

Fallen Dohyo
Cahier des Charges Technique

Willy JACQUET Yoan PARRA

3 Mars 2019

Table des matières

I	Présentation	2
II	Description, <i>Ten-Pager</i>	3
II.1	Présentation	3
II.2	Personnage	3
II.3	Jouabilité	4
II.4	Univers	4
II.5	Expérience, <i>Gestalt</i>	4
II.6	Mécaniques	5
II.7	Rejouabilité	7
II.8	Modèle économique	7
III	Contraintes	8
III.1	Livrables	8
III.2	Temporalité	8
III.3	Budget	8
IV	Déroulement	9
IV.1	Implémentation	9
IV.2	Tâches	9
V	Annexes	11
V.1	Diagramme de Gantt	12
V.2	Diagramme des classes	13

I. Présentation

Ce projet s'inscrit dans le cadre de l'unité d'enseignement LIFAP4 - Conception et développement d'applications, à l'université Claude Bernard Lyon 1, sous la supervision d'Alexandre MEYER et de Nicolas PRONOST.

L'objectif de ce projet est d'avoir un premier contact avec la conception et le développement d'applications, et ce, de manière formelle et rigoureuse. Ce projet est mené par deux étudiants :

- Willy JACQUET, p1806811
- Yoan PARRA, p1505740

Relativement peu de restrictions sont imposées (cf. [III](#)), nous avons choisi de concevoir et de développer un jeu vidéo.

La suite de ce cahier décrit le jeu en question, les contraintes et le déroulement du projet, et enfin, en annexes, se trouve un diagramme de Gantt et un diagramme des classes.

II. Description, *Ten-Pager*

II.1. Présentation

Fallen Dohyo est jeu vidéo de combat et de plateforme multijoueur mettant en scène deux joueurs dans une arène dont le but est d'éjecter l'autre. Jeux similaires :

- Céleste
- Super Meat Boy
- Super Smash Bros
- Tower Fall

II.2. Personnage

Vous incarnez un esclave moderne de la société ; harcelé et exploité par votre patron, le mariage en péril...



Au détour d'un coin perdu, vous entendez des hurlements de rage. Vous découvrez ainsi une arène sauvage, où tous sont aussi pathétiques que vous.

Vous enlevez votre costume, le pliez et le posez. Vous joignez ces affrontements dénués de sens ; vous allez vous laver de l'humiliation qui vous a été faite.

II.3. Jouabilité

Fallen Dohyo se joue avec peu de contrôles. Deux joueurs peuvent jouer sur un même clavier.

Les contrôles consistent en quatre commandes directionnelles et une commande pour foncer, ainsi qu'un bouton pause.

II.4. Univers

Les arènes de **Fallen Dohyo** se situent à différents endroits du monde, à l'abri des regards ; là, où des hommes n'ont aucune valeur.

