



IUT Lyon 1  
Département Informatique

# Projet Tutoré : Cahier des Charges

## Jeu Minetest Course Cubique

Groupe :

- BERTHOME Alyssia
- CYSIQUE Kimberley
- PELOUTIER Maxime
- SAINT-SORNY Evan
- STCHETININE Océane
- TON Hoang Chi

Tuteur :

- CHAMPIN Pierre-Antoine



IUT Lyon 1  
Département Informatique

Intitulé du sujet :

Jeu Minetest

Descriptif du sujet :

Notre sujet consiste en l'utilisation du moteur Minetest, un moteur 3D open-source inspiré du jeu-vidéo Minecraft, dans le but de créer une application 3D originale en étendant le moteur.

Version de Minetest :

Windows 5.1.0 (sortie le 12/10/2019)

Gestion de version :

GIT hébergé sur la forge de l'université <https://forge.univ-lyon1.fr/>

Licence :

Code : [LGPL 2.1](#)

Éléments artistiques : [CC-BY-SA](#)

Résumé du jeu :

**Course Cubique** est un ensemble de 2 mini-jeux basés sur le mouvement, **Square Mile** et **SplashBlock**. Square Mile est un parcours dont le but est d'arriver du début à la fin le plus vite possible et SplashBlock est un mini-jeu d'agilité où le joueur doit sauter de plateforme en plateforme sans tomber dans l'eau le plus longtemps possible. Ces deux mini-jeux seront situés dans le même monde et accessibles directement depuis un hub central.

# Table des matières

<b>I. PRESENTATION DU PROJET</b>	<b>3</b>
A. CONTEXTE DU PROJET	3
B. DESCRIPTION BREVE DU JEU	3
<b>II. ENVIRONNEMENT TECHNIQUE</b>	<b>4</b>
A. MOTEUR DU JEU	4
B. LANGAGES DE PROGRAMMATION	4
C. GESTION DE VERSION	4
D. POSSIBILITES ET LIMITATIONS DE L'API DE MODDING	5
<b>III. MAQUETTES SIMPLES</b>	<b>6</b>
A. SQUARE MILE	6
B. SPLASHBLOCK	7
<b>IV. PLANNING PREVISIONNEL</b>	<b>8</b>
A. ORGANISATION DU PROJET	8
B. LISTE DES LOTS	8
TABLEAU RECAPITULATIF	8
LOT N°1 : ENVIRONNEMENT DE BASE : 12/02/2020	8
LOT N°2 : SYSTEME DE SCORE : 18/02/2020	9
LOT N°3 : MECANQUES CENTRALES : 27/02/2020	9
LOT N°4 : MECANQUES AVANCEES : 16/03/2020	9
LOT N°5 : NIVEAUX DE DIFFICULTE : 01/04/2020	9
LOT N°6 : POLISSAGE GENERAL : 15/04/2020	10
LOT N°7 : MODES MULTIJOUEUR (OPTIONNEL) : 03/05/2020	10
LOT N°8 : CORRECTIONS FINALES : 20/05/2020	10
<b>V. REPARTITION DES ROLES</b>	<b>10</b>

## I. Présentation du projet

### A. Contexte du Projet

Notre sujet consiste à créer un jeu sur le moteur Minetest. Il s'agit d'un moteur 3D en « Voxel » inspiré du jeu Minecraft. Ce moteur étant open-source et scriptable dans le langage Lua, nous devons l'exploiter afin de créer un jeu original en y ajoutant nos propres fonctionnalités.

Nous avons choisi ce sujet pour plusieurs raisons, la première étant que certains membres du groupe considèrent le développement de jeux-vidéo dans leur carrière professionnelle. De plus, le langage Lua se différencie considérablement des langages de programmation que l'on voit en cours (C, Java, etc.). Il est donc intéressant d'apprendre à l'utiliser, et cela dans un moteur 3D.

### B. Description brève du jeu

Nous avons décidé de créer le jeu **Course Cubique**, un ensemble de deux mini-jeux, situés dans le même monde prédéfini, qui pourront être lancés depuis un même hub central où arrivera initialement le joueur. Ainsi ne sera pas nécessaire de retourner sur le menu principal de Minetest pour changer de mini-jeu.

Le premier mini-jeu, nommé **Square Mile** consiste à aller d'un point A à un point B à travers différents obstacles, bonus et malus, dans le but d'avoir le meilleur temps.

