Bordeaux INP
ENSC

PROJET INFORMATIQUE INDIVIDUEL

ENSC Année 2016/2017

Rapport Final

Pierre Bégout

Sommaire

Introduction

Un de mes hobbies préféré, comme beaucoup de gens, est de jouer aux jeux vidéo. Il y a beaucoup de styles différents, multi-joueurs, de sports, les FPS . . . J'ai pu tester beaucoup de styles différents, mais le plus classique et le plus addictifs reste le jeu de plate-forme, où le personnage doit aller d'un point A à un point B sans tomber ou mourir. Cependant, ce style étant le plus ancien, il est difficile d'innover.

Je me suis donc fixé pour but de développer un jeu de plateforme en 2D assez difficile avec des niveaux très courts mais intenses, dans lequel les joueurs auront des difficultés à éviter les pièges. Je voulais aussi créer des ennemis ayant un comportement régi par une IA. De plus, je voulais réaliser un éditeur de niveaux, ainsi qu'une plateforme en ligne pour que les joueurs puissent partager leurs créations. Mais après discussion avec mon tuteur de projet, nous avons abandonné l'idée car cela demandait trop de temps et des notions techniques très avancées (notamment pour l'éditeur de niveau).

Présentation du jeu

Présentation générale

Dans ce jeu, le joueur contrôle Henry, un explorateur dont la ressemblance avec Indiana Jones n'est plus à faire. Henry est courageux mais pas très résistant. En effet, le moindre contact avec un objet un peu piquant le met KO. Le but est donc d'amener Henry au bout de ses aventures en mourant le moins possible.

Chaque niveau tient sur un seul écran. Il faut aller du début du niveau à la fin en évitant les obstacles et de tomber dans le vide.

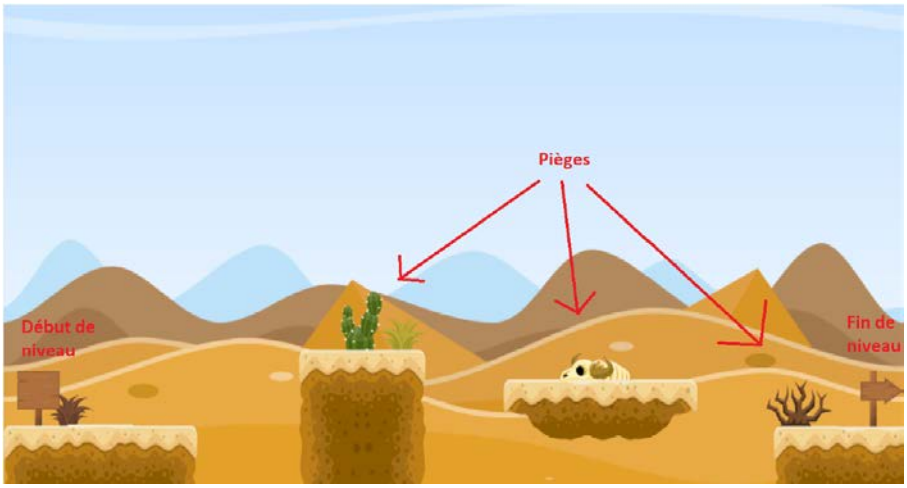


Figure 1 : Capture d'écran du premier niveau

Au cours de ses aventures, Henry va devoir traverser un désert, des montagnes ainsi qu'une usine futuriste. Le joueur va pouvoir choisir de jouer tous les niveaux, ou bien de sélectionner un environnement de départ.

