

VINCENT CÔTÉ

Perspective de carrière

420-156-AT, groupe 101

COMPTE RENDU DE VEILLE TECHNOLOGIQUE

Travail présenté à

Jean-Michel Laliberté

Cégep de l'Abitibi-Témiscamingue

6 décembre 2021

TABLE DES MATIÈRES

1. Introduction.....	3
2. Développement.....	3
2.1 Avant le projet.....	3
2.1.1 Intention	3
2.1.2. But du projet	4
2.2 Planification.....	4
2.2.1 Ordinateur	4
2.2.2 Contrôles.....	5
2.2.3 Cabinet	6
2.3 Configuration du Raspberry Pi	6
2.3.1 Première tentative.....	6
2.3.2 Deuxième tentative.....	7
2.3.3 Configurer la sortie audio	8
2.4 Configuration les contrôles	8
2.4.1 Préparer le <i>Pinscape</i> et les boutons.	9
2.4.2 Construire les fils	9
2.5 Construire le cabinet	10
2.6 Opportunités d'emploi.....	11
Conclusion.....	11
Annexe.....	13

1. INTRODUCTION

Quand les arcades ont connu leur point le plus haut de popularité dans les années 80, on pouvait en voir, éparpillées dans les villes et il n'est pas difficile de reconnaître l'impact redoutable que les jeux vidéo ont eu sur la société à l'aide de ces arcades. Cependant, ils sont devenus de moins en moins populaires avec l'arrivée des consoles de jeux vidéo de maison ainsi que l'énorme disponibilité des jeux et les librairies des consoles qui s'agrandissent, comparées aux arcades où on ne pouvait jouer qu'à un seul jeu sur la machine. En revanche, avec l'avenue des systèmes d'émulation de consoles et de jeux, on vit une renaissance de ces jeux « rétro » des années 80 et 90. Un des exemples que j'ai remarqué sur Internet est les machines de billard électrique (Pinball) virtuelles. Celles-ci utilisent l'application Visual Pinball, qui est conçue pour rouler des tables de billard créées et programmées par des utilisateurs et amateurs de Pinball. Le plus grand avantage de ces machines est que, comparées à de véritables machines de billard, elles sont capables de rouler autant de tables que l'on désire sur la même machine. Cela revient donc beaucoup moins cher à construire, comparé aux vraies tables qui peuvent coûter au-delà de 5000 \$. C'est pourquoi j'ai décidé, comme projet de veille technologique, de construire une machine Pinball virtuelle. Dans ce compte-rendu de mon projet, je vais expliquer l'intention et le but de celui-ci, sa planification et son développement du début à la fin, ainsi que présenter les opportunités d'emploi liées à ceci. Je vais terminer en évaluant le résultat final de mes efforts à l'aide de critères qui sont rédigés plus bas.

2. DÉVELOPPEMENT

2.1 Avant le projet

2.1.1 Intention

L'intention de mon projet de veille technologique était de construire une machine *Pinball* virtuelle incluant le cabinet de la machine. Ce que je veux dire par « virtuelle », c'est que la table elle-même est virtuelle, il n'y a pas de composantes physiques comme des *flippers*¹ ou des pare-chocs (*bumpers*) qui projettent une bille métallique à travers la table de jeu. Également, je me suis imposé que je ne puisse pas faire une machine à bille grandeur nature, car cela requerrait beaucoup d'argent et beaucoup plus de temps que j'avais en main. En fait, je n'avais qu'environ 13 semaines pour faire mon projet et l'achat des différentes composantes ainsi que leur installation aurait dépassé ma limite.

¹ Ce sont les petits leviers contrôlés mécaniquement par le joueur qui permettent d'envoyer la bille vers le haut.

2.1.2. But du projet

Le but principal de mon projet était d'anticiper si la construction d'un cabinet de jeu vidéo et l'installation de contrôles de jeu réalistes, compte tenu du travail que cela demande, du temps nécessaire et de l'argent que l'on doit disposer pour le projet, donne un résultat qui vaut l'effort appliqué sur le projet. Pour réaliser une réponse au but, à la fin de mon projet, j'utilise six critères d'évaluation : le coût des ressources (contrôles, cabinet, ordinateur), la disponibilité des ressources (où je peux les acheter, temps de livraison), la qualité du jeu, la qualité du cabinet, la qualité des contrôles et le niveau de l'immersion.