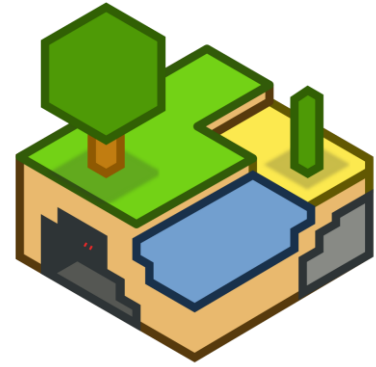
Pour le deuxième mini-jeu, nommé **SplashBlock** le joueur devra éviter de tomber en sautant sur des plateformes qui disparaissent peu de temps après être passé dessus, dans le but de tenir le plus longtemps et faire le plus de sauts possibles.

## II. Environnement technique

### A. Moteur du jeu

Le sujet impose d'utiliser Minetest. Ce moteur 3D est gratuit, open-source et inclut une API Lua pour développer des mods, c'est-à-dire un ensemble de scripts qui permettent d'ajouter ou de modifier des fonctionnalités du moteur, telles que des types de blocs, des mécaniques de jeu, des interfaces, etc.

Dans le cadre de Minetest, un « jeu » est un ensemble de mods. Nous allons nous appuyer sur **Minetest Game**, le jeu inclus dans l'installation de base du moteur, ce qui va nous permettre d'avoir des objets, mécaniques et fonctionnalités basiques. Nous pourrions également inclure et/ou s'appuyer sur des mods créés par la communauté de Minetest à notre jeu, si les licences de ces mods le permettent, pour ajouter des fonctionnalités complexes à développer mais utiles dans le cadre de notre jeu.



### B. Langages de programmation

Le moteur en lui-même a été écrit en C++, mais pour des raisons de complexité et de compatibilité avec les autres mods, nous n'allons pas le modifier.

L'API de modding de Minetest utilise le langage de scripting Lua, qui est donc le langage que nous allons utiliser pour créer notre jeu ([www.lua.org](http://www.lua.org)).



### C. Gestion de version

Nous avons choisi d'utiliser le système de gestion de version décentralisé Git pour notre travail collaboratif. Nous estimons que ses fonctionnalités telles que l'historique de versions, les branches, et un dépôt distant de référence commun sont indispensables pour mener à bien notre projet et faciliter la distribution des itérations du jeu.

Au cours du premier et deuxième semestre, nous avons appris à utiliser les commandes Git ainsi que l'utilisation de GitLab sur la Forge mise à disposition par l'Université Lyon 1. Nous allons donc notre projet sur la [Forge GitLab de l'Université](#).

Une des raisons pour lesquelles nous avons choisi GitLab est la gratuité de la création de dépôts privés, contrairement à GitHub qui est limité à 4 collaborateurs sur un dépôt privé sans version payante.



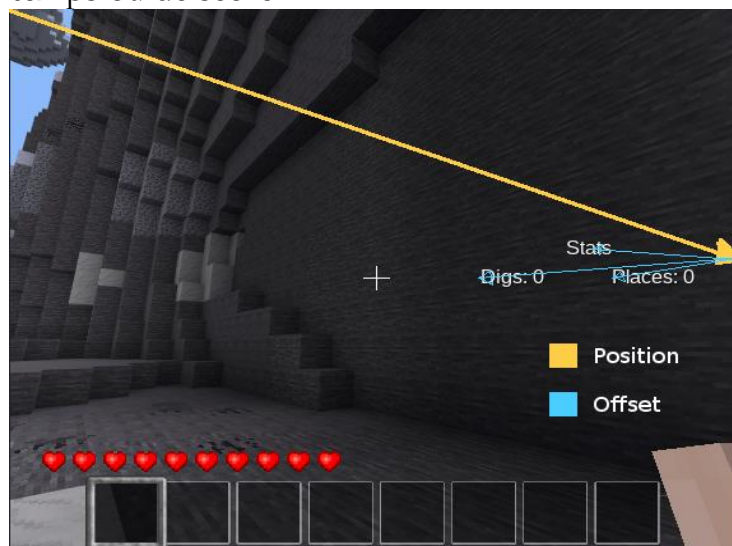
## D. Possibilités et limitations de l'API de modding

Minetest offre des possibilités très intéressantes dans le cadre de la création de nos mini-jeux grâce à son API. Voici quelques exemples extraits de [Minetest modding book](#), écrit par [rubenwardy](#) :

