Thibodeau, Charles-David – 1634039  
Quessy, Evan - 1661769  
Prévost, Alexy - 1636282  
Jason Morneau - 1664122

Profession de programmeur   
420-V10-SF

Planification du projet de Hoffnug

Travail présenté à Benjamin Lemelin  
2 décembre 2016

Techniques de l’informatique – Programmation de jeux vidéo  
Cégep de Sainte-Foy

Préface

Le document suivant présente la planification du projet de développement du jeu « Hoffnug ». Ce projet, fictif, est planifié dans le cadre de Profession de programmeur (420-V10-SF) de M. Benjamin Lemelin.

L’entreprise Foktup Games, citée dans ce document, est fictive est n’est utilisée qu’à titre d’exemple. Aucune entente, contrat, ou projet n’a réellement eu lieu, et toute ressemblance avec une entreprise réelle serait fortuite.

Ce document n’a qu’un but éducationnel et ne peut être utilisé dans d’autres cadres.

Sommaire exécutif

|  |
| --- |
| **Titre du Projet** :  **Nom de la société** : Foktup Games  **Début du projet** : 2 octobre 2017  **Fin du projet** : 2 janvier 2019  **Objectifs** :  *Échéance* : 2 octobre 2017  *Budget* : 300 000$  *Spécifications* : Réalisation d’un jeu de défense nommé « » sur Android et iOS ayant des micro transactions où l’utilisateur doit défendre un village d’une invasion ennemie. |

Table des matières

[1 Introduction 3](#_Toc468445481)

[1.1 Contexte et nature du projet 3](#_Toc468445482)

[1.2 Objectifs du projet 3](#_Toc468445483)

[2 Modélisation du projet 3](#_Toc468445484)

[2.1 Méthode d’ingénierie 3](#_Toc468445485)

[2.2 Tâches du projet et préalables 3](#_Toc468445486)

[2.2.1 Développement des mécaniques du jeu 4](#_Toc468445487)

[2.2.2 Créations des éléments artistiques 4](#_Toc468445488)

[2.2.3 Mise en place des serveurs 4](#_Toc468445489)

[2.2.4 Mise en place du service de paiement en ligne 4](#_Toc468445490)

[2.3 Modélisation des ressources utilisées 4](#_Toc468445491)

[2.4 Réseau de tâches 5](#_Toc468445492)

[2.5 Criticité, risques du projet 5](#_Toc468445493)

[3 Analyse et choix de la planification définitive 6](#_Toc468445494)

[3.1 Calendrier définitif 6](#_Toc468445495)

[3.2 Coût de réalisation du projet 6](#_Toc468445496)

[4 Plan d’exécution 7](#_Toc468445497)

[4.1 Dates des points de contrôles 7](#_Toc468445498)

[4.2 Tâches à surveiller 7](#_Toc468445499)

[4.3 Structure organisationnelle 7](#_Toc468445500)

[4.4 Recommandations générales 7](#_Toc468445501)

[5 Conclusion 8](#_Toc468445502)

[Bibliographie A](#_Toc468445503)

# Introduction

## Contexte et nature du projet

Le projet présenté dans ce document est le développement d’un jeu de défense nommé « », le premier jeu de « » et donc un grand pas dans le marché grandissant des jeux vidéo.

Le jeu sera développé sur Unity afin de faciliter l’apprentissage et le développement de l’entreprise pour, possiblement, offrir un meilleur produit dans un futur proche.

## Objectifs du projet

Le projet est d’une durée de 16 mois, soit du 2 octobre 2017 au 2 janvier 2019. Les critères suivants devront être satisfaits à l’échéance :

1. L’application devra être disponible sur le Play Store (Android) et le App Store (iOS).
2. Les mécaniques du jeu devront respecter les récits utilisateurs décrits dans le document de conception.

# Modélisation du projet

## Méthode d’ingénierie

Ce projet suivra le mode de réalisation de la méthodologie Agile Scrum. Cette méthodologie consiste à séparer le programme en petits bouts de programmation dans le but de rendre le développement facile et rapide. En gros, les programmeurs, à l’intérieur d’un délai de 1-2 semaines, devront réaliser certaines fonctions du jeu. Après un délai, les programmeurs se regroupent pour discuter de ce qu’ils peuvent faire dans le prochain délai.