2.2 Planification

Au début du projet, avant de commander toute composante pour mon projet, j'ai dû faire une planification. Cela comprend la recherche pour trouver quelles technologies que je pouvais utiliser selon l'argent que j'avais à ma disposition ainsi que le temps qui m'était accordé dans le cours.

2.2.1 Ordinateur

Pour débiter, il me fallait un ordinateur central qui serait utilisé uniquement pour rouler mes logiciels, et ce, dans une installation permanente et fermée. En fait, le jeu de tables Pinball que je veux rouler est Full Tilt! Pinball. C'est un vieux jeu sorti en 1995 et développé pour rouler sur des systèmes Windows 95 et XP. Vu que les technologies nécessaires pour rouler ces genres de systèmes étaient assez faibles, il serait facile de faire rouler le jeu sur un vieil ordi ou un microordinateur. J'avais déjà l'idée d'utiliser un Raspberry Pi comme ordinateur non seulement parce qu'il est très petit, mais également peu coûteux, comparé aux autres. Un Raspberry Pi est tout simplement un ordinateur traditionnel, mais en plus petit format. Cela impose que sa puissance et ses capacités soient bien moindres, cependant celles-ci sont suffisantes pour mon jeu. En choisissant le modèle 3B+, j'obtiens le même nombre de ports USB ainsi qu'un port HDMI standard pour connecter un écran qui pourrait être un peu vieux. En parlant d'écran d'ordinateur, j'ai trouvé un écran inutilisé par le cégep et après quelques tests, il fonctionnait parfaitement et je l'ai donc utilisé pour mon projet. Un autre point important pour choisir un Raspberry Pi est la présence de ports GPIO (Annexe : Image 1) directement sur l'ordinateur. GPIO veut dire General Purpose Input/Output, ils permettent de brancher certaines composantes, des boutons et des lumières LED par exemple, et ensuite d'écrire un script pour que telle action se produise sur l'ordinateur lorsqu'on envoie un courant sur le port, comme le clique d'un bouton. J'avais l'intention de brancher un bouton externe qui serait installé sur le côté du cabinet pour

démarrer l'ordinateur, ainsi que brancher un petit ventilateur avec un des ports GPIO 5V, qui pourrait envoyer du courant au ventilateur pour le faire fonctionner.

2.2.2 Contrôles

Ensuite, pour les contrôles de style Pinball que j'avais grande intention d'installer, il a fallu que je passe quelques heures pour trouver où est-ce que je pouvais en acheter. J'aurais préféré en acheter en magasin, mais je me suis rendu compte qu'il n'y avait pas beaucoup de magasins qui vendaient des pièces de cabinet et aucun n'avait les contrôles que je voulais. Je me suis donc retenu à les commander en ligne. En ayant des connaissances de bases sur les machines Pinball, je savais que le site <https://virtuapin.net/> pouvait offrir des composantes. J'ai trouvé un ensemble de boutons (Annexe : Image 2) qui se vendait pour environ 80\$ CAN. Même avec le prix qui semble élevé, j'ai décidé de les commander et de les faire livrer aux résidences. Il ne me fallait au total que cinq boutons pour jouer à mon jeu correctement. J'ai rapidement réalisé que ce projet pourrait être plus coûteux que j'avais imaginé. Pour poursuivre, non seulement il fallait que j'achète les boutons, il fallait également que je trouve une façon de les connecter au Raspberry Pi. Heureusement, en faisant une recherche pour un guide de construction de cabinet[1], j'en ai croisé un qui présentait une liste des matériaux que l'auteur a utilisés pour toutes les parties du cabinet. Le guide en tant que tel était très détaillé et je m'y suis beaucoup fié pour planifier le reste de mon projet. Il présentait l'implémentation des contrôles en utilisant un émulateur de clavier KL25Z, communément appelé Pinscape. C'est une plaque électronique qui comporte également des ports GPIO comme le Raspberry Pi et permet d'émuler des inputs de clavier à partir de contrôles externes. J'ai configuré les inputs de clavier à partir d'une application qui permet de le configurer, cependant elle ne fonctionne que sur Windows. J'ai pu tout simplement le connecter avec un câble USB pour le configurer. En me référant au guide de construction, j'ai remarqué que je pouvais acheter le matériel nécessaire pour créer les fils électriques en ligne. Avec mes boutons, j'ai également acheté sept rouleaux de quatre pieds de fils électrique de grosseur 22 AWG. C'est important que chaque fil ait une couleur différente parce que ça permet de mieux distinguer les différents boutons et de mieux déboguer s'il y a des problèmes de connexion. Je voulais brancher le fil de mise à terre en série à travers chaque bouton, donc que chaque sortie de mise à terre sera connectée à un même fil, et cela permettrait d'utiliser moins de fil si jamais il arrive des imprévus durant la construction. Pour pouvoir connecter les boutons à mon émulateur, j'ai décidé d'acheter des pièces pour installer des connexions GPIO (Annexe : Image 3), ce qui comprenait des connecteurs femelles à sertir ainsi que des boîtiers en plastique (Annexe : Image 4), au lieu de les souder