- Modifier la physique du joueur en surchargeant ses valeurs de base :

```
minetest.register_chatcommand("antigravity", {
    func = function(name, param)
        local player = minetest.get_player_by_name(name)
        player:set_physics_override({
            gravity = 0.1, -- set gravity to 10% of its original value
                           -- (0.1 * 9.81)
        })
    end,
})
```

- Intégrer des éléments à l'interface du joueur, par exemple pour afficher des compteurs de temps ou de score :



- Créer des nouveaux blocs avec leurs propres propriétés et apparences :

```
minetest.register_node("mymod:diamond", {
    description = "Alien Diamond",
    tiles = {"mymod_diamond.png"},
    is_ground_content = true,
    groups = {cracky=3, stone=1}
})
```

Cependant, cette API comporte des limitations qui rend difficile l'implémentation de certaines fonctionnalités, notamment :

- La création d'animaux vivants et de PNJ avec lesquels interagir ;
- La création et la conduite de véhicules.

### III. Maquettes simples

#### A. Square Mile

Le joueur commence le parcours sur le bloc violet et doit arriver sur le bloc rose pour finir sans tomber dans la zone blanche. Avec ses propriétés physiques de base, un joueur peut seulement sauter au-dessus d'un écart de 3 blocs. Le joueur obtient un bonus de vitesse de déplacement et de saut en passant sur la zone verte, ce qui lui permet de faire un saut plus grand et arriver à la dernière plateforme brune.

