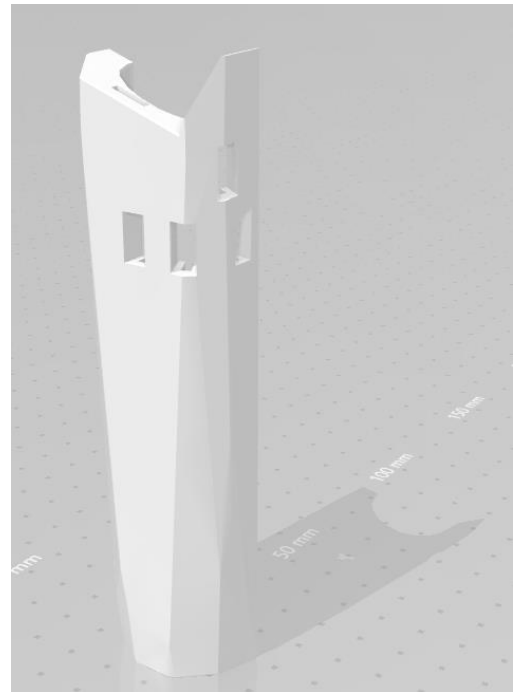
À droite : une vue d'extérieur de la portion de droite de la prise, avec quatre des cinq trous pour des boutons visibles (l'autre se trouvant de l'autre côté de la pièce).



En bas : une vue d'intérieur la portion de gauche de la prise avec les tiges s'insérant dans la portion de droite, ainsi qu'avec le support dans la partie inférieure ayant une petite cavité permettant d'y insérer la partie supérieure du joystick de base.

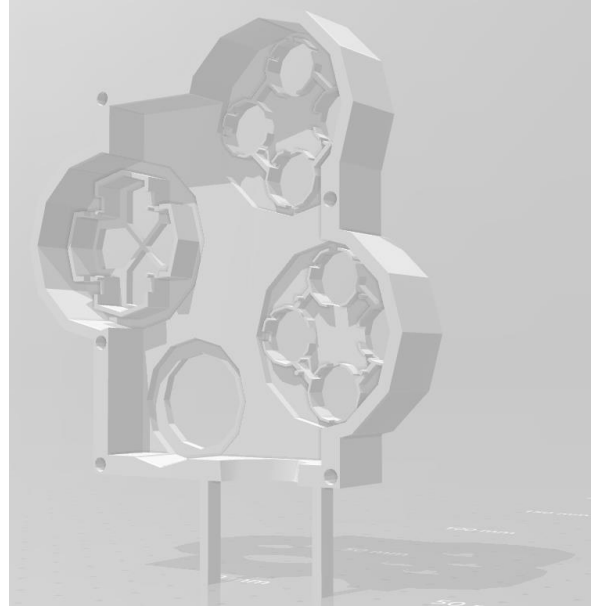


En bas : une vue d'extérieur de la portion de gauche de la manette, avec le trou où se lie la prise et la tête de la manette en évidence.



