

RAPPORT DE SOUTENANCE 3

Juin 2022

Alexandre Agostinho, Sébastien Combe, Nathan Fontaine, Dimitri Brancourt

Avant-propos

Bonjour cher lecteur, ce rapport de soutenance vous permettra d'avoir une meilleure vision de l'avancement de notre projet

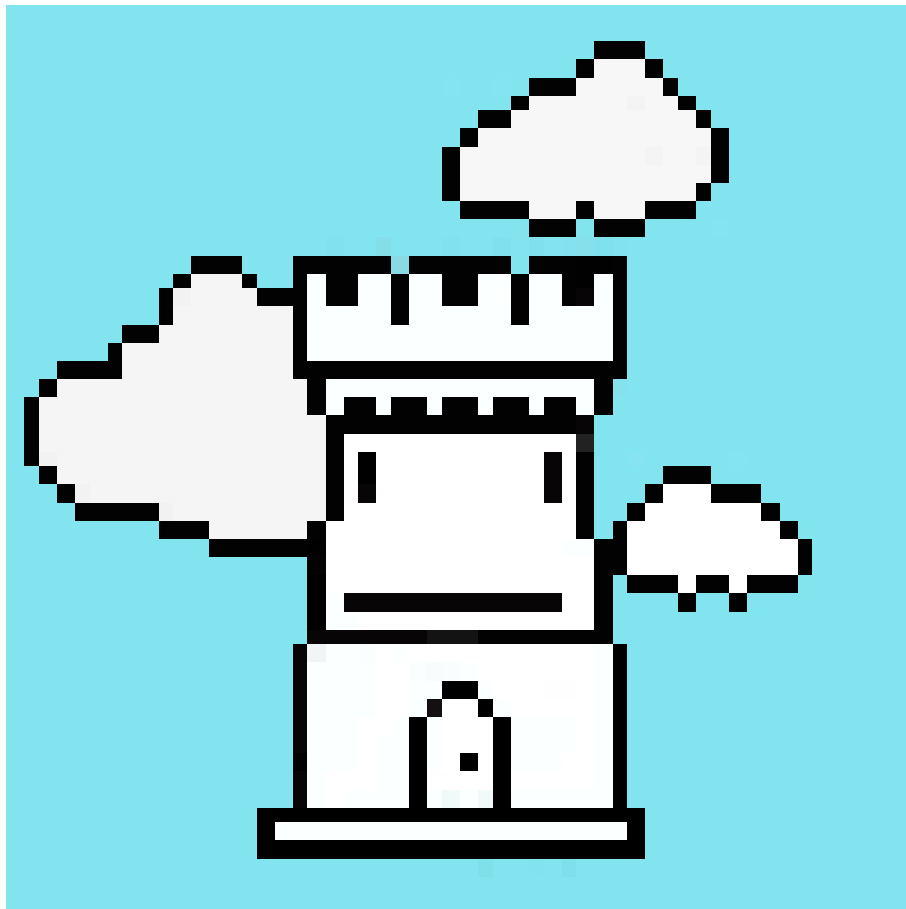


FIGURE 1 – Logo du jeu

Table des matières

1	Introduction	5
1.1	Rappel	5
1.2	FRELU	5
1.3	Tableau de l'estimation de l'avancement du Projet . . .	5
1.4	Comment se présente ce rapport	6
2	Sébastien	6
2.1	Soutenance 1	6
2.1.1	Prise en main des outils de travaux	6
2.1.2	Création des tilemaps	6
2.1.3	Cycle jour/nuit	7
2.1.4	Petite conclusion	8
2.2	Soutenance 2	8
2.2.1	Cycle jour/nuit	8
2.2.2	Gestion des lumières	9
2.2.3	Un peu de pixel art	9
2.2.4	Création du premier niveau	9
2.2.5	Petite conclusion	10
2.3	Soutenance finale	10
2.3.1	Création des nouvelles tilemaps	10
2.3.2	Histoire	11
2.3.3	Création des trois autres niveaux	11
2.3.4	Réglage de petits problèmes	12
2.3.5	Petite conclusion	12
2.4	Conclusion	13
3	Dimitri	14
3.1	Rappel Soutenance 1	14
3.1.1	Découverte de Unity	14
3.1.2	Déplacements du personnage	14
3.1.3	Deathzone	14
3.1.4	Santé	15
3.1.5	Checkpoints	15
3.1.6	portails	16
3.1.7	coffres	16
3.1.8	Canvas	16
3.2	Rappel Soutenance 2	17
3.2.1	Nouveaux déplacements	17
3.2.2	Inventaire	17