## Tâches du projet et préalables

Conformément au document de conception préétabli, ce projet sera divisé en 4 parties contenant chacun leur lot de tâches à effectuer, soit :

1. Développement des mécaniques du jeu
2. Créations des éléments artistiques
3. Mise en place des serveurs
4. Mise en place du service de paiement en ligne

Les prochaines sous-sections visent à clarifier ces parties.

### Développement des mécaniques du jeu

Les mécaniques de jeu à développer sont de sauvegarder la progression du joueur, la réaction des différentes « tours » selon leur type (par exemple, une tour qui fait beaucoup de dommage et a un petit rayon de détection), les ennemis et leur « statistiques » différentes (par exemple, un ennemi pourrait être rapide, mais pas résistant, tandis qu’un autre ennemi pourrait être lent, mais très robuste.), le système d’amélioration de tours, leur placement, le chemin que les ennemis vont prendre, les vies déterminant le nombre de niveaux que l’utilisateur peut jouer avant d’attendre et l’adaptation de l’intelligence artificielle en fonction de la progression du joueur, la progression du niveau (représenté par le soleil qui se couche), les améliorations de jeu (ex. : 0.5% d’or de gagné par ennemis tués) (grandes chances qu’ils se fassent modifier)

### Créations des éléments artistiques

Les ennemis auront l’air différents dépendant du thème du niveau, par exemple, si le milieu est plutôt enneigé, un ours (ceci est un exemple. Il est possible que l’ours ne fasse partie des ennemis du jeu.) va avoir la peau plus blanche que brune.

En ce qui concerne les thèmes, déjà, le jeu va avoir un style « Pixel Art ». Plus le niveau avance, plus le village où se situent les survivants se dégrade au fur et à mesure que l’invasion prend le dessus. Le style des tours et du village sera plutôt de style Steampunk.

### Mise en place des serveurs

Aucun serveur pour le jeu sera mis en place. Le seul serveur mis en place sera celui

### Mise en place du service de paiement en ligne

Le « Play Store » et le « App Store » permettent tous deux d’effectuer des micro transactions via leur système. La première étape

## Modélisation des ressources utilisées

Le projet ne nécessite que 4 programmeurs et 4 animateurs, dont un qui va s’occuper de l’interface graphique, afin de pouvoir se réaliser, et un artiste qui va créer l’icône du jeu et de notre entreprise. Des licences de Visual Studio, Unity et Photoshop devront être achetées pour tous ces intervenants en fonction de leur champ d’expertise.

Les programmeurs seront tarifés à l’utilisation et disponibles seulement du 2 octobre 2017 au 2 janvier 2019 à raison de 35 heures par semaine, avec possibilité de faire 40 heures sans heures supplémentaires.

Pour les artistes, commençons par les animateurs. Ceux-ci, du 2 octobre 2017 au 2 juillet 2018, vont travailler à raison de 35 heures par semaine, avec possibilité de faire 40 heures sans heures supplémentaires, sauf un des quatre animateurs, celui-ci restant un autre 8 mois de plus pour s’assurer que la programmation ne ruine pas ce que les artistes ont fait. Pour la musique, l’artiste travaillera en raison de 4 à 5 chansons. Pour l’artiste faisant le logo, celui-ci prend le temps qu’il faut en raison d’un contrat.

## Réseau de tâches

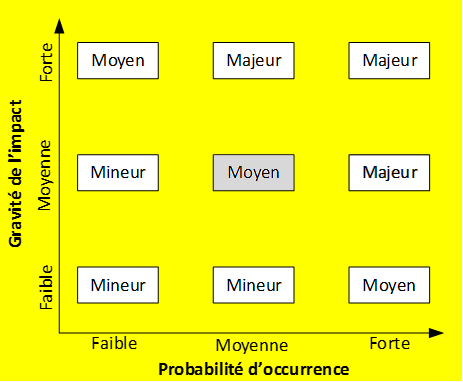
Voici le réseau des tâches de ce projet. Les dates et les durées sont comptés en jours à partir de la date de début du projet.