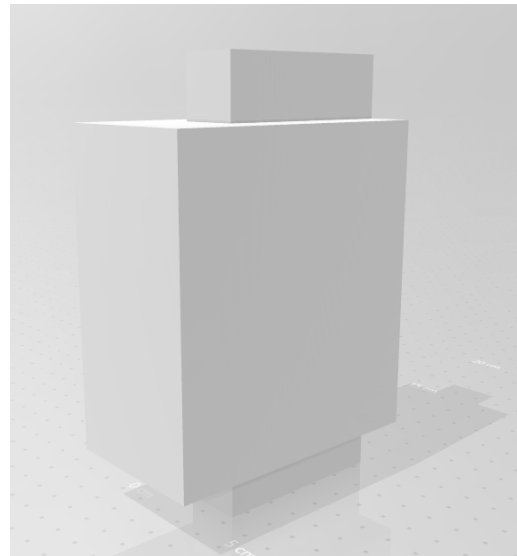
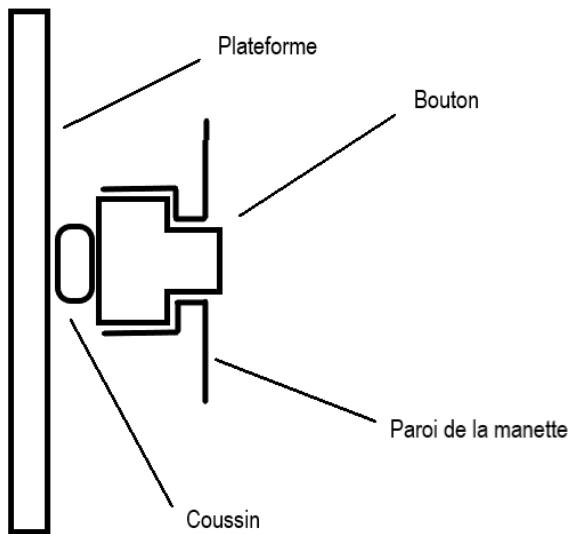
À gauche : la portion inférieure de la tête de la manette, avec les tiges permettant de relier les deux portions de la manette avec la prise

À droite : la portion inférieure de la tête, avec des structures permettant d'y imbriquer les boutons d'une manette de PS4 et les coussins en silicone.



Il y a également plusieurs boutons sur la prise de la manette qui remplacent les gâchettes et autres boutons sur l'arrière d'une manette traditionnelle. Ceux-ci ont été imprimés en 3-D et ont la forme générale suivante :

Figure 2 - Diagramme du mécanisme des boutons sur le manche de la manette



## 2 - Électronique et logiciel

L'UnijeV fonctionne grâce à un microcontrôleur arduino micro. Le arduino micro est une des deux seuls microcontrôleur arduino qui est capable de communiquer par USB. Ceci le permet d'être

reconnaître comme une souris, clavier ou autre interface utilisée pour contrôler un ordinateur. L'arduino a besoin d'envoyer les signaux appropriés à l'ordinateur pour indiquer si chaque bouton est pressé ou relâché, la position des deux joysticks analogues ainsi que comment fort les gâchettes sont pressées. Une librairie appelée ArduinoJoystickLibrary a été utilisée pour simplifier le processus. Cette librairie contient toutes les fonctions besoin pour utiliser un arduino comme une manette de jeux avec deux joysticks analogues, plusieurs gâchettes et jusqu'à 32 bouton digital.

Cette librairie permet à l'arduino d'agir comme une manette pour l'ordinateur mais ne fonctionne pas directement dans avec une PS4. Par contre, théoriquement, cette manette devrait être compatible avec le APEX fournit par le client qui devrait le rendre compatible avec le PS4. Ceci n'a par contre pas été testé puisque la manette n'a pas été complété.

Pour avoir accès au fonctions besoin pour envoyer les signaux à l'ordinateur, il faut inclure et initier la librairie. Une fois que la librairie est installée dans votre IDE arduino, on peut utiliser le code suivant pour faire ceci :

```
#include <Joystick.h>
Joystick_ Joystick;
void setup() {
  Joystick.begin();
}
```

Pour indiquer à l'ordinateur quelle bouton est pesé, on peut utiliser la fonction Joystick.setButton(uint8\_t bouton, uint8\_t value). Donc par exemple si on voulait que le bouton 1 soit actif, on utiliserait la fonction Joystick.setButton(0, 1), ou le 0 est le numéro du bouton (le premier bouton à l'index 0, le deuxième à l'index 1 ect.) et le 1 est le statut du bouton, voulant dire qu'il est activé. Si on voulait indiquer que le bouton ne soit pas actif on mettrait un 0 au lieu du 1.