3.2.3	Sauvegarde	18
3.2.4	Magie	18
3.2.5	Pièges	19
3.2.6	Premier Boss	19
3.3	Derniers Ajouts	20
3.3.1	Barre de PV du boss	20
3.3.2	Le Magasin	20
3.3.3	Les Armes	20
3.3.4	Réglage de erreurs	21
3.3.5	Deuxième boss	21
3.3.6	Troisième boss	21
3.3.7	Quatrième boss	21
3.4	Conclusion	22
3.4.1	Gestion des outils	22
3.4.2	Gestion du temps	22
3.4.3	Travail de groupe	22
3.4.4	Avis	23
3.4.5	Prise en main de la programmation dans unity .	23
3.4.6	Déplacements du perso	24
3.4.7	Gestion de la barre de vie et des dégâts	24
3.4.8	gestion de la deathzone	24
3.4.9	Gestion des Checkpoints	25
3.4.10	Création de portails	25
3.4.11	Ouverture de coffres	25
3.4.12	Problèmes et difficultés	26
4	Alexandre	27
4.1	L'intelligence artificielle	27
4.1.1	Description	27
4.1.2	Implémentation	27
4.1.3	Difficultés	28
4.2	Le Multijoueur	30
4.2.1	Description	30
4.2.2	Implémentation	31
4.2.3	Difficultés	32
4.3	Site Web	34
4.3.1	Au commencement	34
4.3.2	Création	34
4.3.3	Remplissage	34
4.4	Mon ressenti sur le projet	35

5	Nathan	37
5.1	Soutenance 1	38
5.1.1	Menus d’affichages	38
5.1.2	Personnages	38
5.1.3	Petite Conclusion	39
5.2	Soutenance 2	39
5.2.1	Launcher	39
5.2.2	Menu Options	39
5.2.3	Personnages	39
5.2.4	Travail sur les maps	40
5.2.5	Petite Conclusion	40
5.3	Soutenance 3	40
5.3.1	Menu Options	40
5.3.2	Personnages	40
5.3.3	Jaquette DVD et Notice du Jeu	40
5.3.4	Installateur	41
5.3.5	Fonds d’écrans et visuels	41
5.3.6	Musique et sons	41
5.4	Ressentie	42
6	Conclusion	43
7	Annexes	43
7.1	Impression DVD	43
7.2	Personnages	43

1 Introduction

1.1 Rappel

Notre projet est un jeu plateforme 2D de style pixelart développé à l'aide de Unity.

1.2 FRELU

Notre équipe s'est répartie les tâches de la manière suivante :

Tâches	Alex.A	Dimi.B	Séba.C	Nath.F
Objects Design		S	S	R
Mobs/Characters Design			S	R
Game Design			R	S
GUI Design		S		R
GUI Programing		R	S	S
Gameplay Programing	S	R	S	S
IA Programing	R	S		
Website	R		S	
Multiplayer	R	S	S	
Sound Design				R

R : responsable, S : suppléant

1.3 Tableau de l'estimation de l'avancement du Projet

Soutenance	3 réel	3 estimation
Objects Design	95%	100%
Mobs/Characters Design	100%	100%
Game Design	95%	100%
GUI Design	100%	100%
GUI Programing	100%	100%
Gameplay Programing	100%	100%
Multiplayer	90%	100%
IA Programing	100%	100%
Website	100%	100%
Sound Design	100%	100%

Nous avons surestimer nos capacités pour réaliser certains objectifs c'est pourquoi nous estimons que certains catégories n'atteignent pas 100% comme par exemple le fait que notre multijoueur soit très différents du jeu solo de base car c'était assez compliqué à mettre en place dans les conditions que nous avons eues, mais pour autant nous sommes très fier de ce que nous avons réalisé jusqu'à maintenant.