sur les connecteurs directement. Celles-ci permettront de déconnecter les fils à la main sans modification permanente ou besoin de dessouder des composantes.

2.2.3 Cabinet

Finalement, pour la construction du cabinet, j'avais décidé d'utiliser des morceaux de bois qui traînaient dans le garage de mon père. J'avais peur qu'acheter des planches de bois neuf coûte trop cher. Par chance, en regardant un peu chez mon père, on a trouvé de longues planches de bois bleues de 2X4 qu'on a utilisées pour faire les côtés du boîtier ainsi qu'une planche de bois mince assez grande pour faire le dessus et le dessous du boîtier. Je voulais déjà peindre le cabinet en noir, donc le fait que le bois était coloré n'était pas un problème.

2.3 Configuration du Raspberry Pi

Quand j'ai reçu mon Raspberry Pi, il fallait que j'installe le système d'exploitation Raspberry Pi OS (formellement Raspbian) sur la carte SD qui lui servait de disque dur principal. Ce système d'exploitation (OS) est conçu spécialement pour rouler sur des Raspberry Pi et est basé sur la branche Debian de Linux. Il est très semblable à ce dernier, mais l'interface est simplifiée afin de moins consommer de ressources. En fouillant à l'Infocentre, j'ai trouvé un lecteur de cartes SD que j'ai connecter à mon ordinateur et j'ai ensuite installé le système d'exploitation avec l'application Raspberry Pi Imager. Celle-ci simplifie beaucoup l'installation et j'ai pu rapidement télécharger et installer l'OS sur la carte. Ensuite, il me fallait un adaptateur pour connecter l'écran d'ordinateur que j'ai dépêché à l'ordinateur, car celui-ci n'a qu'un port HDMI et l'ordinateur n'en a pas. Heureusement, j'avais sur moi un adaptateur de câble VGA vers HDMI (Annexe : Image 5) et, à l'aide d'un câble VGA que je pouvais connecter à l'écran, j'ai pu connecter l'ordinateur. J'ai ensuite démarré l'ordinateur et fait les configurations générales du système, comme changer le langage du clavier et configurer le fuseau horaire. Cependant, en travaillant dans le local du cégep, il n'y avait pas assez de câbles Ethernet pour avoir une connexion, dont j'avais besoin pour installer mes logiciels. J'ai donc utilisé un petit commutateur qui permettait de diviser les connexions internet à l'ordinateur du local ainsi que mon Raspberry Pi.

