

INSTITUT UNIVERSITAIRE D'INFORMATIQUE

ANNÉE 2017-2018



RAPPORT FINAL DE PROJET BORNE D'ARCADE



Version 1.1

Auteurs : Romaric Bougard / Pierre Delobel / Bastien Ducloy

Tuteur :
Rémi Synave

UNIVERSITÉ DU LITTORAL CÔTE D'OPALE
IUT CALAIS-BOULOGNE
Rue Louis David - 62100 CALAIS
Téléphone (33) 03.21.19.06.60 Télécopie (33) 03.21.19.06.61

Remerciements

Nous tenons tout d'abord à remercier le lycée Giraux Sannier qui nous a fourni la structure en bois nous permettant d'avoir une base sur laquelle construire notre borne.

Nous tenons aussi à remercier le Fablab Côte d'Opale qui nous a accueilli, et plus particulièrement Laurent Zunquin, directeur du Fablab, qui nous a guidé et prodigué des conseils techniques tout au long de la réalisation de notre projet.

Nous tenons également à remercier Franck Vandewiele pour son implication dans la modification de la bibliothèque graphique MG2D afin d'avoir le mode plein écran.

Toute notre gratitude revient à Rémi Synave, notre tuteur, sans qui ce projet n'aurait jamais vu le jour. Nous le remercions pour son aide et ses conseils apportés au fur et à mesure de l'avancée du projet, mais aussi pour son implication ainsi que la motivation qu'il a su nous apporter.

Nous remercions également l'Institut Universitaire de Technologie du Littoral Côte d'Opale ainsi que l'ULCO de nous avoir proposé ce projet et son financement.



FABLAB
CÔTE D'OPALE



Lycée
Giraux Sannier

Page de garde	
Remerciements	2
Introduction	4
But du document	4
Cadre du document	4
Contexte	4
Objectifs	4
Caractéristiques	5
Etude du projet et présentation général	6
Contraintes et situations initiales	6
Élaboration du cahier des charges	6
Objectifs à réaliser	7
Matériel à disposition	7
Elaboration du projet	10
Les différentes phases et aspects du projet	10
Prise en main du Raspberry Pi 3B	10
Elaboration du menu JAVA	11
Explications techniques	13
Adaptation des jeux JAVA	14
Montage et finalisation du meuble	16
La répartition du travail	17
Les problèmes et les solutions envisagées	17
Résultats et perspectives sur le projet	18
Etat final du projet	18
Améliorations et extensions possibles	18
Conclusion	19
Annexe	20
Tutoriel d'intégration d'un jeu dans la borne	20

I. Introduction

1. But du document

Ce document a pour but de présenter l'ensemble des tâches réalisées durant la période du projet.

2. Cadre du document

Ce document est issu de la fin de la période disposée pour faire le projet. Nous expliquerons toutes les étapes depuis le mois d'Octobre jusqu'à aujourd'hui.

3. Contexte

Dans le cadre de notre seconde année de DUT Informatique nous avons dû réaliser un projet tutoré. Nous avons donc composé une équipe de 3, en attribuant les rôles suivants:

- Romaric Bougard, chef de projet
- Pierre Delobel, chargé de documentation
- Bastien Ducloy, chargé de communication

Notre choix s'est porté sur la réalisation d'une borne d'arcade faite main car c'est le projet qui coïncidait le plus avec nos compétences ainsi que nos envies. Ce projet était proposé par Rémi Synave qui est par la suite devenu notre tuteur.

4. Objectifs

Tous les ans, les étudiants développent des petits jeu en java. Le but du projet est la fabrication d'une borne d'arcade permettant la mise en valeur des meilleurs projets étudiants. Cela permet de créer une certaine motivation chez l'étudiant car son jeu pourrait être rendu jouable sur un vrai support destiné au jeu.

Par la suite, la borne servira de démonstration lors des forums ou des salons étudiants et elle permettra de présenter ce qui est réalisé lors des cours et ainsi de promouvoir le département Informatique de l'IUT de Calais.

Le reste du temps la borne sera accessible en dehors des cours dans une salle dédiée aux projets réalisés par les étudiants. Elle devra donc être jouable, fonctionnelle et esthétique.

5. Caractéristiques

Le meuble de bois de la borne sera fourni par un prestataire extérieur (ici le lycée Giroux Sannier). La borne devra être simple à transporter et l'électronique devra être facilement accessible afin d'apporter des modifications à posteriori ou rajouter des jeux par exemple. Elle devra aussi disposer de hauts-parleurs. L'écran devra être de préférence au format 4:3 afin d'adapter plus facilement les jeux provenant des étudiants.

Pour la partie logicielle, un menu doit être conçu et il se devra d'être "flexible", c'est-à-dire qu'on pourra ajouter autant de jeux qu'on le désire. La borne se doit d'être fluide, comme une vraie borne d'arcade, où chaque action utilisateur résulte d'une action à l'écran.

II. Etude du projet et présentation général

1. Contraintes et situations initiales

La plus grosse contrainte de ce projet était de savoir comment nous allions faire pour concevoir la borne. Après l'étude de multiples solutions, nous nous sommes tournés vers un Raspberry Pi modèle 3B (voir fig. 1) que nous avons à disposition.