1.4 Comment se présente ce rapport

Ce rapport suit un ordre chronologique de chacun des membres un à un.

2 Sébastien

2.1 Soutenance 1

Reprenons depuis le début, notre projet est parti de rien et nous avons dû nous répartir les tâches, je suis donc devenu le level designer couplé d'un peu de création de tilemap et de gestion du cycle jour/nuit.

2.1.1 Prise en main des outils de travaux

Premièrement il a fallu prendre en main nos outils de travail et principalement Unity, élément centrale de notre projet, mais aussi de Gimp dans mon cas pour créer les visuels des tilemaps.

La principale difficulté au début à surtout été de prendre en main des outils inconnus mais aussi de créer les bases de notre projet, mais nous y reviendrons plus tard. En réalité cette appréhension d'outils n'a été qu'un problème que durant les premières semaines, une fois habitué tout était plus fluide et plus rapide et nous avons commencé à créer les bases de notre jeu.

2.1.2 Création des tilemaps

On ne peut pas créer un niveau sans tilemap, une tilemap est utilisé pour créer les niveaux depuis la nuit des temps (pour le jeu vidéo du moins), une tilemap n'est qu'une tuile composé de plusieurs éléments pour pouvoir ensuite créer nos niveaux.

Elles sont généralement utilisées dans les jeux 2D comme le nôtre nous avons donc tout naturellement utilisés ces tilemap pour construire nos niveaux.

Il faut tout d'abord savoir que je ne suis pas un artiste dans l'âme, c'est donc naturellement qu'un problème est venu me barrer directement la route : c'est pas beau.

Et oui, on ne s'improvise pas graphiste en quelques semaines et ça je l'ai bien compris, mes débuts était assez catastrophique et mes premiers essais ressemblais plus à une bouillie de pixel qu'à autre chose. Mais après plusieurs essais les premiers visuels était enfin créé, pas très beau mais pas trop moches non plus, utilisable.

La création de tilemaps n'est pas un travail fondamentalement « difficile » (même si il faut un certain talent que je n'ai apparemment pas) il est plus un travail long et fastidieux. Il faut , en effet, pour chaque niveau créer les visuels associés et cela prend un temps monstre. Au final, cette soutenance était plus orientée sur le graphisme, j'ai plus utilisé Gimp que tout autre logiciel, mais cela m'a quand même aidé pour la suite, ces semaines longues mont permis d'être plus à l'aise par la suite pour créer de nouvelles tilemap, cette avec cette soutenance que j'ai le plus appris sur le graphisme et les tilemaps.

2.1.3 Cycle jour/nuit

Mais je n'ai pas fait que du graphisme durant dette soutenance j'ai aussi commencé à coder le cycle jour/nuit. J'ai donc commencé à m'intéresser à ce fameux cycle jour/nuit, ne connaissant pas le sujet j'ai dû me renseigner un minimum.

Et c'est ainsi, après un peu de recherche, que j'ai trouvé comment faire le cycle jour/nuit, et cela repose sur un principe très simple : un global volume et un peu de post- processing. Cela peut sembler complexe au premier abord et c'est ce que j'ai aussi pensé, mais en m'y intéressant un peu plus j'ai rapidement compris que coder le cycle jour/nuit n'allait pas être si compliqué.

Cependant, mon cycle jour/nuit n'était pas 100 % opérationnel lors de

cette soutenance, la nuit était trop sombre, et je n'avais encore pris pleinement en main ces nouvelles notions. Au final ce cycle jour/nuit n'a pas énormément avancé il est resté sur des bases que je n'avais pas encore assimilé.