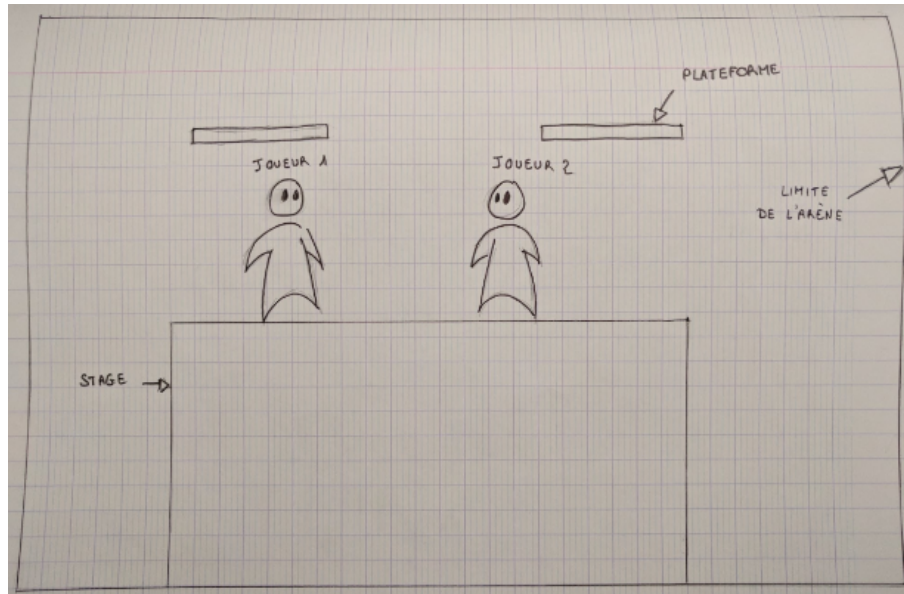


II.5. Expérience, *Gestalt*

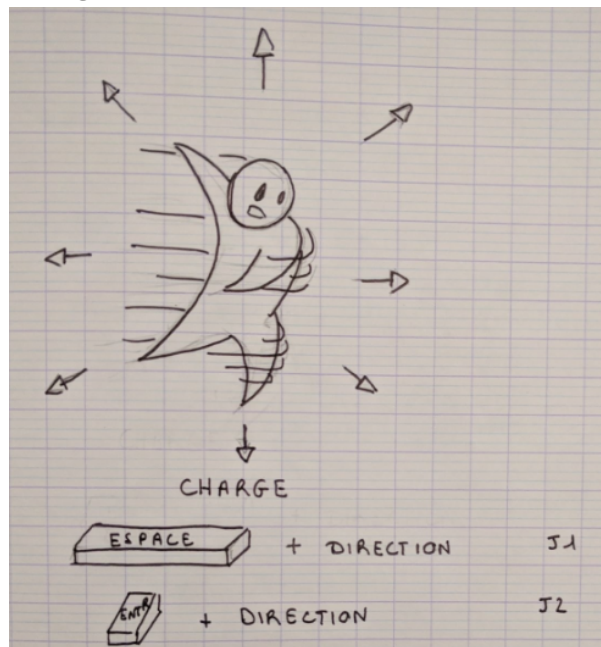
En quelques secondes, deux joueurs peuvent débiter une partie. Les contrôles se veulent réactifs et nerveux.

Fallen Dohyo est un défouloir excitant et compétitif épousant la loi de Bushnell : *easy to learn, hard to master*.

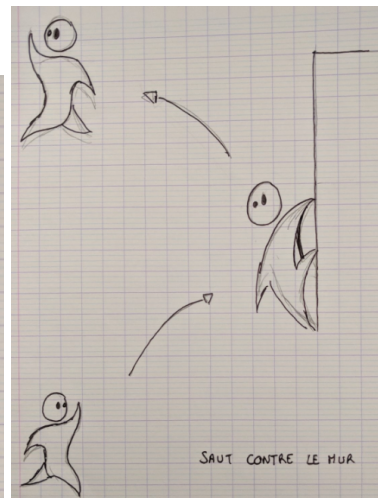
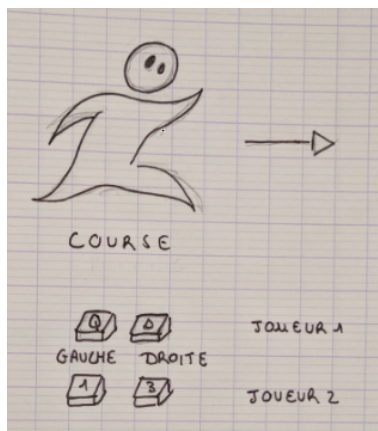
II.6. Mécaniques



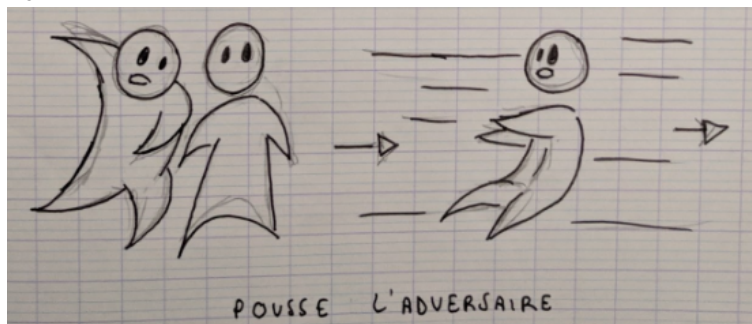
- Charge.



- Déplacements.



- Éjection.



- Saut.



II.7. Rejouabilité

Fallen Dohyo est un jeu d'arcade multijoueur, les parties sont rapides et sans prérequis, il sera donc joué occasionnellement entre amis, tel un jeu d'arcade, ou en compétition.