Figure 1 : Photo d'un Raspberry Pi 3 modèle B

Nous avons choisi le Raspberry tout d'abord pour sa petite taille, afin de maximiser l'espace à l'intérieur de la borne, mais aussi pour son poids léger : la borne doit être transportable, il a donc fallu alléger un maximum son contenu. Nous avons aussi choisi ce nano-ordinateur pour son système d'exploitation Raspbian qui se base sur un noyau Linux et qui est donc totalement modifiable et personnalisable car il est open source, ce qui correspondait parfaitement à nos attentes. De plus, il dispose d'une connectique complète digne d'un ordinateur classique (port USB, HDMI...).

Nous étions également limités dans le temps. Nous avons débuté le projet le lundi 2 Octobre 2017. Nous avons de suite rédigé le cahier des charges et la charte graphique. En parallèle à cela, nous avons commencé le développement du menu de la borne en JAVA.

2. Élaboration du cahier des charges

Le cahier des charges a été réalisé sur Google Docs pour que toute l'équipe puisse voir son contenu et le modifier à sa guise.

Avant de commencer sa réalisation, nous avons pris rendez-vous avec le tuteur afin de définir vraiment les objectifs ainsi que les contraintes imposées. Par la suite, nous avons suivis la méthode MAETIC qui permet de fournir un travail efficace en groupe. La rédaction

du cahier des charges suit un objectif principal : la conceptualisation et la réalisation d'une borne d'arcade.

Pour remplir au mieux cet objectif, le menu réalisé doit être fonctionnel, fluide et ergonomique, les jeux mis à notre disposition doivent être fonctionnels et fluide, le meuble doit être design et facile à transporter.

3. Objectifs à réaliser

L'équipe de projet en accord avec l'enseignant tuteur/client, s'est accordée à remplir les objectifs suivants:

Au préalable, avant de commencer le développement en lui-même, il nous a fallu nous documenter sur les différentes conceptions d'une borne d'arcade, les différents types de bornes, etc... Nous avons donc choisi un format bartop, c'est à dire un format plus petit pour pouvoir le poser sur une table par exemple. Ce format réduit d'ailleurs le poids du meuble.

Ensuite est venue la deuxième étape du projet, où nous avons dû élaborer le cahier des charges ainsi que le charte graphique.

La troisième étape a été consacrée au développement du menu et à l'adaptation/optimisation des jeux JAVA des étudiants afin qu'ils puissent être jouable mais aussi fluides car le Raspberry Pi a des performances limitées.

4. Matériel à disposition

Suite à ces décisions, nous avons commandé le matériel nécessaire à la réalisation de la borne.

Nous avons premièrement contacté le lycée Giraux Sannier en leur fournissant les plans de la borne que nous voulions afin qu'ils mettent en oeuvre la création des panneaux de bois. Les plans ont été choisi par notre client (voir fig. 2).