2.1.4 Petite conclusion

la première soutenance était surtout là pour poser les bases. Des bases qui nous seront ensuite bénéfique dans le future de notre projet. Cette soutenance était donc centré sur les débuts (peu être un peu bancale) et sur le commencement d'un réel travail d'équipe. Cela nous à donc permis d'assimiler les bases.

2.2 Soutenance 2

Cette deuxième soutenance était plus centré sur la gestion des lumières et sur le cycle jour/nuit.

2.2.1 Cycle jour/nuit

Je l'ai déjà dit plus tôt mais mon cycle jour/nuit était déjà « opérationnel » lors de la première soutenance il fallait donc l'améliorer.

Tout d'abord il fallait régler le problème de cette nuit trop sombre, et ça c'est assez simple il suffisait juste de changer une ligne du programme pour que la nuit soit « moins forte » et ainsi avoir une nuit agréable. Mais le principale problème venais de la sauvegarde, en effet c'est Dimitri qui a codé la sauvegarde et là se posais un petit problème : lorsque l'on sauvegardais la nuit, la luminosité n'était pas sauvegardé mais le timer oui, il était donc possible de sauvegarder la nuit de recharger la partie et de se retrouver en plein jour.

Il a donc fallu qu'avec l'aide de Dimitri, on fouille dans ce qu'avais codé Dimitri et dans mon code pour faire marcher ce cycle jour/nuit avec la sauvegarde, et après un peu de recherche nous avons réussi à le faire marcher, et maintenant le cycle jour/nuit marche parfaitement

avec la sauvegarde. Avec ça j'ai aussi implémenter le fait que les lumières s'éteigne la nuit et s'allume le jour, ainsi maintenant à une certaine heure la nuit tombe correctement et les lumières s'éteigne de même et inversement le matin.

2.2.2 Gestion des lumières

Étant donné que je suis responsable du cycle jour/nuit il était naturel que je m'occupe des lumières, et je ne m'attendais pas à ce que ce soit si complexe.

Dans n'importe quel jeu la gestion des lumières est un calvaire et notre jeu n'y fait pas exception, gérer les lumières est en effet assez complexe cependant Unity aide beaucoup sur la gestion des lumières et la rend un peu moins complexe.

Cependant gérer les lumières engendre quelques erreurs, comme toutes les textures qui deviennent noires par exemple. Outre les erreurs, il faut bien gérer le rayon, la puissance des lumières pour un bon rendu.

2.2.3 Un peu de pixel art

Nous avons maintenant de belles lumières mais nous n'avons pas de sprite à attacher à ces lumières. Il fallait donc en créer et c'est ce que j'ai, j'ai créé un sprite de torche ainsi qu'un sprite de lanterne. Le sprite de torche à ensuite été animé par Nathan.

2.2.4 Création du premier niveau

C'est maintenant que les choses sérieuses commencent. Comme dit plus tôt la première soutenance était centrée sur les bases, les bases maintenant entre nos mains il fallait commencer à créer les niveaux et ça c'est mon travail principal.

Étant level designer c'est moi qui m'occupe de la création des niveaux et de l'agencement du niveau en général.

Mais pour un premier niveau je ne voulais pas créer à 100niveau est un grotte et faire un niveau labyrinthique comme celui d'une grotte à la main est très long et n'est en réalité qu'une perte de temps. Ainsi, avec l'aide de Nathan et de sa génération procédurale qui m'a servis de calque, le premier niveau est né.

La génération procédurale n'est cependant pas parfaite et j'ai du changer quelques petits détails pour rendre le niveau plus agréable. Après avoir créer le corps du niveau il fallait placer tout les monstres et tout les pièges dans le niveau. Ainsi, le premier niveau était terminé, il n'est peut être pas parfait mais il reste un premier niveau fonctionnel.

2.2.5 Petite conclusion

Cette deuxième soutenance était donc plus centré sur le travail d'équipe avec le cycle jour/nuite ou avec la création du premier niveau et l'utilisation de la génération procédurale. Cette soutenance a surtout servis à approfondir se que j'avais déjà fait et de rendre le tout plus joli et plus performant, mais aussi de m'essayer à la création de niveau.