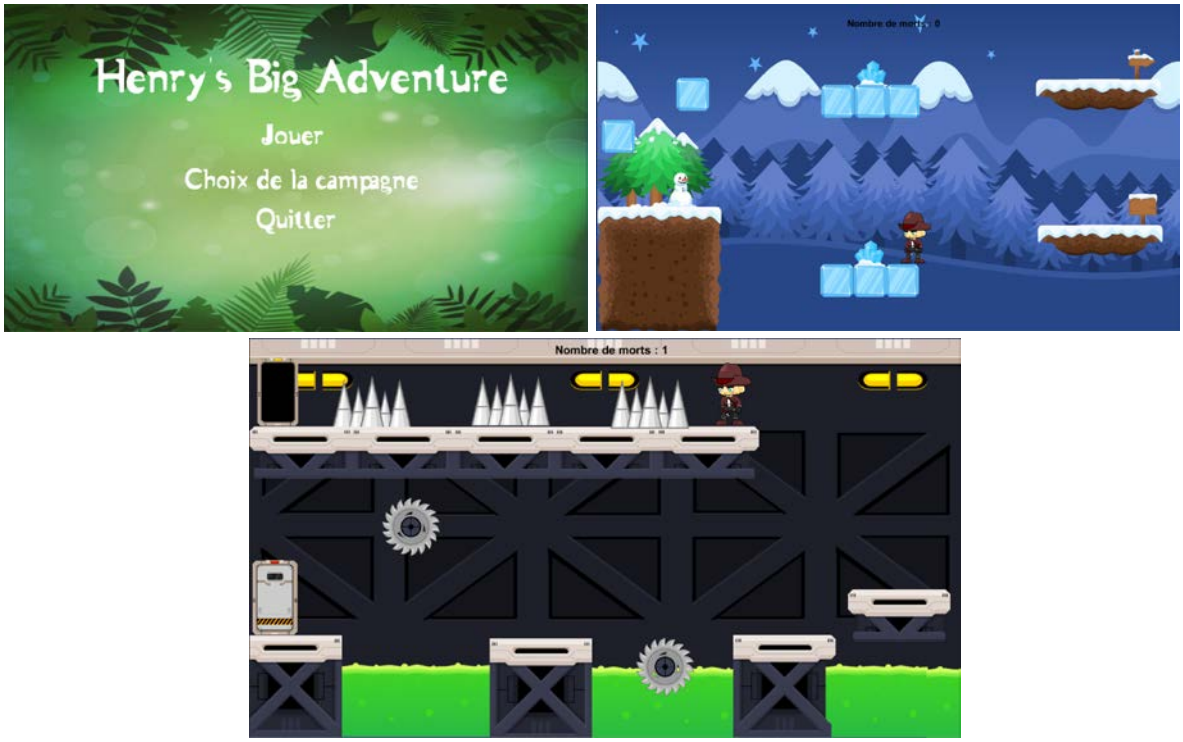


Figure 2 : Différentes captures d'écran du jeu

Comment jouer ?

Le jeu est très simple à prendre en main, mais dur à maîtriser. Pour jouer il faut cliquer sur « Jouer » dans le menu, ou sur « Choix de la campagnes » si vous ne voulez pas jouer tous les niveaux. Ensuite, il n'y a que quatre touches à retenir. Les flèches gauche et droite pour se déplacer vers la gauche ou la droite, et la flèche du haut pour sauter. A savoir qu'il y a du « air control » dans le jeu, les déplacements sont possibles lorsque le personnage est dans les airs. La dernière touche à connaître est « ECHAP », qui sert à revenir au menu principal.

Figure 3 : écran de fin du jeu

Fiche technique

Création des niveaux

Pour réaliser mes différents niveaux, j'ai téléchargé des packs d'assets gratuit qui contiennent les décors, mais aussi des éléments pouvant être utilisés comme pièges. Chaque niveau est composé de la même façon :

- Un background, qui n'est pas sur la même position z que les éléments avec lesquels le personnage interagit, ainsi que le personnage. Ce background comprend l'image de fond ainsi que les éléments décoratifs du niveau.
- Un Foreground, qui comprend les éléments avec lesquels le personnage peut interagir ainsi que le personnage lui-même.
- Une caméra qui permet la visualisation du tout.

Les éléments permettant de construire les niveaux sont tous munis d'un BoxCollider, permettant les collisions et empêchant le personnage de passer à travers le décors. Les éléments ayant un Layer « Plateforms » sont les éléments composant l'architecture des niveaux. Ce layer permet de tester si l'élément est une plateforme pour savoir si le personnage peut sauter ou se déplacer. Les éléments ayant un tag « Death » sont les pièges. Le personnage meurt lors de la collision avec un de ces éléments.