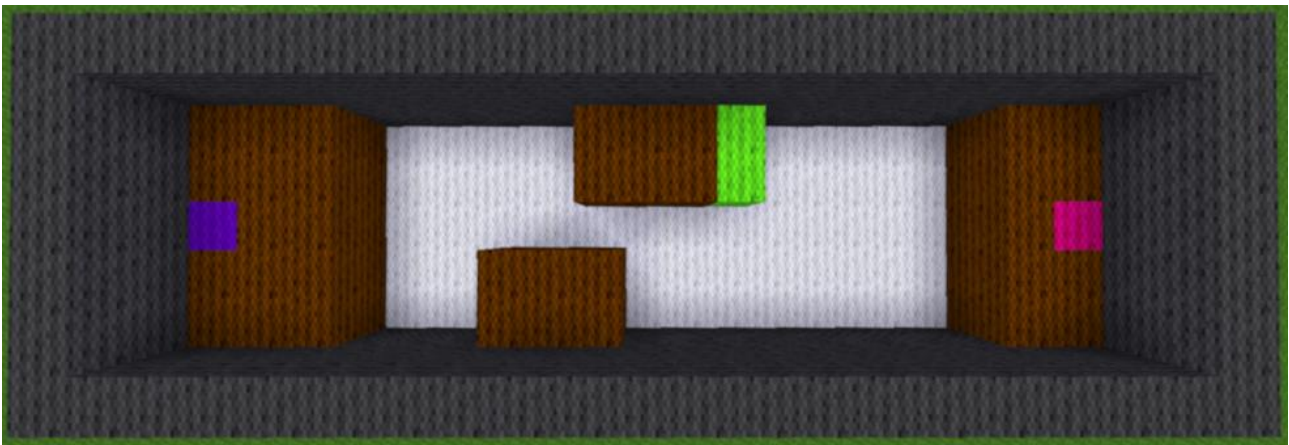


Figure A-1 : Vue du dessus

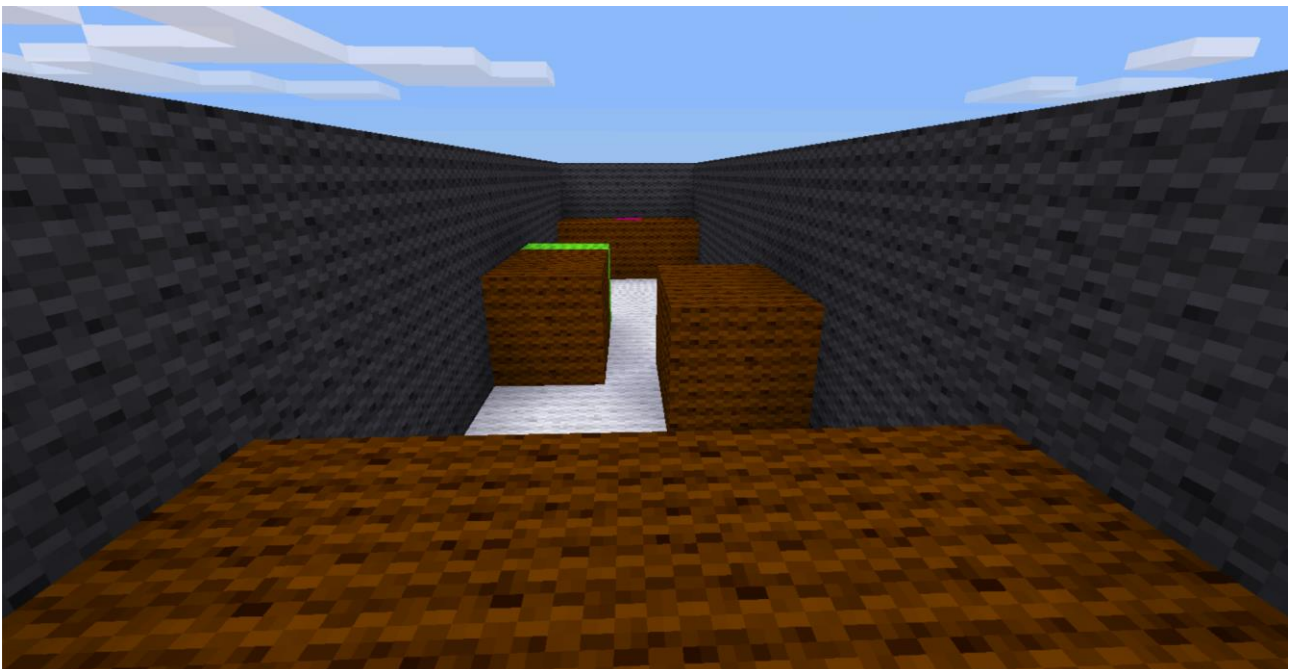


Figure A-1 : Vue subjective au début du parcours



## B. SplashBlock

Le joueur apparaît et est enfermé dans la zone circulaire. Il doit survivre tomber dans l'eau le plus longtemps possible en sautant de plateforme en plateforme, qui disparaissent quelques instants après avoir marché dessus. Les plateformes ont différents types avec des propriétés physiques associés. Ici, les plateformes rouges sont sans propriétés particulières, alors que les plateformes bleues sont glissantes.