2.3 Soutenance finale

Cette soutenance est la dernière c'est maintenant que mon travail prend tout son sens, c'est moi qui doit m'occuper de la création des trois autres niveaux et cette fois sans génération procédurale.

2.3.1 Création des nouvelles tilemaps

Il faut savoir que les premières maps créées ne corresponde pas avec le thème des autres niveaux, il a donc fallu créer de nouvelles tilemaps pour les nouveaux niveaux.

Des pierres enneigés, des nuages ou encore des maisons, malgré que j'ai déjà créé des tilemaps cela reste toujours un peu tendu de faire du pixel art (problèmes de proportions avec les maisons et les plates- formes

nuages qui ne sont pas très jolies).

Je n'ai pas eu plus de difficultés étant donné que j'ai déjà créé des tilemaps.

2.3.2 Histoire

L'histoire n'est pas énormément présente elle ne l'est qu'au travers des panneaux qui se trouve le long des niveaux.

Cependant, et assez logiquement, elle a grandement jouée sur les aspects et la construction des niveaux ou alors tout simplement dans leurs thèmes.

2.3.3 Création des trois autres niveaux

Cette fois-ci je n'ai pas utilisé la génération procédurale et tout les niveaux restant ont été créés de A à Z. Ces niveaux sont plus orientés plates-formes que le niveau 1, ils sont donc moins rempli de pièges. Avec cet aspect là, il a fallu réfléchir au placement des plates-formes pour ainsi, même avec moins de pièges, rendre les niveaux quand même challengeant.

J'ai donc du réfléchir au placement de chaque plates-forme en fonction de la longueur du saut, de la hauteur du saut avec tout ces paramètres en compte j'ai donc pu placer les plates-formes de façon à se que les niveaux sois agréable à parcourir mais aussi challengeant.

Ainsi la création de ces trois niveaux était complètement différent de la création du premier niveau, étant donné que je n'ai pas utilisé la génération procédurale, mais j'ai du à la place réfléchir un peu plus sur le placement des éléments pour rendre le tout cohérent et agréable.

Les thèmes des trois niveaux sont biens différents et sont construit de cette façon :

Le deuxième niveau est la première phase de l'ascension et n'est pas constitué de nouveaux éléments de tilemaps (hormis les maisons)étant donné que ce niveau est principalement constitué de plat-formes en

pierres je n'ai pas eu à utilisé les nouvelles tilemaps. Cependant il a sans doute été le niveau le plus dur à créer étant donné que c'est le premier niveau créé sans génération procédurale.

Le deuxième niveau est la suite de l'ascension et est un niveau enneigé, c'est le premier niveau qui utilise les nouvelles tilemaps, de plus ce niveau est beaucoup plus remplie de monstres et donc moins plat-formes et moins remplie de pièges que les deux niveaux d'avant, il reste quand même assez plat-formes étant donné que ça reste une ascension.

Le quatrième et aussi dernier niveau, est un niveau littéralement dans les nuages, il est le niveau le plus challengeant en terme de plat-formes, du principalement à son thème puisque les plat-formes elles mêmes sont des nuages.

2.3.4 Réglage de petits problèmes

Cette soutenance étant la finale, nous avons déjà toute les bases et cela étant donc plus orienté sur le réglages de certains problèmes.

Des détails certes, mais qui pourrait nuire à l'expérience de notre jeu. Ainsi les légers problèmes était de légers erreurs sur les tilemaps, certains placement de pièges ou de placement dans les niveaux, de légers erreurs de lumières.

Régler ces petits problèmes à principalement été un travail d'équipe, en temps que level designer je faisais tester mes niveaux fraîchement créer aux autres membres du groupe pour qu'ils me fassent un retour sur les choses qui n'allaient pas : des textures trop répétitives, un problèmes de plat-formes, etc...

2.3.5 Petite conclusion

Cette dernière soutenance était donc centrée sur la création des trois autres niveaux, un travail complètement différent du premier niveau mais pas non plus inconnu pour autant mais aussi de réglages de petits problèmes par-ci par-là.