Pour indiquer à quelle position sont les deux joysticks on peut utiliser les méthodes Joystick.setXAxis(int16\_t value), Joystick.setYAxis(int16\_t value), Joystick.setRxAxis(int16\_t value) et Joystick.setRyAxis(int16\_t value). En paramètre, on passe un numéro de 0 jusqu'à 1023 pour indiquer l'angle du joystick. Passer la valeur 512 indique donc que le joystick est neutre dans l'axe indiquer.

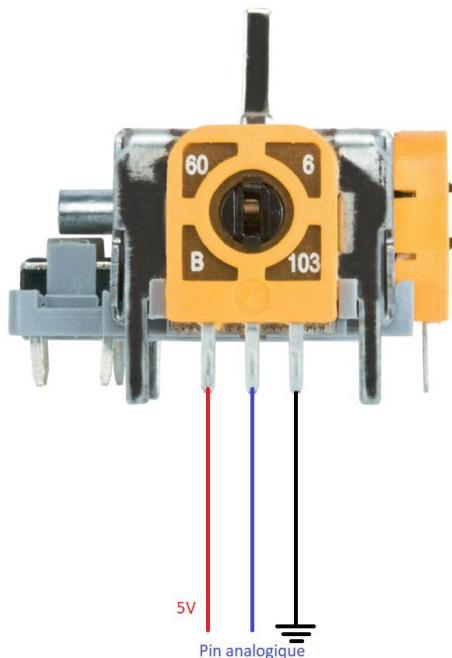
Finalement le dernier type d'information qui faut passer à l'ordinateur pour ce projet est comment fort la gâchette est peser. Pour faire ceci, la méthode Joystick.setThrottle(int16\_t value) a été utiliser. En paramètre on passe un numéro de 0 à 1023 qui indique comment fort la gâchette est peser. 0 étant le plus fort possible et 1032 étant le moin fort possible.

La documentation pour toutes les méthodes utilisées peut être trouver dans le README.md de la page de la librairie sur GitHub. Voici le lien vers cette page:

<https://github.com/MHeironimus/ArduinoJoystickLibrary>. La librairie peut aussi être télécharger de ce lien.

Les joysticks analogues sont capables de communiquer leur position grâce à deux potentiomètres placés sur les côtés du joystick. Un potentiomètre contient une bande de résistance placée dans un arc. Un conducteur fait ensuite contact avec cette bande à différentes positions sur la bande dépendamment de l'angle du joystick. Ceci veut dire que le potentiomètre change de résistance dépendamment de l'angle que le joystick est placé. Un changement de résistance cause un changement de voltage qui peut être mesurer par un pin analogue de l'arduino. Pour mesurer le voltage, le circuit suivant peut être assembler :

Figure 3 - Joystick analogue et sorties électriques



Certains joysticks par exemples doivent être câblée différemment, tels que celle utilisée à la base de la manette pour notre dernier prototype, mais le principe reste le même.

La valeur mesurée peut ensuite être passer en paramètre dans une des fonctions dédiées à communiquer la position des joystick, tels que par exemple la fonction `Joystick.setRyAxis(int16_t value)`, pour que la vraie position du joystick physique soit passée à l'ordinateur. Voici un exemple du code utiliser pour contrôler un axe d'un joystick. Le code serait placé dans la fonction `loop()`.

```
Joystick.setXAxis(analogRead(A6));
```

La gâchette elle fonctionne grâce à une résistance à capteur de force, une résistance qui change sa valeur dépendamment de comment fort elle est pesée. En connectant donc le port 5V de l'arduino d'un côté et un pin analogue de l'autre, il est possible de mesurer comment fort la résistance est pesée.