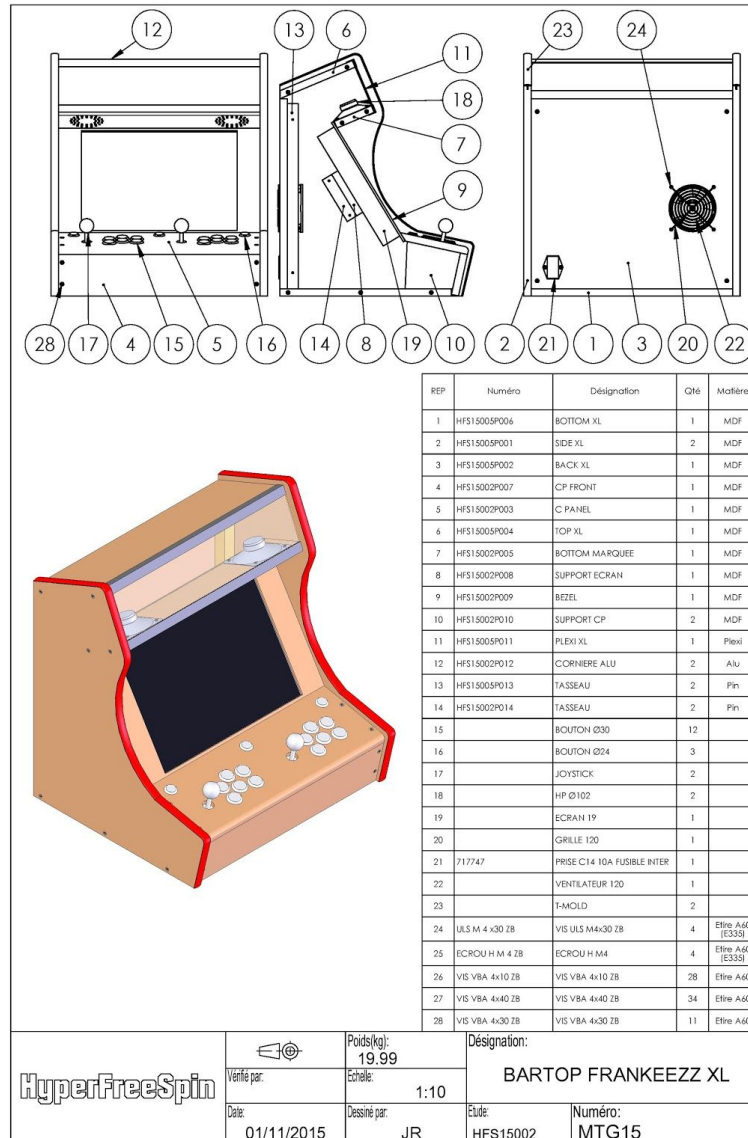


Figure 2 : Vue d'ensemble du plan fourni

Par la suite, nous avons eu besoin de boutons ainsi que d'un encodeur clavier afin que les pressions sur les boutons soient reconnues comme des touches de clavier classique pour que JAVA reconnaisse les actions faites par les joueurs. Nous nous sommes donc tournés vers un kit tel que celui ci-dessous (voir fig. 3).



Figure 3 : kit Raspberry de joysticks et de boutons — Smallcab

Les boutons devant être reliés à un ordinateur, nous avons choisi le Raspberry Pi 3 que nous avons présenté précédemment. Il sera relié à un écran 4:3 d'une résolution de 1280x1024. Le format 4:3 est privilégié lors des projets, nous avons donc choisi un écran au même format pour faciliter l'adaptation des jeux. Nous nous sommes aussi munis d'un amplificateur stéréo auquel on a connecté 2 hauts-parleurs afin d'avoir une sortie sonore.

III. Elaboration du projet

1. Les différentes phases et aspects du projet

Avant toute chose, notre projet a été découpé en différents phases de travail afin de planifier et de se répartir le travail au mieux. Nous avons découpé notre axe de travail en 4 grandes phases qui sont les suivantes :

- ❖ Phase 1 : Prise en main du Raspberry Pi 3B
- ❖ Phase 2 : Elaboration du menu JAVA
- ❖ Phase 3 : Adaptation des jeux JAVA
- ❖ Phase 4 : Montage et finalisation du meuble

1.a. Prise en main du Raspberry Pi 3B

Le Raspberry Pi 3B est un nano-ordinateur propulsé par une distribution Linux spécifique du nom de Raspbian, qui lui est propre. Il est équipé d'un processeur quatre coeurs cadencé à 1,2GHz. Soit des performances relativement limitées en ce qui concerne les tâches de grosse envergure. Lors de l'adaptation des jeux, nous avons évité au maximum l'affichage d'images en continu, car le Raspberry Pi le supporte mal.

Il dispose de 4 ports USB, indispensables pour nous car il nous a fallu brancher la souris, le clavier, aussi pour l'encodeur clavier pour tester les joysticks/boutons, et aussi une clé USB pour y mettre les jeux. Il dispose d'une sortie audio Jack 3.5mm, que l'on utilise pour les jeux qui disposent de sons mais aussi pour le menu. A noter aussi que le Raspberry ne pèse que 45g. Voici les fonctionnalités les plus importantes à noter concernant le Raspberry pour ce projet.