Mais elle a surtout été un travail d'équipe car sans les tests de mes coéquipiers je n'aurais sans doutes jamais vu certaines erreurs.

2.4 Conclusion

Ce projet à donc été lourd d'enseignement, tout d'abord sur le travail de groupe qui est central dans ce projet, savoir se coordonnés avec les autres, se répartir les tâches, etc.. Mais ce n'est pas la seule chose que j'ai appris durant ce projet : j'ai appris à créer des tilemaps, à gérer des lumières et un cycle jour/nuit, à concevoir un niveau. Ce projet a donc été en tout point intéressant et m'a permis d'en apprendre beaucoup sur le travail de groupe mais aussi sur la création d'un jeu vidéo, qui est loin d'être quelque chose de simple.

3 Dimitri

Ma partie était centrée sur le Gameplay programming, cette partie englobait donc les actions du personnages et ses interactions avec d'autres objets ...

3.1 Rappel Soutenance 1

3.1.1 Découverte de Unity

Pour commencer le premier point qui fut abordé lors de l'initialisation du projet était bien sûr la prise en main des outils dont Unity.

Unity est un outil très puissant permettant de lier des images à des propriétés comme la physique, les collisions ou même des scripts. Pour ce faire il possède une grande quantité d'options qui rendent l'outil complexe.

Il a donc fallut dans un premier temps apprendre à gérer toutes ces options différentes, la documentation en ligne à été très utile.

3.1.2 Déplacements du personnage

Maintenant, Passons à la première interaction du jeu, le déplacement du joueur.

Ces déplacements sont donc le déplacement à gauche, à droite et le saut. Plusieurs difficultés se sont posées, dont la principale, comment gérer le mouvement dans Unity ? Pour programmer dans Unity il a fallut une connaissance de la bibliothèque de Unity, que la documentation a pu apporter.

Les autres petits problèmes étaient plus d'ordre technique, comment faire tourner le personnage si on se déplace dans le sens opposé, ou encore comment faire un seul saut et non pas à l'infini dans les airs.

Cette partie reste quand même une des plus simples et n'a pas posé tant de problèmes.

En parallèle j'ai pu découvrir une fonctionnalité de Unity qui m'a permis de faire des animations du personnage via les différents dessins simulant le mouvement.

3.1.3 Deathzone

Comme notre jeu est un platformer, il était primordial de créer une "deathzone".

Si vous n'êtes pas trop familier des jeux vidéos, une deathzone est une zone située dans le vide qui a pour but de replacer le personnage s'il

rentre dedans et ainsi d'éviter qu'il tombe à l'infini.

Lors de la réalisation de la deathzone j'ai rencontré mon premier bug, le personnage venait parfois se coincer dans le terrain. ceci était du à une collision du personnage et de la map lors de remplacement du personnage. Pour rendre le déplacement plus joli j'ai aussi fait un écran de fondu noir comme transition du remplacement du joueur.

Mis à part le problème de collision cette partie était aussi simple et n'a pas posé de problèmes ultérieurs.

3.1.4 Santé

Un autre point important était bien sur la santé du joueur.

ici la difficulté était juste de gérer les différents dessins des cœurs pour donner l'impression que la vie descend, bien sûr il y avait différent cas à gérer, comme par exemple si la vie était négative ou l'invincibilité.

J'ai pu donc lier des dégâts à la deathzone, on prend des dégâts de chute dans le vide.

J'ai ajouté un système d'armure, cela remplace les cœurs quand la valeur d'armures est supérieures à zéro, cependant une méthode permet de viser directement les points de vie du joueur en ignorant son armure.

Et enfin la dernière partie liée à la santé est bien sur le gameover, la défaite du joueur si les points de vie restants sont de zéro. Dans ce cas là le menu gameover est appelé, mais c'est la partie de Nathan. Dans cette partie il n'y a pas eu trop de problèmes non plus, mais il fallait savoir gérer les priorités d'affichage.