II.8. Modèle économique

L'objectif financier de **Fallen Dohyo** n'est pas de générer un bénéfice mais de rembourser les coûts de développement et de gagner en visibilité dans le secteur du jeu vidéo. Le produit sera vendu de manière unitaire pour une durée illimitée.

Le produit ne comportera pas de microtransactions ou d'extensions payantes car nous, les développeurs, sommes hostiles à ces pratiques et mettons un point d'honneur à ne pas les implémenter. Des mises à jour correctives seront distribuées gratuitement. Le prix estimé du produit devrait être dans la tranche de 2€ à 5€.

Seules des copies dématérialisées seront distribuées, c'est à dire qu'il n'y aura pas de production d'un support physique du produit. La distribution du produit sera confiée à une plateforme tiers, telle que Steam ou GOG.

III. Contraintes

Le produit final, ainsi que sa conception et sa réalisation, est soumis à des consignes dont le détail est disponible au terme les liens suivants :

- [Barème de notation](#)
- [LIFAP4 - Conception et développement d'applications](#)

III.1. Livrables

Le produit final est constitué d'un exécutable pour les systèmes Windows (.exe) et de bibliothèques de liens dynamiques (.dll), d'un exécutable pour les systèmes GNU Linux (.out), des sources ainsi que de la documentation technique et fonctionnelle.

III.2. Temporalité

La réalisation du produit s'effectuera du mardi 26 février 2019 au mardi 6 mai 2019.

- jeudi 7 mars 2019, dépôt du cahier des charges technique sur la plateforme Tomuss.
- mardi 26 mars 2019, présentation de la démonstration intermédiaire.
- lundi 6 mai 2019, dépôt du produit final sur la plateforme Tomuss.
- mardi 7 mai 2019, présentation du produit final.

III.3. Budget

Aucune rémunération, ni aide monétaire n'est prévu pendant ou à terme de la réalisation du produit. Par conséquent en tant qu'étudiants, nous, les développeurs, ne pourrons pas accorder plus de quelques dizaines d'euros à la réalisation du produit.

Ce budget sera investi principalement dans l'achat de ressources graphiques (scènes, textures, *sprites sheet*, etc) et audio (effets sonores, musiques, etc).

IV. Déroulement

IV.1. Implémentation

Le langage de programmation utilisé est le C++.

Le code source et la documentation technique sont en anglais, tandis que la documentation fonctionnelle est en français.

Les bibliothèques *Standard Template Library* et *Simple and Fast Multimedia Library* sont utilisés à divers usages.

IV.2. Tâches

1. **Écriture du makefile.**

Fichier ./makefile.

Cibles pour les exécutables, la documentation et le nettoyage.

2. **Écriture du readme.**

Fichier ./readme.md.

Introduction au projet ; à son usage, à ses dépendances, à sa politique, à sa documentation.

3. **Création du Doxyfile.**

Fichier ./doc/Doxyfile.

Configuration de la génération automatique de la documentation.

4. **Écriture du cahier des charges.**

Fichiers ./doc/technical_specification/*.

Cahier des charges techniques, diagramme de Gantt, diagramme des classes.

5. **Conception du module Audio.**

Fichiers ./src/audio/*.

Gestion, stockage et chargement des sons, des musiques et des effets sonores.

Liaison avec la bibliothèque SFML.

6. **Conception du module Graphics.**

Fichiers ./src/graphics/*.

Gestion, stockage et chargement des fenêtres, des sprites, des textures, des caméras et des shaders.

Affichage de la fenêtre et des éléments graphiques.

Liaison avec la bibliothèque SFML.

7. Conception du module Logic.

Fichiers ./src/prototype/*, ./src/logic/*.

Gestion du déroulement du jeu. Logique interne, implémentation du gameplay.

Création nombreuses de prototypes.

Dépend du module Physics.

8. Conception du module Physics.

Fichiers ./src/physics/*.

Outils pour la résolution de problèmes mathématiques ou physiques.

Détection et réponses des collisions, mécanique classique (position, vitesse, accélération).

9. Conception du module System.

Fichiers ./src/system/*.

Coordination des différents modules entre eux.

Élément central de l'application.

10. Conception du module UI.

Fichiers ./src/UI/*.