Pour finir, des rectangles invisibles placés à gauche et à droite du niveau empêchent le personnage de tomber sur les côtés du niveau. Un même rectangle est placé sous le niveau, avec un tag Death, pour tuer le personnage si celui-ci tombe dans le vide.

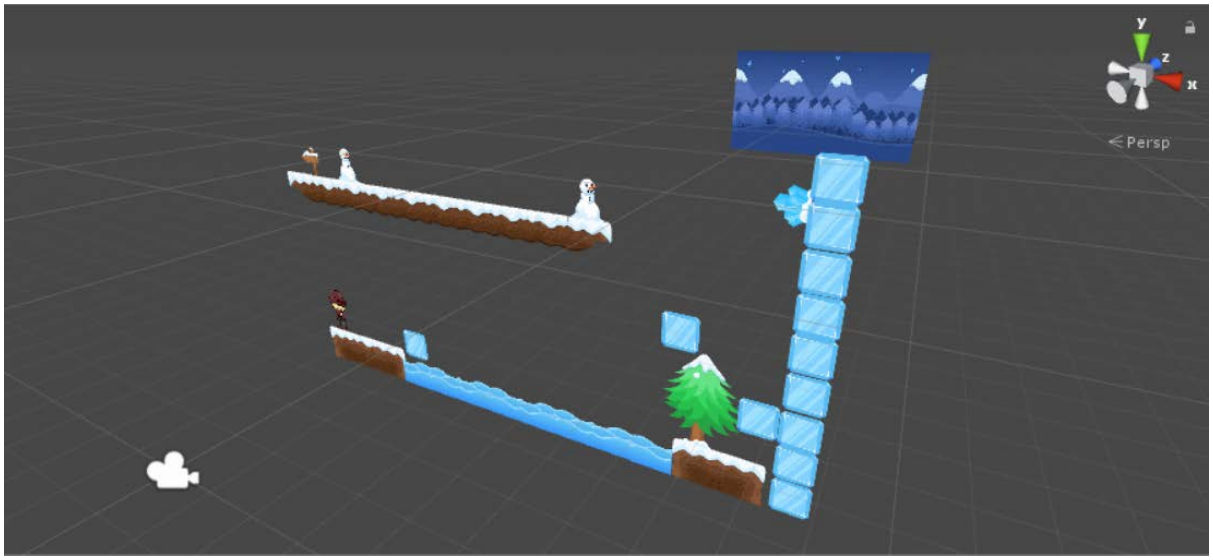


Figure 4: Représentation 3D de la construction d'un niveau

Déplacements du personnage et saut

Pour gérer les déplacements et le saut du personnage, j'ai créé un script qui permet au joueur d'avancer suivant les touches que l'on presse. Nous avons 2 valeurs, une valeur de vitesse et une valeur de poussée. Il y a aussi un vecteur de 3 dimensions, qui représente la nouvelle position du personnage. Lorsque l'on appui sur la touche gauche par exemple, nous allons soustraire notre valeur de vitesse à notre vecteur, à la dimension x. Pour la flèche de droite, on ajoute cette valeur au vecteur à la dimension x.

Pour la gestion du saut lorsque l'on appui sur la flèche du haut on va ajouter une force sous le personnage qui va le faire sauter.