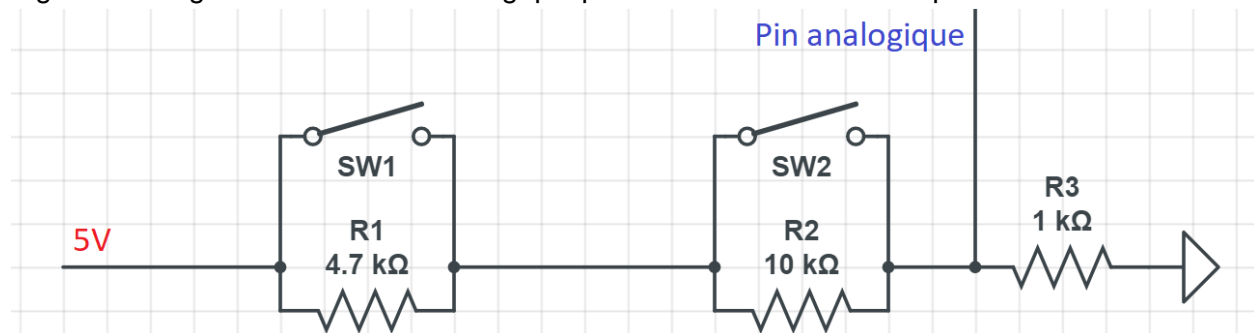
Figure 4 - Senseur de pression analogique et sorties électriques



La valeur mesurée peut ensuite être passer en paramètre dans la fonction `Joystick.setThrottle(int16_t value)` pour indiquer comment fort le trigger est peser.

L'unijeV possède trop de bouton pour que chaque bouton ait leur propre pin digital. On a donc dû utiliser un circuit pour pouvoir placer 2 boutons sur un pin analogique. Ce circuit a été répété 5 fois dans la manette pour avoir dix boutons normaux qui utilise en total 5 pins analogiques au lieu de 10 digitales. Voici le diagramme de ce circuit :

Figure 5 - Diagramme de circuit analogique pour deux boutons numériques



Chaque combinaison de bouton peser et pas peser cause un voltage unique à l'endroit où est placé le fil qui se rend au pin analogique. Ce voltage peut être lue et ensuite on peut déterminer avec ce voltage quelle bouton, si aucun, est peser.

Pour déterminer ceci on a besoin de connaître quelle valeur est lue par la pin analogique pour chaque combinaison de bouton peser/pas peser. Pour trouver ces valeurs, on lit la valeur de la pin analogique (valeur de 0 à 1023) à l'aide de `analogRead()` et l'affiche à l'aide de `Serial.print()`. Avec ces valeurs, il est possible d'écrire des intervalles pour chaque combinaison pour minimiser les chances que le mauvais signal soit envoyé à l'ordinateur. Voici un exemple de tel code :

```

int reading = analogRead(A2);
if (reading > 400 && reading < 415){
  Joystick.setButton(9, 0);
  Joystick.setButton(10, 0);
} else if(reading > 505 && reading < 515){
  Joystick.setButton(9, 1);
  Joystick.setButton(10, 0);
} else if(reading > 525 && reading < 695){
  Joystick.setButton(9, 0);
  Joystick.setButton(10, 1);
} else if(reading > 695){
  Joystick.setButton(9, 1);
  Joystick.setButton(10, 1);
}

```

Les intervalles peuvent changer de circuit a circuit, donc il est important de les tester.