Éléments interactifs (boutons, menus, texte), gestion de la souris, du clavier et des joysticks.

Dépend du module Graphics.

11. Conception de la démonstration intermédiaire.

Fichier ./bin/demonstration.(exe/out).

Exécutable qui sera présenté lors de la démonstration.

12. Conception du produit final.

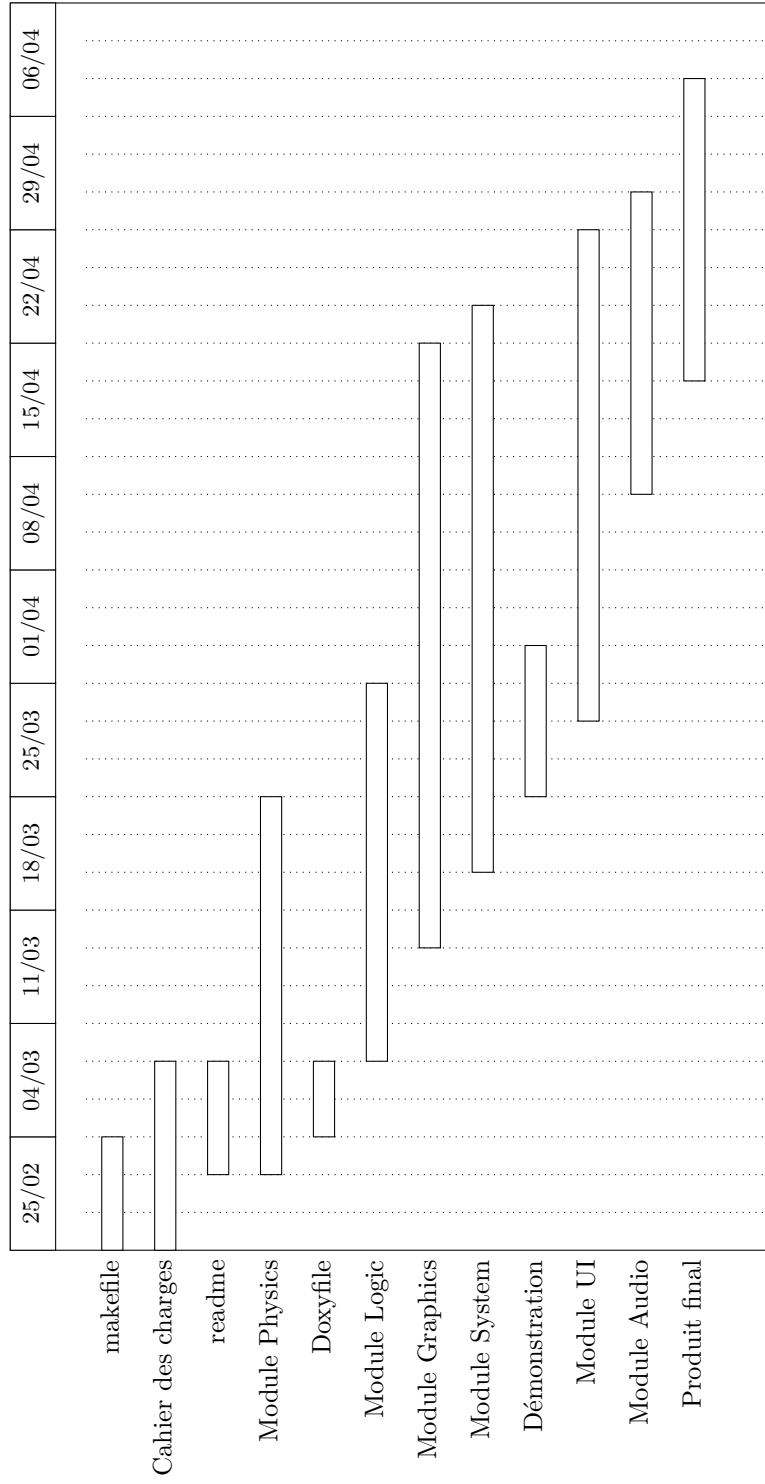
Fichier ./bin/final.(exe/out).

Exécutable contenant le jeu final.

Nettoyage des sources.

V. Annexes

V.1. Diagramme de Gantt



V.2. Diagramme des classes