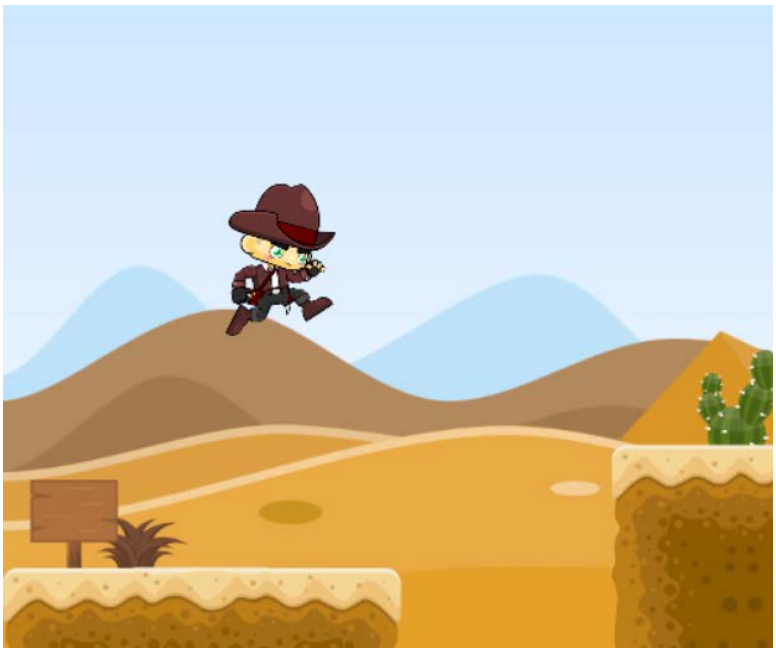
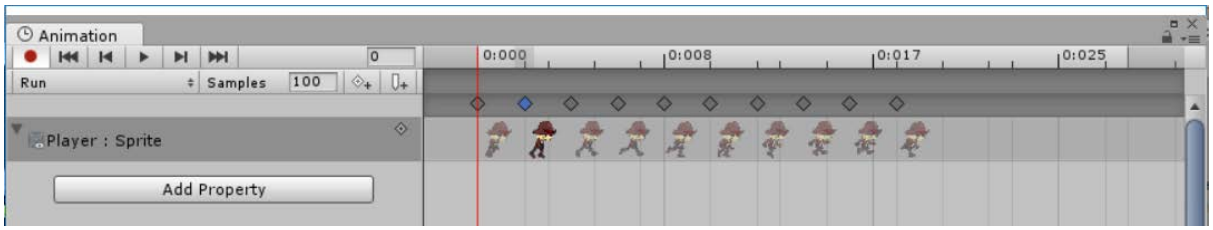


Figure 5 : Personnage en plein saut

Animations

Une fois les déplacements terminés, il a fallu insérer les animations de mon personnage. Il y a l'animation de course, celle de saut, et enfin celle de mort. Pour faire cela j'ai créé une feuille de sprites contenant toutes les images permettant de réaliser les animations. Ensuite, dans la fenêtre Animation de Unity, j'ai créé 4 états d'animations : Idle (par défaut, que l'on peut traduire par « respiration »), Run, Jump et Death. Dans chacun de ces états se trouvent à la suite les images composant l'animation.