2.3.1 Première tentative

Après cela, j'ai fait la majorité de mes recherches sur l'ordinateur du local parce que l'écran est plus grand que celui de mon projet. Parce que je savais que mon jeu ne fonctionnait que sur des

systèmes en architecture x86, comme Windows, j'ai commencé mes recherches là-dessus. En premier, j'ai trouvé que le logiciel *WineHQ* permettait de faire rouler des applications Windows sur des systèmes Linux. C'est en fait une couche de compatibilité qui traduit les appels de l'application Windows en appels POSIX, qui est le mode d'appels utilisé par les systèmes comme Linux et macOS qui sont conformes aux normes POSIX. Le site officiel offre une page pour chaque version du logiciel ainsi qu'un guide des commandes pour l'installer. Parce qu'il y a différentes versions de Debian, j'ai dû faire un peu de recherches sur le site officiel de Raspberry Pi pour trouver la version sur laquelle elle est basée et j'ai trouvé que c'était la version Debian 10 (buster). J'ai donc commencé l'installation.

Au début de l'installation, j'ai tout de suite rencontré des problèmes avec des fichiers de configuration sur l'ordinateur qui causaient des problèmes pendant que les packages du logiciel tentaient de s'installer. Un package, c'est en fait un bloc d'instruction qui dicte à l'ordinateur les fichiers à télécharger et où les installer, comment changer des fichiers de configuration, etc. J'ai dû faire plusieurs recherches sur Internet pour régler les problèmes. Cependant, pour la majorité de ceux-ci, j'avais de la misère à trouver des solutions et j'avais souvent besoin de regarder plusieurs vidéos pour comprendre quel était le problème. À travers les solutions que j'avais trouvées, il fallait soit que j'installe d'autres packages qui supposément allaient régler mes problèmes. Finalement, je n'ai pas pu tous les régler et j'étais coincé sur un problème dont la seule solution était que je change de système d'exploitation. C'est à ce moment que j'ai réalisé que le site de *WineHQ* dit clairement que le logiciel ne fonctionne que pour les architectures x86 et non ARM et le Raspberry Pi est basé sur une architecture ARM. Il me fallait donc trouver une autre alternative.

2.3.2 Deuxième tentative

À partir de cette première tentative, je savais qu'il fallait que je simule un système d'architecture x86 sur mon ordinateur. J'ai commencé par rechercher comment faire rouler *WineHQ* sur un Raspberry Pi et j'ai parcouru plusieurs publications sur des forums différents. J'ai réussi à assembler les informations acquises de mes recherches pour déterminer que je devais utiliser le programme box86 pour simuler l'architecture. Ce programme est un émulateur de système x86 utilisant les bibliothèques de ce système qui permet de rouler des programmes Linux x86, comme *WineHQ*, sur des systèmes non-x86, par exemple ARM dans le cas de mon Raspberry Pi. Pour installer le programme, j'ai regardé courte vidéo de guide d'installation et j'ai remarqué que l'application Pi Apps avait les logiciels que j'avais besoin d'installer dans sa librairie. Pi Apps est simplement un magasin d'applications du même style que l'App Store sur macOS qui roule sur des systèmes ARM. J'ai été capable d'installer mes logiciels,

box86 et *WineHQ*, à partir de cette application et de les configurer. Pour tester le jeu, j'ai pu simplement téléverser le dossier complet du jeu sur le Raspberry Pi et le rouler à partir de *WineHQ*. J'ai heureusement constaté que le jeu fonctionnait parfaitement hormis quelques moments où il y avait du screen tearing, c'est lorsque plusieurs images du jeu sont affichées dans le même instant d'affichage et qui se superposent une sur l'autre, causant un mélange de plusieurs images sur l'écran.

Finalement, pour terminer la configuration, je voulais avoir une façon que le jeu roule automatiquement quand je démarre l'ordinateur. J'avais installé un bouton externe pour démarrer la machine et je ne voulais absolument pas avoir besoin de connecter un clavier ou une souris pour démarrer le jeu, c'était donc impératif. J'ai encore fait des recherches sur comment faire rouler un script au démarrage d'un ordinateur Linux et j'ai trouvé qu'en modifiant des fichiers de configuration de démarrage, l'ordinateur pouvait exécuter des lignes de codes ou même un script externe. Après plusieurs essais, j'ai finalement créé deux fichiers de code. En premier, quand l'ordinateur démarre, il exécute un fichier `.desktop` qui est sauvegardé dans le dossier de scripts d'initialisation. Les fichiers avec le suffixe `.desktop` sont des raccourcis permettent de rouler diverses commandes lorsque l'ordinateur est complètement démarré et sont souvent utilisés pour démarrer des applications. Dans mon cas, j'en ai créé un nouveau qui démarre un deuxième fichier de code dont celui-ci fait toutes les configurations nécessaires pour rouler le jeu proprement. Il roule d'abord des commandes avec la librairie *pigpiod* pour rediriger le son vers le port GPIO 12 sur lequel est connecté le haut-parleur. Ensuite, pour s'assurer que tous les processus sont démarrés correctement, il fait une pause de 5 secondes avant de faire rouler le fichier exécutable du jeu avec *WineHQ*.