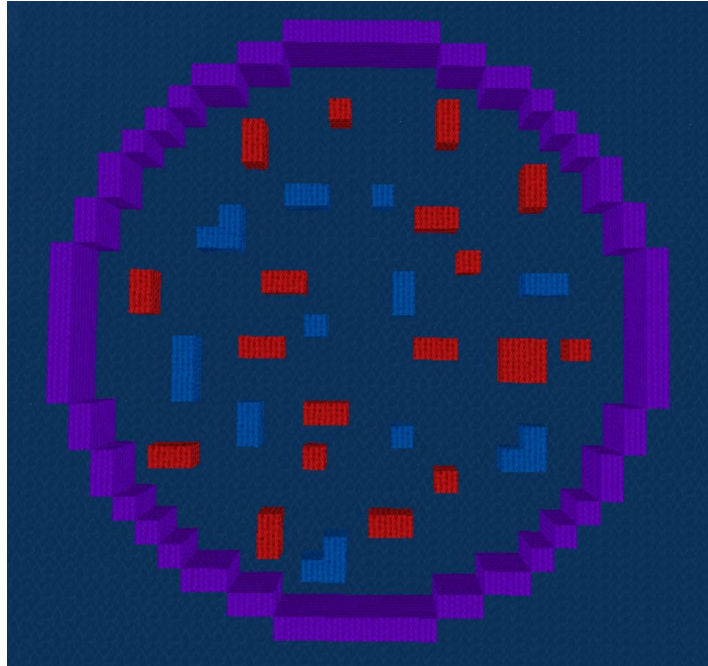


Figure B-1 : Vue du dessus

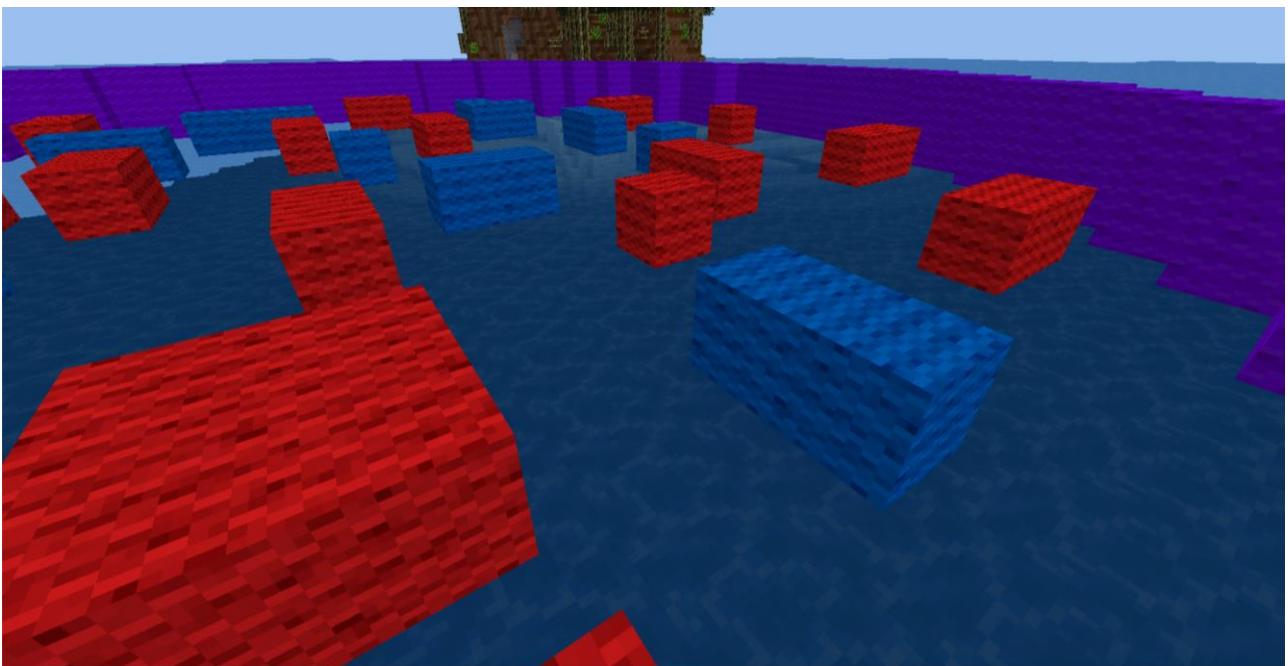


Figure B-2 : Vue subjective



## IV. Planning prévisionnel

### A. Organisation du Projet

Notre projet est divisé en lots de fonctionnalités, définies pour chaque mini-jeu et Course Cubique en général, à créer et à rendre selon des échéances durant le semestre 3, précisées ci-dessous. Notez que ces échéances sont approximatives en raison de notre manque d'expérience dans cet environnement technique.

Le projet sera réalisé selon une méthode agile. Pour chaque itération, le jeu sera livré au tuteur et testé (« playtesting ») pour récupérer des retours et éventuellement réévaluer le contenu des lots.

Le rendu pourra se faire directement via un clonage du dépôt distant sur la Forge, ou par envoi d'un dossier compressé par mail, selon la préférence du tuteur. Le dépôt/dossier contiendra les fichiers de Course Cubique (scripts, assets, etc.), mais pas l'exécutable Minetest pour des raisons de taille. Les instructions d'installation de la version Windows 5.1.0 Minetest et de Course Cubique seront comprises dans les fichiers inclus.

Voici quelques abréviations pour des termes récurrents :

- Course Cubique (jeu global) : **CC**
- Square Mile (premier mini-jeu) : **SM**
- SplashBlock (deuxième mini-jeu) : **SB**
- Documentation utilisateur : **DU**

### B. Liste des Lots

#### Tableau récapitulatif

N° du lot	Intitulé	Date de livraison estimée
1	Environnement de base	12/02/2020
2	Système de score	18/02/2020
3	Mécaniques centrales	27/02/2020
4	Mécaniques avancées	16/03/2020
5	Niveaux de difficulté	01/04/2020
6	Polissage général	15/04/2020
7	Modes multijoueur (optionnel)	03/05/2020
8	Corrections finales	20/05/2020

#### Lot n°1 : Environnement de base : 12/02/2020

<b>CC</b>	Bloc qui téléporte le joueur à une coordonnée quand il passe dessus.
<b>SM</b>	Déclencheur de début et un déclencheur de fin du mini-jeu, intégrés dans un parcours avec des obstacles simples.
<b>SB</b>	Déclencheur de début et un déclencheur de fin du mini-jeu, intégrés dans une zone de jeu avec des plateformes inactives.
<b>DU</b>	Instructions d'installation de Course Cubique dans Minetest ainsi qu'un tutoriel du jeu dans un fichier texte README.