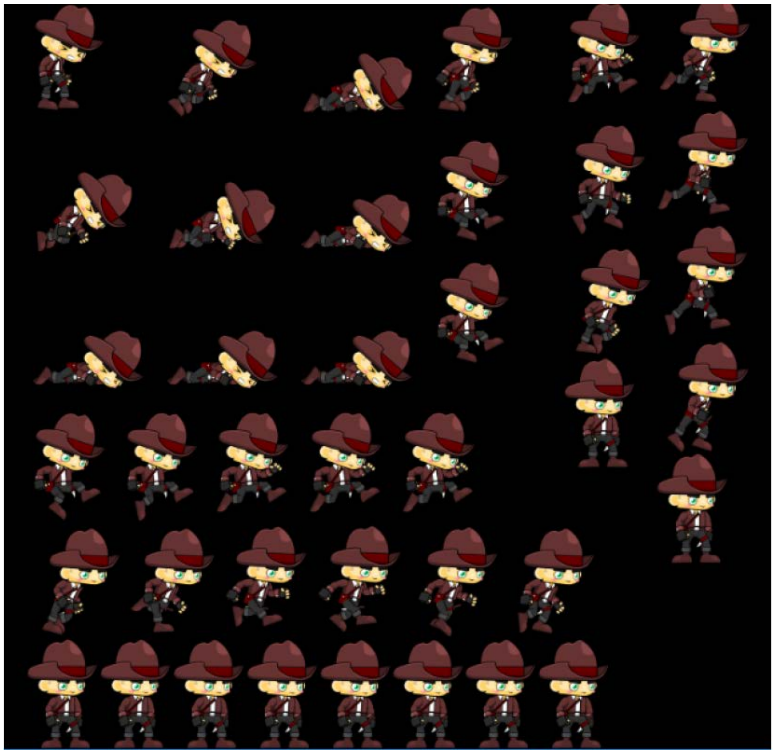


Figure 6 : Feuille de sprites

Ensuite, il a fallu établir des transitions entre ces animations, pour cela j'ai utilisé la fenêtre Animator de Unity, qui permet de définir un état de base, puis établir des transitions entre les états en fonction de la touche sur laquelle on appui. Pour cela j'ai écrit un script faisant changer une variable qui correspond à une animation. Par exemple, si on appui sur la flèche gauche ou droite, la variable passe à 2, ce qui correspond à l'animation de course.

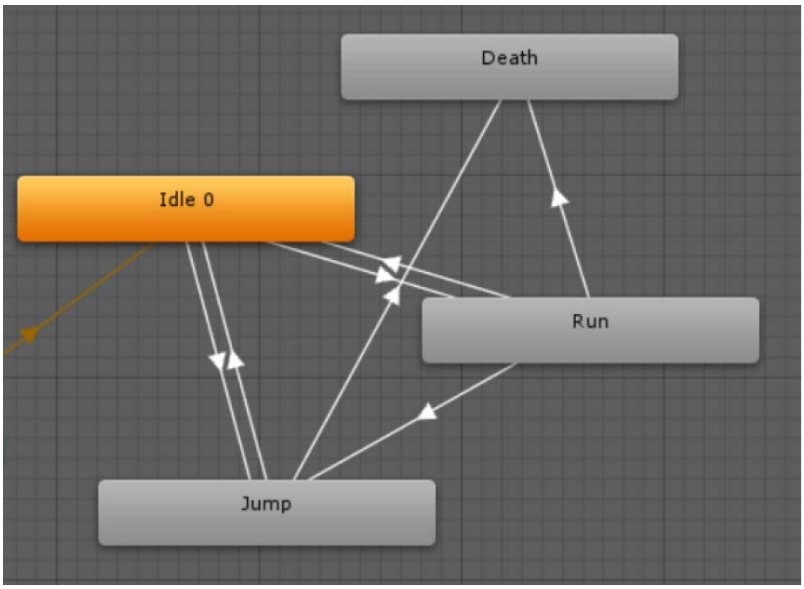


Figure 7: Graphe de gestion des animations

Gestion des collisions et fin de niveau

Pour gérer la mort du personnage, ainsi que l'affichage du nombre de morts, j'ai créé un script. Dans ce script, lors de la collision avec un GameObject dont le tag est « Death », on va jouer l'animation de mort, incrémenter un compteur, sauvegarder la valeur du compteur, puis changer l'affichage du nombre de morts. Au bout d'une seconde, on recharge le niveau.

Figure 8 : Animation de mort

Pour gérer la fin de niveau, j'ai ajouté un BoxCollider sur le panneau de fin afin de pouvoir déclencher le changement de niveau lors de la collision. Le script va juste repérer la collision et regarder quel est le niveau suivant, avant de le charger au bout d'une seconde d'attente.

Mouvement des ennemis

Pour faire bouger les ennemis de gauche à droite ou de haut en bas, j'ai créé 2 scripts qui vont faire faire des aller-retours aux GameObjects auxquels le script est attaché. Ce script prend comme paramètres la vitesse, le poids, l'amplitude et la fréquence. Il faut ensuite ajuster ces valeurs pour chaque objet pour obtenir le mouvement voulu à la vitesse voulu.