2.3.3 Configurer la sortie audio

Au début, je n'avais pas l'intention de mettre un haut-parleur à cause la contrainte de temps qui était imposée, mais vers la fin de mon projet, j'ai décidé de chercher pour en installer un dans le cabinet. J'avais trouvé un petit haut-parleur qui avait une connexion sur les ports GPIO. J'ai ensuite configuré le Raspberry Pi pour diriger la sortie audio sur ces ports au lieu de la sortie audio standard et j'ai pu connecter le haut-parleur sur les ports.

2.4 Configuration les contrôles

Pour configurer et installer les boutons qui me serviraient de contrôles pour mon jeu, j'ai grandement suivi le guide de construction dont j'ai mentionné dans la section *Contrôles* à la page 5.

2.4.1 Préparer le *Pinscape* et les boutons.

En premier, il fallait que je prépare la plaque d'émulation Pinscape pour permettre de brancher et d'utiliser les boutons à partir de celui-ci. D'abord, quand j'ai reçu la plaque, il n'y avait pas de ports GPIO présents dessus. J'ai alors commandé un bloc de 16 ports GPIO (Annexe : Image 6) pour ensuite le souder sur la plaque. Cependant, je n'avais pas beaucoup d'expérience en soudure et j'avais donc peur de ruiner la plaque ou de faire un mauvais travail. Heureusement, j'ai pu demander à Gabriel, le technicien du cégep, de m'aider pour souder les connecteurs parce qu'il avait de l'expérience avec ces genres de soudure. Le résultat final était excellent et je suis très heureux d'avoir eu son aide. En fait, il a utilisé de la pâte à souder, cela a aidé à l'étain de se déposer sur les connecteurs de la plaque et de former une belle soudure beaucoup plus facilement.

Ensuite, j'ai configuré l'émulateur avec l'application disponible pour celle-ci, *PinscapeConfigTool*. Cette application permet de configurer les ports GPIO sur la plaque (Annexe : Image 11) pour qu'ils simulent une activation d'une touche de clavier à l'ordinateur. Cependant, vu qu'elle ne roule que sur Windows, il fallait que je le configure en avance et j'ai donc fait un plan de construction de mes composantes (Annexe : Image 7). Dans mon cas, selon le plan que j'ai fait basé sur les essais de mon jeu, j'aurais besoin de cinq boutons : deux sur les côtés pour les flippers, un pour projeter la balle avec le piston, un pour démarrer une partie et un autre pour mettre la partie en pause. Pour les flippers, j'ai utilisé des boutons avec interrupteur à lame (Annexe : Image 9) qui font passer le courant quand le bouton est pressé, donc forçant les deux lames à se toucher. Pour les trois autres boutons, j'ai utilisé des boutons d'arcade traditionnels (Annexe : Image 10) qui comprennent un système d'activation comparable à des interrupteurs de maison, où la fermeture de la connexion se fait en un coup sec lorsque le bouton est enfoncé.

2.4.2 Construire les fils

Il fallait ensuite que je fabrique les fils afin de connecter les boutons à la plaque d'émulation. Avant de commencer, j'avais besoin de pinces de sertissage (Annexe : Image 8) qui pouvaient fermer les connecteurs en métal (Annexe : Image 4) sur les fils électriques que j'avais commandés. J'ai finalement acheté une paire parce que l'Infocentre n'en possédait pas dans ses divers outils d'électronique. Pour m'assurer que je n'allais pas manquer de longueur de fil, j'ai coupé les rouleaux de fil en deux pour ainsi avoir 2 pieds de fils pour chaque bouton. Je n'avais pas le choix d'être autant prudent parce que je n'avais pas encore construit le cabinet ou même fait le plan de construction. Pour connecter les fils,