## Criticité, risques du projet

Nous remarquons que le projet, tel que présenté avec le réseau de tâches précédent, n’a que peu de marges de manœuvre. Par exemple, la tâche « Créer les armes du joueur » n’a qu’une marge de deux jours. Ceci est un élément très problématique puisque …

La probabilité que le projet ne termine pas avant le 1er septembre 2017 est plutôt forte, car le jeu en question sera le premier jeu de l’entreprise, donc ce sera la première fois que l’entreprise va expérimenter avec la programmation mobile (sans compter l’expérience possible qu’ils pourraient avoir accumulé durant leurs années de collège), rendant la tâche un peu plus longue, mais plus ou moins complexe.



Le niveau de criticité de ce risque est « Moyen ». Les seuls plans pour régler cette solution sans grosses répercussions serait de soit engager des programmeurs ayant de l’expérience en application mobile ou de faire en sorte que les programmeurs se pratiquent en programmant d’autres jeux mobiles. Après tout, comme le proverbe anglais le dit : *"Practice makes perfect".*

# Analyse et choix de la planification définitive

## Calendrier définitif



## Coût de réalisation du projet

En prenant en compte la planification déterminée à la section précédente, nous avons donc :

**Coûts directs**

Après des discussions entre employeurs en ce qui concerne les coûts indirects, nous en somme arrivé à ces chiffres :

* Cout d’une nouvelle licence de Visual Studio 2015 : 500 $, pour un total de 2000 $
* Coût de la location d’un local de travail avec bureaux : 1400% / mois pour un total de 22 400
* Coût de l’Internet : 140$ / mois pour un total de 2 240$
* Coût de liscence de Unity : 125$/personne pour un total de 500$

Les coûts indirects sont donc :

Le coût total du projet est donc de :

# Plan d’exécution

## Dates des points de contrôles

En plus des réunions quotidiennes reliées à la méthodologie Agile Scrum, un point de contrôle officiel sera effectué chaque mois. Nous aurons donc un point de contrôle aux dates suivantes :

* 2 novembre 2017
* 2 décembre 2017
* 2 janvier 2018
* 2 février 2018
* 2 mars 2018
* 2 avril 2018
* 2 mai 2018
* 2 juin 2018
* 2 juillet 2018
* 2 août 2018
* 2 septembre 2018
* 2 octobre 2018
* 2 novembre 2018
* 2 décembre 2018

Le dernier point de contrôle, soit le 2 décembre2019, sera aussi la date où un récapitulatif du déroulement du projet sera effectué. Advenant une fin prématurée, ce point de contrôle spécial pourrait être mis une semaine à l’avance, si possible. Dans le cas d’une fin après le 2 janvier 2019, des points de contrôle supplémentaires pourraient être placés sur le calendrier.

## Tâches à surveiller

Le type de tâche la plus problématique est la tâche …. En effet, cette tâche risque d’être beaucoup plus longue si …. La durée finale de cette tâche sera aussi un bon indicateur de la durée totale du projet. En d’autres termes, la durée des autres tâches risque fort d’être proportionnelle à la durée de cette toute première tâche.

## Structure organisationnelle

L’équipe de développement sera composée de trois programmeurs et d’un programmeur « élite ». Celui-ci aura aussi le titre de « General of the Army ». En gros, le programmeur élite, durant les « Gamebang Sessions », sera celui qui va prendre note du progrès des programmeurs et artistes, tout en étant celui qui va s’occuper la gestion des tâches de base de chacun.

## Recommandations générales

Les « Gamebang Sessions » vont se passer à chaque semaine de façon à ce que les programmeurs aient le temps de s’adapter à leurs outils (car la programmation d’applications mobiles est une nouveauté chez nous) tout en développant le jeu à temps. Celui-ci se passera la fin de semaine, soit le samedi ou le dimanche, durant une soirée où programmeurs, animateurs et artistes passeront une soirée agréable tout en discutant de leur progression. À la fin, une grande rencontre va se passer de sorte à ce que tout le monde ait un débat sur si le jeu a besoin de plus de temps ou si tout le monde est prêt à se concentrer sur les mises à jour. Aussi, à la fin d’une « Gamebang Session », le programmeur élite devra prêter son titre à un autre programmeur pour que tout le monde ait un poids égal sur ses épaules.

# Conclusion

La réussite de ce projet est fortement lié à …

# Bibliographie

**Feathers, Michael. 2011.** *Working Effectively With Legacy Code.* Upper Sadle River : Prentice Hall, Peason Education, 2011.