Il y a quand même assez de pins dans l'arduino micro pour que certains boutons ne sont pas dans tels circuits. Pour ces boutons là, tout ce qu'il y a faire est de connecter un bout du bouton à la terre et l'autre côté a un pin digital. Ensuite, dans la fonction setup() du code arduino, il faut set le pin utiliser au mode input pullup en faisant pinMode(int numéroDePin, INPUT\_PULLUP). On peut ensuite déterminer si le bouton est peser ou non en faisant digitalRead(int pinNum) et passant le numéro de la pin en argument. Il est important de noter que dans le mode pin pullup, la valeur lue par le pin sera 0 quand elle est pesée et 1 quand elle ne l'ait pas. Il est donc important d'inverser la valeur lue avec l'opérateur "!" quand on passe la valeur dans Joystick.setButton(uint8\_t button, uint8\_t value). Par exemple :

```

#include <Joystick.h>
Joystick_ Joystick;
void setup() {
  Joystick.begin();
  //joystick btns
  pinMode(2, INPUT_PULLUP);
}
void loop() {

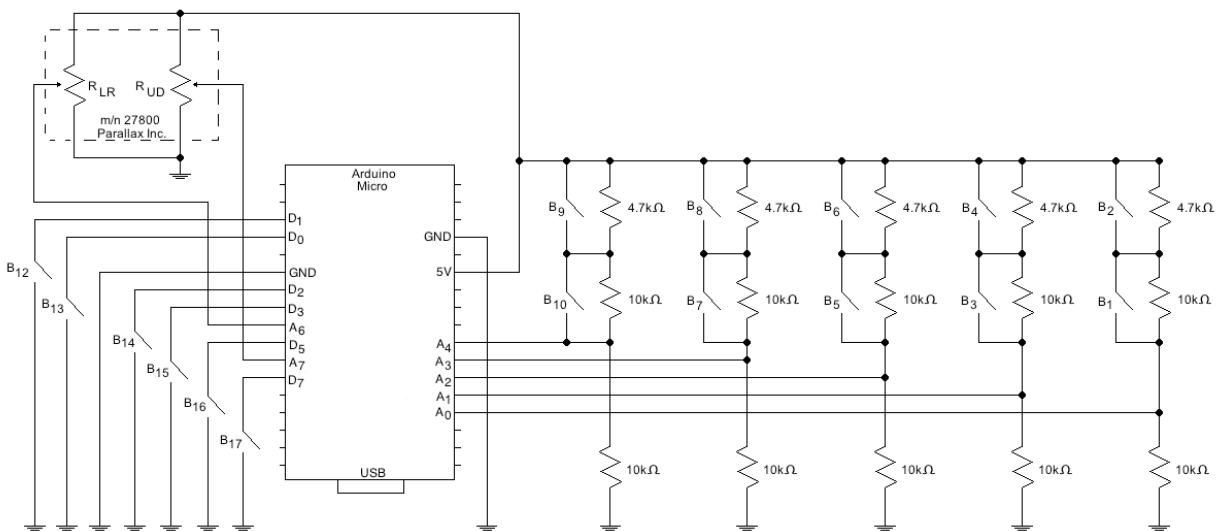
  Joystick.setButton(1, !digitalRead(2));

```

### 3 - Électrique

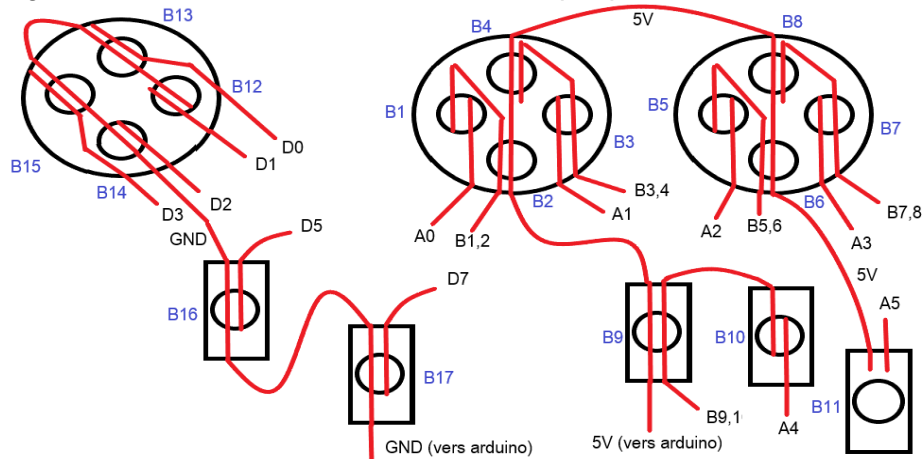
La portion électrique principale est contenue sur un grand circuit imprimé dans la plaque démontrée dans la section mécanique. Le fonctionnement de toutes les parties individuelles a été expliqué dans la section précédente, donc cette section portera principalement sur l'intégration de tous ces modules dans la manette. Comme cité précédemment, la manette contient deux joysticks, l'un contenant aussi un bouton digital, et 17 autres boutons, dont 10 d'entre eux se trouvent sur un circuit à entrée analogique en paires de deux, comme illustré dans la section précédente. Ainsi, la figure suivante illustre l'emplacement de chaque circuit et module sur la plaque principale :