vu que les boutons possèdent des sorties de connexion en languette, j'ai utilisé des bornes de sectionnement (Annexe : Image 12) qui permettront de déconnecter les fils au cas où il y aurait des problèmes quelconques. J'ai testé chaque fil un par un immédiatement après que je l'aie construit et je n'ai pas rencontré de mauvais fil ou tout autre problème. Pour le fil de mise à terre, j'ai attendu que le cabinet soit assemblé pour déterminer où couper le fil. J'ai fait une connexion en série à travers tous les boutons en coupant le fil à l'endroit planifié, puis en dénudant les deux bouts de fil et, par la suite, entortiller les bouts afin d'assembler la borne sur les deux bouts en même temps. Pour le bouton d'alimentation, j'ai fait deux fils différents, donc également un pour la mise à terre, parce qu'il était connecté directement sur le Raspberry Pi et à cause de la façon que l'ordinateur peut être démarré, qui est de faire un raccourci entre les ports GPIO 5 et 6 (Annexe : Image 1).

Finalement, j'ai testé tous les boutons en les connectant à la plaque, puis en connectant la plaque à mon Raspberry Pi et tout marchait parfaitement.

2.5 Construire le cabinet

Pour construire le cabinet, moi et mon père avons découpé les morceaux nécessaires dans des retailles de bois qui traînaient chez lui. Pour les côtés, on a pris des planches de $\frac{3}{4}$ de pouce d'épaisseur et d'au moins 6 pouces de largeur. Cela était amplement grand pour placer même le plus grand bouton qui fait au moins 3 pouces de diamètre. Pour le dessus et le fond, on a pris une grande planche de tableau. Ensuite, j'ai dessiné un plan de construction rapide afin de déterminer où les boutons et les composantes seraient placés dans le cabinet. On a découpé une ouverture sur la planche qui allait sur le dessus à l'endroit où l'écran d'ordinateur serait placé. On a mesuré et marqué les endroits où les boutons seraient placés. Pour les flippers, j'y ai été au touché avant de mesurer l'emplacement pour le copier sur l'autre côté. Sinon, les trois autres boutons, ceux pour projeter la balle, commencer une partie et mettre la partie en pause sont tous alignés à l'horizontale sur le côté avant du cabinet. Celui pour lancer la balle est situé à droite, comme sur une machine Pinball traditionnelle, et à égale distance du côté que du dessus et dessous du cabinet. Les deux autres boutons sont placés à gauche, le bouton pour démarrer la partie en premier en direction du milieu, suivi par celui pour mettre le jeu en pause. Heureusement, les boutons pour lancer la balle et démarrer la partie sont indiqués par du texte sur la tête du bouton lui-même. Pour alimenter la machine, j'ai utilisé une extension noire de 9 pieds comprenant deux prises électriques au bout et qui sort du derrière du cabinet. Afin de faciliter le débogage de la machine, on a posé le Raspberry Pi et le *Pinscape* sur la planche de dessus pour que je puisse dévisser le fond du cabinet et accéder pleinement à toutes les composantes. On a également

percé des trous dans le fond du boîtier pour permettre la circulation de l'air et de refroidir l'ordinateur, ainsi que des fentes dans le devant où est installé le haut-parleur.

Après cela, on a assemblé tout le cabinet pour le sabler et s'assurer que tous les morceaux étaient bien alignés. On a ensuite peinturé les morceaux complètement en noir mat, installé toutes les composantes à l'intérieur. En tout, le cabinet faisait 16½ pouces de large par 19½ pouces de profond et 7½ pouces de hauteur. On a également ajouté de courtes pattes en dessous afin qu'on puisse poser le cabinet sur une table et qu'il soit penché vers le joueur.