1.b. Elaboration du menu JAVA

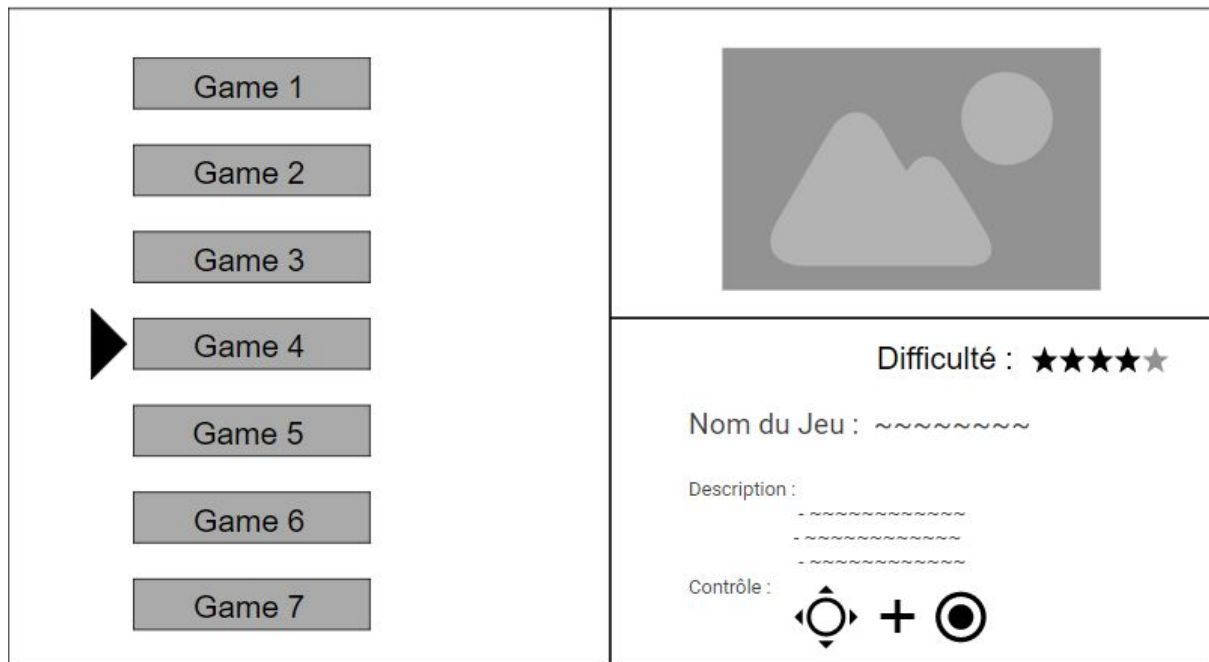


Figure 4 : Maquette du menu au début du projet

Avant d'entamer la phase de développement nous nous sommes intéressés au maquettage du menu, afin d'avoir une base à suivre pour avoir un ensemble cohérent. Cela nous a mené à un travail de conceptualisation où nous avons réfléchi aux différentes classes que nous allions créer. Une fois la maquette validée par le client nous avons commencé le développement.

Nous avons choisi la bibliothèque graphique MG2D pour le réaliser car nous avons déjà été confrontés à utiliser cette bibliothèque et nous savions comment en disposer. Nous avons tout d'abord utilisé de simples carrés afin de délimiter 3 grandes zones, puis nous avons constitué le menu autour de ces zones. Suite à quelques heures de travail le résultat était déjà proche de ce que nous recherchions (voir fig. 5).

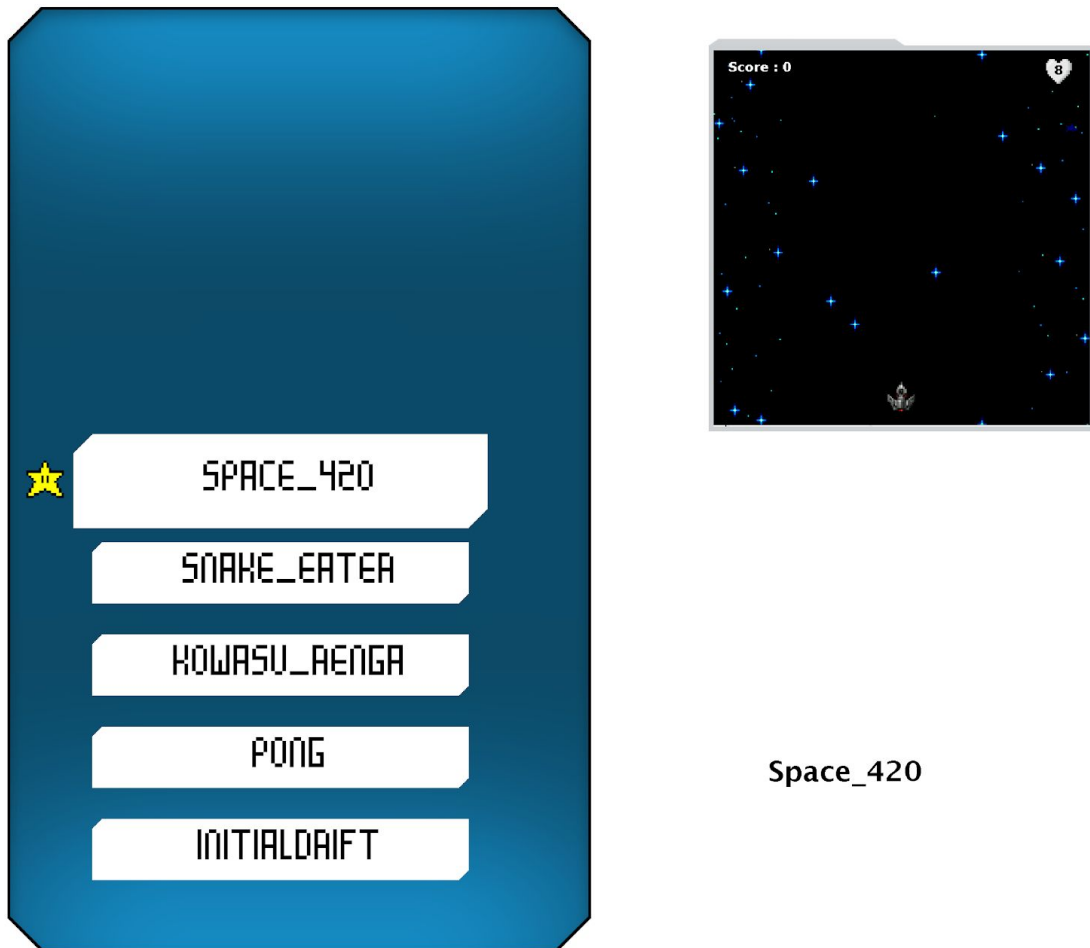


Figure 5 : Première version du menu

Le menu est flexible. On peut ajouter ou supprimer des dossiers contenant des jeux pour ajouter ou supprimer des jeux à la borne. Nous avons utilisé une police personnalisée pour l'affichage des noms des jeux afin de donner du relief et de la cohérence.