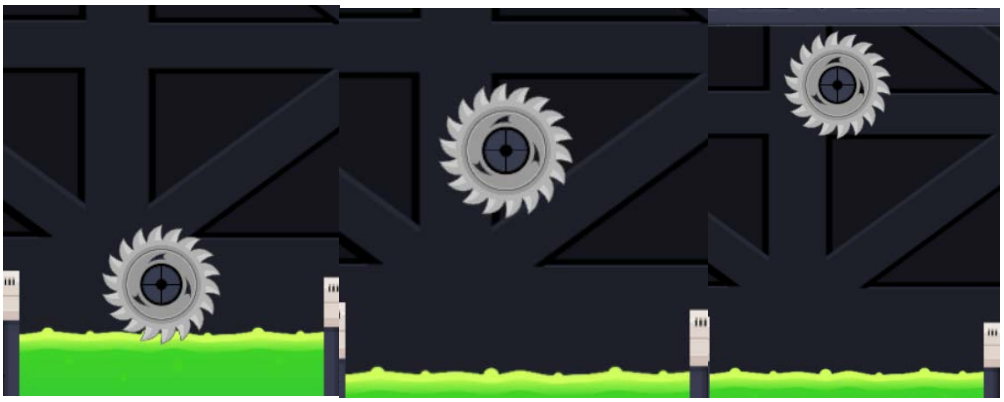


Figure 9: Mouvement des ennemis

Menus & écran de fin

Dans les menus et l'écran de fin, les éléments sont des GUItext, ce qui permet d'interagir dessus dans des scripts. Il y a un premier script qui gère le changement de couleur et le grossissement du texte lors du survol de la souris. Le deuxième script gère les actions au clique de la souris. Par exemple, cliquer sur « Jouer » entraîne le chargement du premier niveau.

L'écran de fin à un seul bouton, qui permet de revenir au menu principal. Le script va intervenir sur le GUI qui permet d'afficher le nombre de morts. Le script va récupérer la sauvegarde du nombre de morts et va mettre à jour le texte.

Figure 10 : Menu principal

Gestion de projet

J’ai commencé à travailler sur le projet dès que j’ai trouvé mon projet. J’ai d’abord commencé par suivre un tutoriel pour prendre en main Unity, me familiariser avec la création d’environnement graphique, la physique ainsi que les collisions. Ce tutoriel permettait de réaliser un jeu de type Shoot’em up, où le joueur contrôle un petit vaisseau qui peut tirer des missiles sur des ennemis qui bougent et qui tirent eux aussi. Ce tutoriel m’a permis de mieux comprendre le fonctionnement des game objects dans les scripts, ainsi que les interactions qu’il peut y avoir entre ces game objects. Cela m’a aussi permis de comprendre l’utilisation des sprites et des différents assets.

J’ai ensuite réalisé le premier niveau de mon jeu, pour ensuite créer les scripts de déplacement, les animations.

L’accumulation de projets a fait que j’ai un peu délaissé ce projet, pour m’y remettre et finir les niveaux, les menus et les finitions une semaine et demi avant le rendu. J’ai pu me consacrer ensuite aux rendus à faire en anglais.

J’ai fait plusieurs réunions avec mon tuteur Mr Salotti, pour faire le point sur mon avancement, et pour me débloquer sur quelques scripts sur lesquels j’avais des soucis.