2.6 Opportunités d'emploi

Bien que l'industrie des cabinets d'arcade ait considérablement diminué depuis plusieurs années, j'ai quand même trouvé quelques offres d'emploi en ligne. Bien souvent, on recherche des techniciens qui peuvent venir réparer des machines d'arcade, tant de jeux vidéo que de machines à bille, à des centres de jeux ou à tout endroit. Il est demandé que l'employé se dirige à l'endroit parce qu'on ne peut pas facilement transporter un cabinet d'arcade n'importe où. Sinon, il y a également des emplois d'assembleur de cabinets et ceux-ci ressemblent beaucoup plus à ce que j'ai fait dans mon projet. Un des emplois que j'ai trouvés se situe à Terrebonne chez la compagnie *Adrenaline Amusements*, qui est une compagnie spécialisée dans la construction de cabinets de jeux d'arcade. Le travail est très manuel et consiste à assembler les cabinets d'arcade dans une usine de machines. Le salaire de base est de 18 \$ de l'heure et l'horaire est de 7 h 30 à 16 h du lundi au vendredi. Même s'il est connecté au domaine du jeu vidéo, l'emploi semble beaucoup plus traditionnel que d'autres emplois du domaine.

CONCLUSION

Pour conclure, je suis très content du résultat final de mon projet et je ne regrette pas le temps que j'y ai mis dessus. Cependant, je dois avouer que ce genre de projet sera probablement intéressant qu'à de grands amateurs d'arcades. Le coût des pièces à acheter et commander en ligne ainsi que le temps demandé pour le projet sont des facteurs très importants afin de déterminer si cela vaut la peine de le commencer. En revanche, je crois que si vous désirez les avantages qu'une machine virtuelle procure, cela vaut la peine. La simulation d'un véritable billard électrique est très bonne, même pour un format qui est considérablement plus petit. Le cabinet de bois est très solide et je n'ai pas peur de le secouer un peu. Le jeu roule également très bien, ce qui m'a un peu surpris pour la puissance de l'ordinateur. Je suis aussi très content de la qualité des boutons que j'ai achetés, ainsi que le jeu démarre automatiquement quand on allume la machine.

Pour l'instant, même si le coût de faire un tel projet est un peu élevé, je crois que cela sera plus abordable dans le futur avec la récente croissance en popularité des jeux vidéo rétros. De plus, il est assez facile pour de nouveau arrivants de s'introduire dans ces genres de projets avec la quantité et la qualité des guides publiés sur Internet.