Chaque jeu comporte un titre, une image ainsi qu'une description. A chaque clic sur un jeu un script shell se lance et exécute la classe exécutable du jeu en question. L'exécution se fait en plein écran, couvrant alors le menu pour laisser place à l'espace de jeu. Une fois que le jeu est terminé ou que l'utilisateur le quitte, il revient directement sur le menu. A noter que chaque déplacement dans le menu est confirmé par un retour audio.

i. Explications techniques

A la fin, voici ce que donne le diagramme de classe UML du menu:

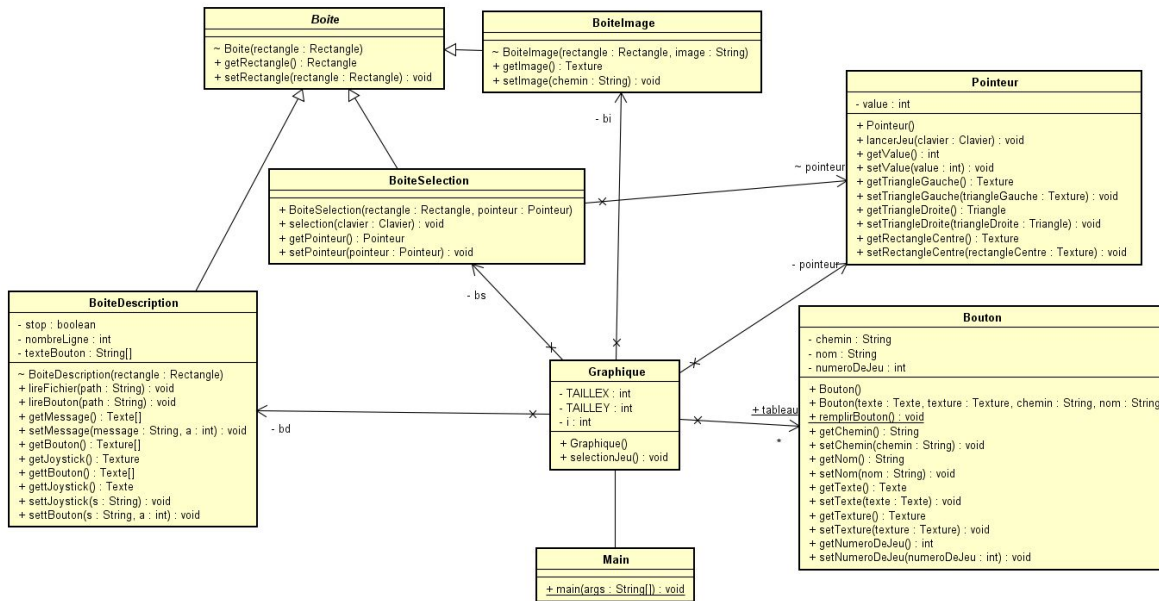


Figure 6 : Diagramme de classe UML du menu final

Les 5 classes de base du Menu sont :

- BoiteImage : elle s'occupe de l'image de chaque jeu.
 - setImage(chemin : String) : qui permet de modifier l'image du jeu sélectionné.
- BoiteSelection : qui s'occupe de la sélection des jeux.
 - selection() : méthode qui permet de naviguer dans la sélection des jeux.
- BoiteDescription : qui s'occupe de la description de chaque jeu.
 - lireFichier(path : String) : permet de lire le document "description.txt" pour afficher la description du jeu dans le Menu.
 - lireBouton(path : String) : permet de lire le document "bouton.txt" pour afficher la description des boutons dans le Menu.
- Pointeur : classe qui s'occupe de donner une valeur au jeu sélectionné dans la boîte de sélection.
 - getPointeur() : renvoie le pointeur actuel de la boîte de sélection.
 - getPointeur().getValue() : renvoie un entier désignant la "position" du jeu sélectionné.
 - setPointeur(Pointeur : pointeur) : modifie le pointeur par celui passé en paramètre.
- Graphique : est la classe qui sert uniquement à gérer le graphisme du Menu.
 - selectionJeu() : méthode globale pour lancer le Menu.