Figure 6 - Schéma technique de la plaque contenant tous les circuits et modules



Ces boutons sont reliés ensemble et au circuit sur les plaques imprimées ainsi :

Figure 7 - Diagramme des connexions entre circuits et plaques de boutons



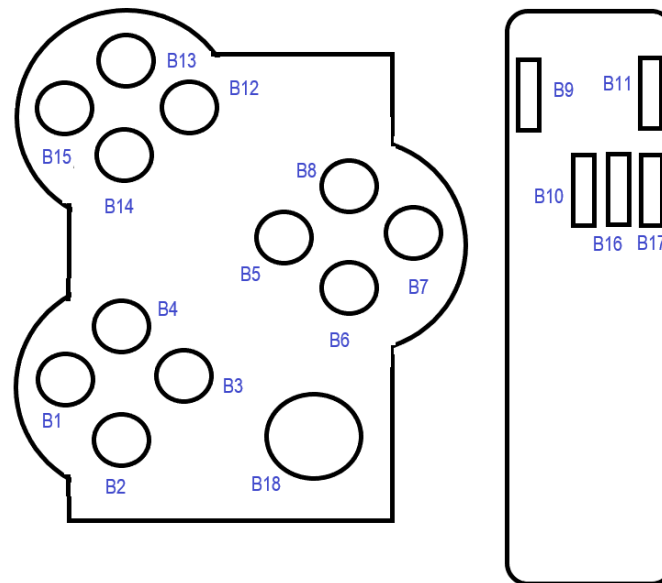
En rouge sont les connexions par fil conducteur, en texte noir les connexions entrant dans le circuit intégré et l'arduino, et en texte bleu la nomenclature de chaque bouton. Il est à noter que

le bouton B11 contient le capteur à pression, et que le bouton sur le joystick ne figure pas dans ce schéma.

Il se pourrait que l'utilisation des entrées D0 et D1 dans l'arduino cause des problèmes de communication avec l'ordinateur. Ainsi, ces connexions peuvent être acheminées aux entrées D11 et D12, assurant le fonctionnement de tous les boutons.

Tous les boutons mentionnés se trouvent aux endroits suivants sur la manette physique :

Figure 8 - Schéma de la nomenclature de chaque bouton correspondant sur la manette



Tous les fils se dirigeant vers l'arduino et la plaque de circuits passent à l'intérieur du manche de la manette, avec assez d'espace pour assurer le mouvement libre du joystick inférieur et l'appui des boutons B9, B10, B11, B16 et B17.



## Comment utiliser et maintenir la manette

Notre produit se démarque des autres sur le marché par son utilisation avec une seule main. Le prototype qui a été fabriqué par notre équipe est spécialisé pour la main gauche par la disposition de ses boutons, mais une image miroir du produit permettrait de l'utiliser plus facilement avec la main droite. Il ne s'agit que de tenir le manche de la manette avec la main gauche assez vers le haut pour que le pouce atteigne confortablement le joystick supérieur et les huit boutons autour de celui-ci. Comme expliqué dans la section électronique, la manette interface soit avec un ordinateur ou avec APEX si l'utilisateur en possède un. En manipulant la manette ainsi, il est possible de programmer la fonction de chaque bouton, ce qui demeure à la discrétion de l'utilisateur selon ses préférences.

Le maintien pour ce produit est assez simple. Il ne s'agit que de nettoyer régulièrement la coquille extérieure, comme on le fait pour toute autre électronique domestique. Il pourrait s'avérer nécessaire de remplacer certains fils, qui peuvent s'user avec le temps. Afin d'accomplir ceci, il faut suivre le guide dans la partie électrique afin de bien maintenir les bonnes connexions.