ANNEXE

Image 1

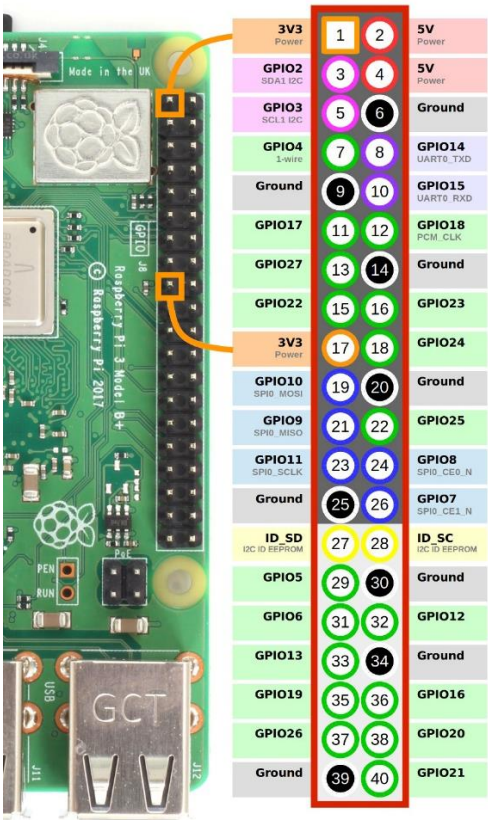


Image 2



Image 3



Image 4

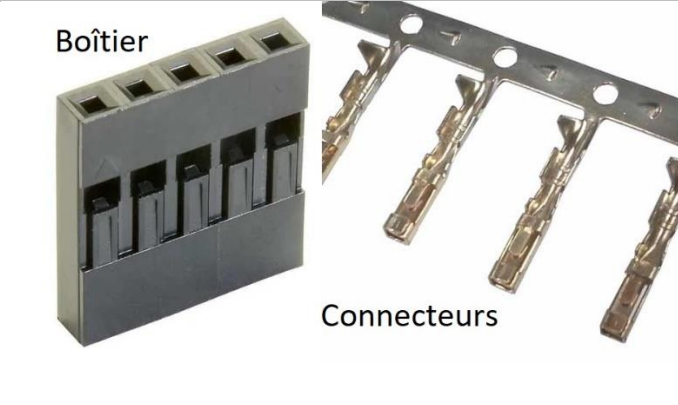


Image 5

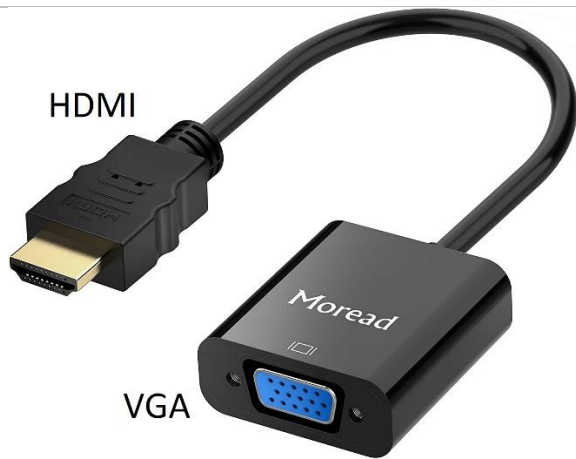


Image 6



Image 7

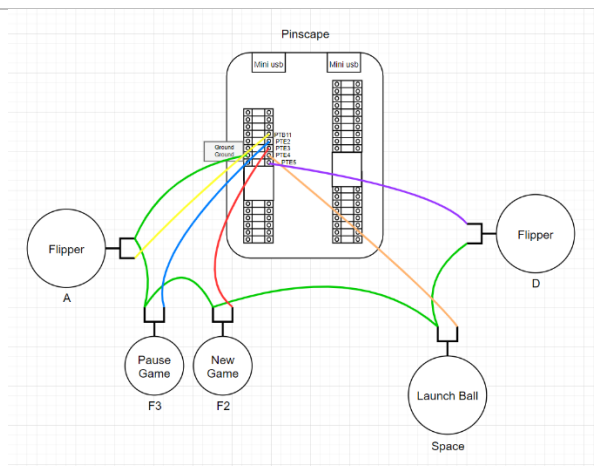


Image 8



Image 9

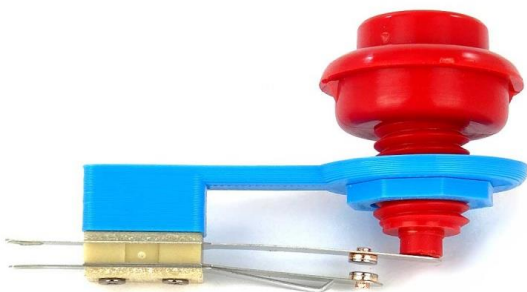


Image 10



Image 11

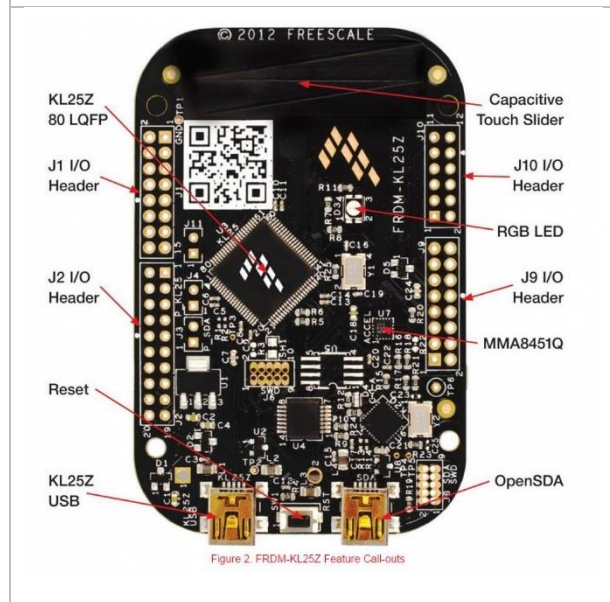


Image 12