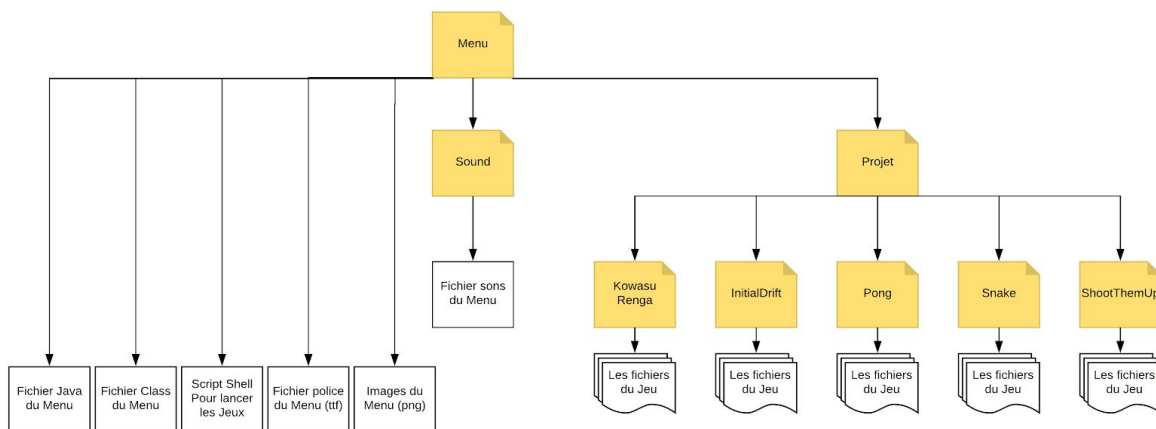


Figure 7 : Arborescence du Menu de la borne

1.c. Adaptation des jeux JAVA

- Pong : Nous avons tout d'abord adapté le Pong à la résolution de notre écran. Ensuite, nous avons remarqué qu'il était assez monotone. Nous avons ajouté de la vitesse à la balle et aussi réduit la taille des raquettes au fur et à mesure de la partie. Nous avons aussi modifié les entrées clavier afin de les faire correspondre aux touches retournées par les mouvements des joysticks.
- Kowasu Renga : Nous avons premièrement adapté le jeu à la résolution de notre écran. Nous avons vite constaté un taux d'images par seconde très bas (~20 IPS) ce qui rendait le jeu quasiment injouable. Nous avons corrigé le problème en changeant le compteur de points avec des images, qui est très consommateur en ressources, par un compteur avec du texte seulement. Comme précédemment nous avons dû réadapter les touches.
- SpaceShoot : La fenêtre était déjà à la bonne résolution 1280x1024, mais le jeu avait un taux de 20 images par seconde, donc injouable. On a dû l'optimiser au niveau du chargement des images mais aussi modifier le compteur d'image en compteur de texte. Finalement, nous avons aussi modifier les entrées clavier. Nous avons laissé 6 secondes à la fin du jeu pour qu'il se quitte automatiquement.
- Initial Drift : Suite à l'adaptation de la résolution, nous avons rajouté des bandes sur les côtés pour éviter d'avoir trop de voies de circulation. Cela aurait rendu le jeu trop facile. Nous avons optimisé le code pour éviter les ralentissements. Nous avons modifié la vitesse du défilement du décor sur les côtés pour rendre le jeu plus crédible et plus fluide. Nous avons aussi modifié les touches. Nous avons réduit le temps de la musique de fin. En effet, elle durait de base 2h.

- Snake_Eater : Suite à l'adaptation de la résolution, nous avons rajouté des bandes noires tout autour pour éviter que le terrain de jeu soit trop grand ce qui aurait rendu le jeu ennuyant. Nous avons d'ailleurs changé l'image de fond car il était difficile de distinguer le serpent.