## Conclusion et recommandations

Après multiple itération et plusieurs heures de travail nous sommes arrivés avec un produit presque complet. Le projet est conçu comme une manette de pilot pour contrôler les avions soit un long bras connecté à la base avec un joystick et connecté en haut avec la tête comprenant tous les boutons. Inclure un joystick dans la base et l'autre sur la tête permet aux utilisateurs de pouvoir contrôler à la fois la caméra et le mouvement du personnage dans le jeu. Le reste des boutons sont dispersés sur la tête et le manche de la manette en assurant un certain confort et accessibilité.

Plusieurs leçons ont été apprises durant ce long projet. Bien diviser les tâches permet de diminuer la charge qu'une seule personne doit accomplir en plus d'augmenter les chances de réussite du projet dans un délai raisonnable. Lorsque plusieurs tâches sont faites en même temps il est possible de voir un produit final plus rapidement et d'anticiper des problèmes qui pourraient survenir. Nous avons négligé le côté physique de la manette à cause d'une mauvaise estimation de la complexité de cette partie du projet et cela nous a coûté un produit final complètement fonctionnelle. Une grande expertise sur le côté électronique de la manette a été acquise grâce à bien des recherches et pratiques.

Pour le futur, il est important de garder en tête tous les aspects qui sont inclus dans la conception d'un produit pour assurer aucune négligence pour arriver à une solution fonctionnelle dans le temps requis.

La première s'agit de s'assurer que la distribution des connaissances et des tâches est bien homogène. Ainsi, plus de membres peuvent mieux accomplir une plus grande diversité de tâches plus rapidement, et les recherches et étalonnages peuvent être faits par tous les membres du groupe.

Deuxièmement, il est nécessaire de bien dimensionner les modèles en 3-D. Les parties électroniques devraient être commandées aussitôt que possible et leurs dimensions devraient être bien mesurées afin qu'aucun problème mécanique ne se produise. Il faut également être au courant des limitations techniques des outils à la disposition de l'équipe de conception, puisque des erreurs d'impression en 3-D étaient le problème le plus courant au cours du projet.

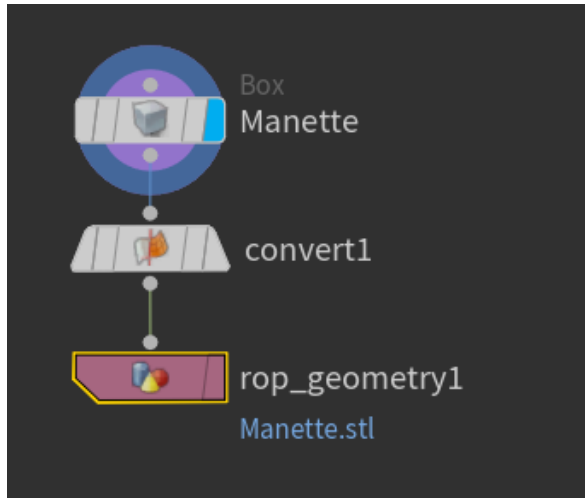
Finalement, il est nécessaire de concevoir tout d'abord des systèmes électroniques et logiciels, et s'assurer de leur bon fonctionnement. Par la suite, la conception des composantes mécaniques peut être entamée, car elle dépend énormément des modules électroniques et des circuits électriques utilisés. La partie logicielle est plutôt indépendante et peut être accomplie peu à peu au cours du projet, mais la partie clé du projet est la mécanique, qui dans notre cas a malheureusement été négligée.

# Appendices

## Appendices A - Logiciels employés

### **Modélisations en 3-D :** Houdini Apprentice 17.5.391

\*afin de convertir les fichiers .hipnc de ce logiciel en fichier .stl lisible par le logiciel d'impression, les nodes suivantes ont été ajoutées dans le réseau de flux de travail de Houdini :



### **Impression en 3-D :** UltiMaker Cura 4.4

### **Programmation :** Arduino 1.8.10