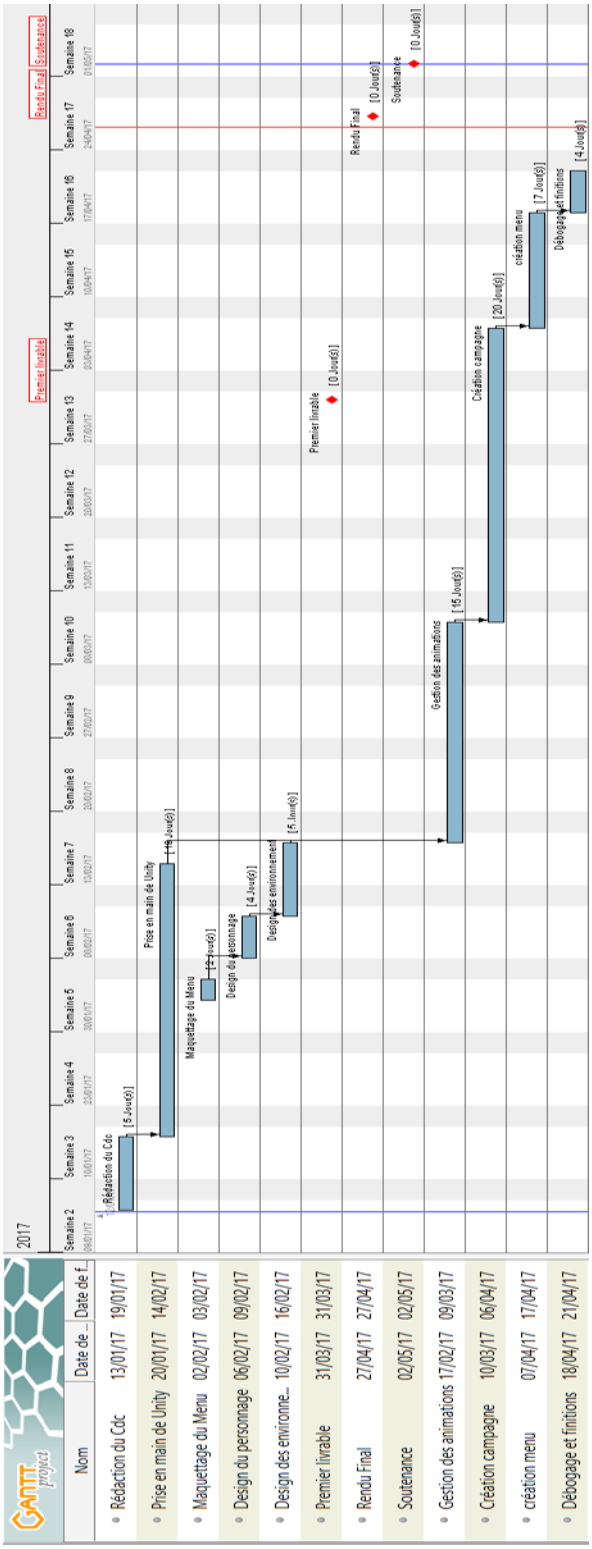
Conclusion

Ce projet m’a permis de découvrir un nouveau logiciel, de le prendre en main, et de découvrir une nouvelle approche dans le développement. Je ne connaissais rien du tout dans la gestion des scenes, des assets, prefabs et scripts et toute la réflexion en amont pour faire fonctionner tout cela ensemble.

J’ai aussi dû faire face à des problèmes de gestion de projet et d’attentes un peu trop grandes. C’est là que l’on se rend compte de l’importance d’avoir un tuteur à qui en parler.

Les futures améliorations que j’aimerais apporter à ce projet seraient d’ajouter des sons, de créer des ennemis plus diversifiés, et enfin de créer plus de niveaux.

Annexes


GANTT project
2017
Premier livrable
Rendu Final
Soutenance
Nom
Date de ...
Date de f...
Rédaction du Cdc 13/01/17 19/01/17
Prise en main de Unity 20/01/17 14/02/17
Maquettage du Menu 02/02/17 03/02/17
Design du personnage 06/02/17 09/02/17
Design des environne... 10/02/17 16/02/17
Premier livrable 31/03/17 31/03/17
Rendu Final 27/04/17 27/04/17
Soutenance 02/05/17 02/05/17
Gestion des animations 17/02/17 09/03/17
Création campagne 10/03/17 06/04/17
création menu 07/04/17 17/04/17
Débogage et finitions 18/04/17 21/04/17
Semaine 2
Semaine 3
Semaine 4
Semaine 5
Semaine 6
Semaine 7
Semaine 8
Semaine 9
Semaine 10
Semaine 11
Semaine 12
Semaine 13
Semaine 14
Semaine 15
Semaine 16
Semaine 17
Semaine 18
[5 Jour(s)]
[15 Jour(s)]
[3 Jour(s)]
[4 Jour(s)]
[5 Jour(s)]
[0 Jour(s)]
[145 Jour(s)]
[20 Jour(s)]
[17 Jour(s)]
[4 Jour(s)]
[0 Jour(s)]
[0 Jour(s)]