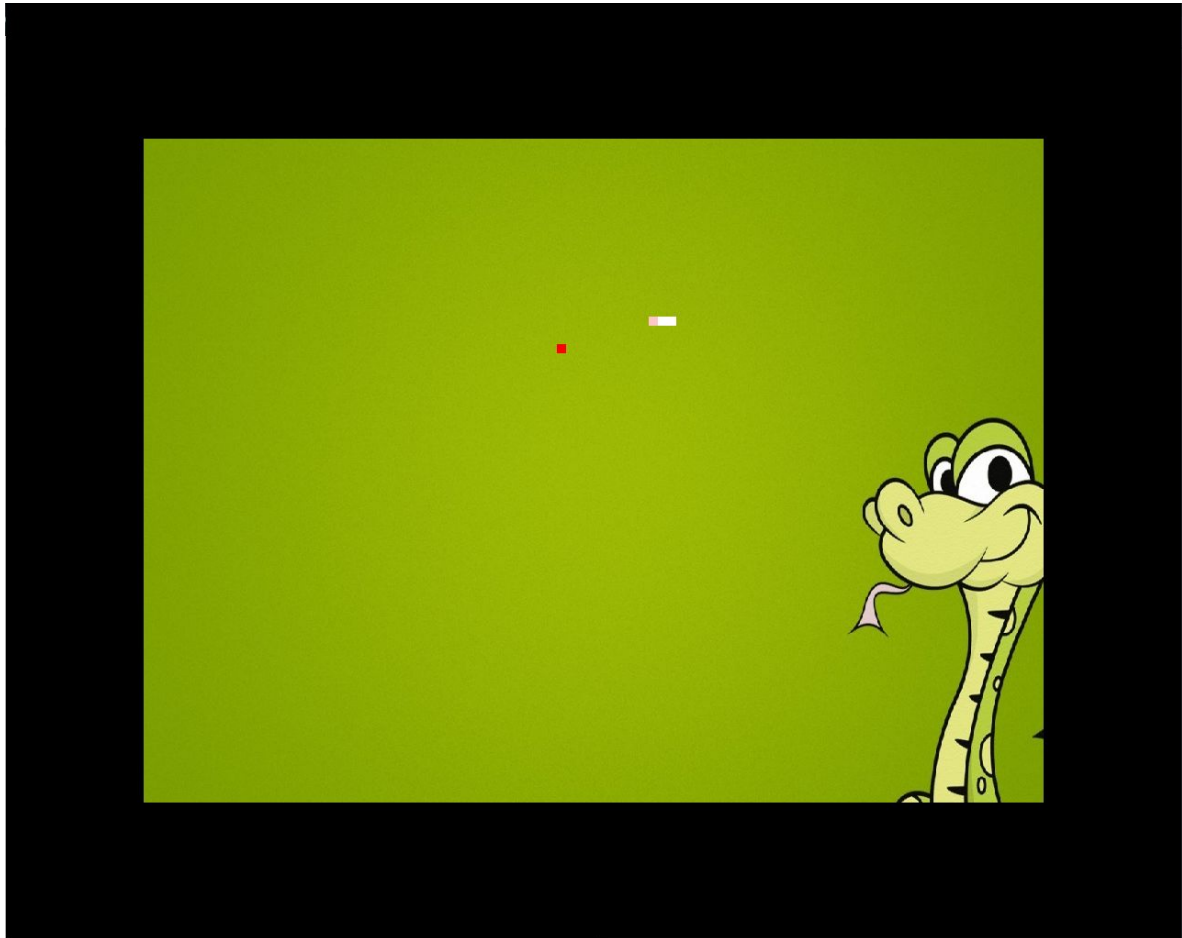


Figure 8 : Version finale du Snake Eater

1.d. Montage et finalisation du meuble

Nous avons reçu le meuble fabriqué par le lycée Giroux Sannier de Boulogne, en février. Nous l'avons donc emmené au FabLab. Nous avons constaté que le meuble était mal fini (cadre de l'écran pas droit, planche sur les côtés non parallèles, trou de la planche de derrière mal formé, fente pour le t-molding inexistante, trous pour les hauts-parleurs non percés). Cela a rajouté une contrainte, et nous avons perdu du temps car nous avons dû finaliser la découpe le bois.

Nous avons donc commencé par peindre toutes les planches en blanc. Nous avons ensuite agrandi les trous pour les boutons qui en avaient besoin et placé les boutons. Nous avons fait les trous pour les enceintes et les avons placées et fixées ainsi que les grilles. Nous avons aussi fait un trou pour placer l'amplificateur audio.

2. La répartition du travail

Étant trois membres dans notre équipe, la répartition s'est faite comme suit :

- Pierre Delobel s'est chargé de réaliser la documentation, l'adaptation des jeux JAVA ainsi que la finalisation du meuble.
- Romaric Bougard s'est chargé de relire la documentation, du câblage ainsi que la réalisation du design du menu.
- Bastien Ducloy s'est chargé de l'élaboration du menu, de l'adaptation des jeux JAVA ainsi que la finalisation du meuble.

3. Les problèmes et les solutions envisagées

La majeure partie des problèmes se sont situés au niveau du développement du menu et de l'adaptation des jeux. En effet, il a fallu ajouter une description pour chaque jeu, nous avons donc dû utiliser un fichier texte pour chaque jeu et y mettre la description pour pouvoir l'afficher sur le menu en JAVA. Ensuite, il a fallu trouver un moyen de compiler et lancer le jeu en appuyant sur entrée depuis le menu. Nous avons donc créé un script Bash pour chaque jeu qui effectue la compilation et l'exécution du jeu qui se lance en appuyant sur entrée quand on sélectionne le jeu.

Un autre problème était le retard de la livraison du meuble qui nous a freiné dans la finalisation du projet. De plus, de nombreuses imperfections étaient présentes sur le meuble mais elles ont été corrigées.

Enfin, nos horaires de projet constituaient elles aussi un problème quand on prévoyait d'aller au FabLab. Beaucoup de nos heures de projet étaient divisées, au lieu d'être regroupées.

IV. Résultats et perspectives sur le projet

1. Etat final du projet

A ce jour, les différents points fonctionnant de la borne sont les suivants :

Premièrement, le menu est pleinement fonctionnel. Il est fluide, permet de lancer un jeu sélectionné et est aussi esthétique. La souris n'est pas voyante à l'écran, cela permet une meilleure expérience pour les joueurs.

Ensuite, les jeux sont eux aussi fonctionnels. Ils ont été adaptés à l'écran mis à disposition, tournent correctement même si on pourrait encore améliorer la fluidité en réduisant les chargements d'images, et sont refermables ou se quitte tout seul à la fin.

Le meuble lui, est presque fini, il ne reste que certains détails à finir, comme l'ajout d'un bouton ON/OFF au dos de la borne ou encore coller les stickers latéraux.