## Lot n°2 : Système de score : 18/02/2020

<b>CC</b>	Empêchement du déclenchement d'un mini-jeu lorsqu'un autre est actif.
<b>SM</b>	Compteur de temps et de score dans l'interface du joueur. Le score dépendra du temps réalisé entre le début et la fin du parcours.
<b>SB</b>	Compteur de temps et de score dans l'interface du joueur. Le score dépendra du temps passé sans tomber et du nombre de sauts réalisés.
<b>DU</b>	Informations sur le calcul du score dans le fichier README.

## Lot n°3 : Mécaniques centrales : 27/02/2020

<b>CC</b>	Tableau des scores dans le hub central et/ou dans un menu.
<b>SM</b>	Blocs accordant des bonus (hauteur de saut, vitesse, réduction de gravité) et obstacles qui nécessitent ces bonus.
<b>SB</b>	Disparition des plateformes après un certain délai quand le joueur passe dessus, et faire réapparaître un bloc aléatoirement dans la zone.
<b>DU</b>	Explications sur les bonus et la disparition des blocs dans le fichier README.

## Lot n°4 : Mécaniques avancées : 16/03/2020

<b>CC</b>	Moyen de recommencer/quitter un mini-jeu en cours.
<b>SM</b>	Bonus de temps accordés en prenant des chemins alternatifs plus difficiles.
<b>SB</b>	Types de blocs aux propriétés physiques différentes (glissants, rebondissants, etc.) et réapparition des blocs avec un type aléatoire et à une certaine distance d'une autre plateforme.
<b>DU</b>	Tutoriels dans un menu accessible directement dans le jeu et informations sur les chemins alternatifs et les différents types de blocs.

## Lot n°5 : Niveaux de difficulté : 01/04/2020

<b>CC</b>	Choix de la difficulté par le joueur dans le hub central.
<b>SM</b>	3 niveaux de difficulté : <ul style="list-style-type: none"> <li>- Facile : Bonus plus nombreux.</li> <li>- Normal : Nombre de bonus moyen.</li> <li>- Difficile : Peu ou pas de bonus.</li> </ul>
<b>SB</b>	3 niveaux de difficulté : <ul style="list-style-type: none"> <li>- Facile : 3 vies, les plateformes disparaissent moins vite.</li> <li>- Normal : 3 vies, les plateformes disparaissent à vitesse normale.</li> <li>- Difficile : 1 ou 2 vies, les plateformes disparaissent plus vite.</li> </ul>
<b>DU</b>	Informations sur les difficultés et les modifications qu'elles apportent.

**Lot n°6 : Polissage général : 15/04/2020**

<b>CC</b>	Tableau des scores pour chaque difficulté et polissage de l'environnement visuel et sonore.
<b>SM</b>	Polissage de l'environnement visuel et sonore.
<b>SB</b>	Polissage de l'environnement visuel et sonore.
<b>DU</b>	Polissage de l'apparence et finalisation de la documentation.

**Lot n°7 : Modes multijoueur (optionnel) : 03/05/2020**

<b>CC</b>	Mode multijoueur en ligne.
<b>SM</b>	Implémentation multijoueur : Mode de course à plusieurs.
<b>SB</b>	Implémentation multijoueur : Mode « versus ».
<b>DU</b>	Informations sur l'hébergement de parties multijoueur et les modes multijoueur.

La réalisation de ce lot sera déterminée selon la difficulté du projet.

**Lot n°8 : Corrections finales : 20/05/2020**

<b>Totalité du jeu</b>	Tests détaillés et polissages finaux.
------------------------	---------------------------------------

**V. Répartition des rôles****Chef de Projet :**

**Océane S.**

Tâches : répartir les tâches au sein du groupe et valider les modifications de ce document.

**Lead Developers :**

**Maxime P. (Square Mile) et Kimberley C. (SplashBlock)**

Tâches : encadrer le développement du jeu, être le référent en matière de code et apporter de l'aide aux autres développeurs.

**Concepteurs d'environnement :**

**Alyssia B., Kimberley C. et Océane S.**

Tâches : concevoir et construire l'environnement graphique et sonore du jeu.

**Documentalistes :**

**Evan S., Hoang Chi T.**

Tâches : rédiger la documentation technique et utilisateur.