2. Améliorations et extensions possibles

Notre borne d'arcade étant fonctionnelle, il reste néanmoins des imperfections et des points à améliorer :

- En particulier, le meuble n'est pas fini totalement et quelques détails sont à finir comme dit précédemment.
- Il faudrait trouver un moyen d'optimiser encore plus la partie logicielle pour pouvoir avoir une expérience utilisateur encore meilleure.
- Un affichage de la difficulté pourrait être ajouté sur le menu comme prévu sur la maquette.

V. Conclusion

Dans le cadre de notre projet, nous devions réaliser une borne d'arcade pour la présentation du DUT Informatique de Calais dans les forums. Dans un premier temps, nous avons dû nous documenter sur ce qu'était précisément une borne d'arcade. Ensuite, nous avons effectué le travail en commençant de rien en étudiant les meilleurs moyens pour avoir une partie logicielle fonctionnelle. De plus, nous avons dû habiller et modifier le meuble reçu pour qu'il corresponde à ce que nous avions prévu.

A travers ces différentes exigences imposées par le cahier des charges, nous aboutissons au résultat final qui est le suivant :

- Le menu codé en JAVA est fini. Il est fonctionnel, fluide, design. Nous pouvons lancer des jeux depuis ce dernier et nous pouvons rajouter des jeux dans le menu juste en créant un dossier dans le dossier "projet" du dossier "Menu". Il vous faudra aussi créer un fichier BASH qui permet de lancer le jeu. (voir le tuto pour l'intégration d'un jeu dans la borne en annexe). Il présente les jeux avec une description, une image et les contrôles du jeu.
- Tous les jeux sont jouables, fluides et les bugs ont été corrigés. Nous pouvons quitter les jeux et revenir au menu.
- En ce qui concerne le meuble, il nous reste quelques détails à finir. Il faut fixer l'écran, mettre le bouton d'alimentation, finir le câblage du Raspberry ainsi que de l'écran et enfin mettre les poignets sur les côtés.

A partir de ces éléments, l'ensemble du groupe pense avoir réussi à concevoir une borne d'arcade fonctionnelle pouvant être utilisée par tous les utilisateurs et répondant aux exigences du client. Certes quelques détails sont à finir, mais nous avons encore 1 semaine à compter du jour où nous rendons ce rapport pour peaufiner et clôturer le projet.

D'un point de vue humain, le travail en groupe est une bonne chose, cela permet de travailler en équipe, de respecter les choix et les idées de tous les membres afin de servir une cause commune. Également, le fait de se répartir le travail permet de se concentrer sur une partie précise où l'on va s'investir à fond dans l'intérêt du groupe.

VI. Annexe

1. Tutoriel d'intégration d'un jeu dans la borne

Tutoriel pour l'intégration d'un jeu dans la borne d'arcade:

Requis :

- Résolution du jeu : 1280x1024
- Touche clavier :
 - Joystick : ←,↑,→,↓ : O,K,L,M
 - Boutons : A,Z,E,Q,S,D : 1,2,3,4,5,6

Étape 1 : Premièrement, Il faut ajouter dans votre projet une image “preview” qui permettra de donner un visuelle sur votre Jeu.

Exemple :

Pour cela, déplacez vous à la racine de votre jeu et ajouter votre image. celle-ci devra impérativement s'appeler “photo.png”.



Étape 2 : Il faut aussi ajouter dans votre projet un fichier texte qui contiendra une description de votre Jeu, celle-ci ne devra pas dépasser 10 lignes sous peine d'une suppression des lignes supplémentaires.

Pour cela, déplacez vous à la racine de votre jeu et ajoutez votre description. celle-ci devra impérativement s'appeler “description.txt”.

Exemple :

```
1  Auteur : Loan Caecke
2  Année : 2017
3  Type : Jeu de Voiture
4  Description :  Jeu développé dans le cadre
5                  du module M2103 - Programmation
6                  Orientée Objet en 1ère année
7                  de DUT Informatique.
```

Étape 3 : Il faut aussi ajouter dans votre projet un fichier texte qui contiendra une description des différentes utilisations des touches et joystick de votre jeu, celle-ci contiendra donc : la description du joystick puis la description de chaque bouton individuellement.

Il est important que chaque description soit séparée par un “:” sans retour à la ligne.

Exemple :

```
1  Mouvement voiture:aucun:aucun:aucun:aucun:aucun:aucun
```

Pour cela, déplacez vous à la racine de votre Jeu et ajoutez votre description de boutons. Celle-ci devra impérativement s'appeler “bouton.txt”.

Étape 4 : Il va falloir, pour lancer votre jeu dans la borne d'arcade, créer un “script shell” très simple.

pour cela, créer un fichier avec l'extension “.sh”.

Exemple :

```
#!/bin/bash
xdotools mousemove 1280 1024
cd projet/InitialDrift
java Main
```

Écrivez un script comme celui de l'exemple modifiez juste “InitialDrift” par le nom du répertoire de votre projet.

Une fois votre script fini, déplacez le à la racine du Menu de la borne d'arcade.

Étape 5 : Déplacez vous dans le répertoire “projet” du Menu de la borne : “Menu” ► “projet”

Étape 6 : Déplacer votre projet dans ce répertoire et **n'oubliez pas de compiler vos fichiers “.java” dans le même répertoire !**