3.1.5 Checkpoints

Un autre élément important de jeu vidéo est le point de sauvegarde.

Ce point est très utile car il permet d'éviter de recommencer de zéro lorsqu'on meurt dans le jeu. Dans notre jeu il aura plusieurs utilités la première utilité à ce moment sera juste de réapparaître ici lorsqu'on tombe dans le vide. ici on va juste changer le point de remplacement de la deathzone sur un point de sauvegarde lorsqu'il est activé.

La touche d'interaction général est la touche E, avec un petit texte d'interaction lorsqu'on est à proximité de l'objet.

Ici donc la principale difficulté était de trouver un moyen de bouger le point de respawn et de créer la possibilité d'interaction avec le point de sauvegarde.

Cette partie était un peu plus difficile que les autres, mais n'a pas posé encore une fois trop de difficultés.

3.1.6 portails

Parlons maintenant des portails, cette partie était la plus difficile à ce moment du projet.

Dans le jeu il y a deux types de portails : le portail dans le même niveau et le portail permettant justement de changer de niveau.

Le portail permettant de changer de niveau était sûrement le plus simple des deux. il suffisait simplement d'utiliser des méthode préexistantes de Unity lorsque le personnage est détecté sur le portail, une méthode permettait de changer de scène et l'autre permettait de garder des éléments d'une scène à l'autre.

à partir du moment que j'avais connaissance de l'existence de ces deux méthodes il n'y avait plus aucune difficulté pour ce type de portail.

Pour le deuxième portail toute la difficulté reposait sur une erreur en particulier, le fait que le mouvement du joueur impacte sa position et s'interpose avec sa téléportation.

Cela empêchait la téléportation mais résoudre ce problème était relativement simple, il suffisait de désactiver le script de déplacement.

Une autre difficulté venait aussi s'interposer, celle de trouver le bon portail, grâce à la documentation ce fut chose relativement aisée.

Bien sûr je me suis servi de l'écran de fondu noir réalisé précédemment comme transition de téléportation.

3.1.7 coffres

Enfin, pour la dernière partie, parlons des coffres.

Les coffres sont assez simples, si le personnage est dans la zone du coffre et appuie sur le bouton d'interaction il va activer l'animation d'ouverture du coffre, gagner un nombre de pièces défini et enfin annuler la possibilité d'interagir encore avec le coffre.

Il fallait juste bien penser à rendre l'interaction impossible après l'ouverture du coffre.

L'implémentation du coffre a aussi vu le début de l'inventaire, possédant juste une variable pour "sauvegarder" les pièces obtenues.

3.1.8 Canvas

Pour conclure la partie de la première soutenance je vais parler rapidement de l'affichage sur l'écran.

Je me suis occupé de l'affichage sur l'écran des points de vie mais aussi de textes d'interactions et des pièces. Tous ces affichages sont gérés sur un composant le canvas.

3.2 Rappel Soutenance 2

3.2.1 Nouveaux déplacements

Pour la Deuxième soutenance j'ai rajouté quelques fonctionnalités liées aux déplacements du personnage.

J'ai rajouté un système de double saut. La première difficulté était d'empêcher le saut infini et la seconde était de rendre le second saut plus joli, car la force de chute du personnage après le premier saut annulait le second saut.

En parallèle j'ai ajouté un saut, il y a donc possibilité de faire soit un grand saut soit un petit saut.

Un autre déplacement est la dash.

Si vous n'êtes pas familier avec les jeux vidéos, un dash est un déplacement horizontal rapide sur une courte durée. Pour ce mouvement la difficulté était d'instaurer un délai pour éviter de pouvoir utiliser non stop le dash.

Le dernier mouvement avancé est la montée aux échelles.

Le système est assez simple lorsque le personnage interagit avec l'échelle il peut se déplacer vers le haut ou vers le bas.

La difficulté était donc de créer une plateforme en haut de l'échelle qui disparaît lorsque le personnage monte et réapparaît lorsque le personnage est en haut.

Afin d'optimiser un peu les échelles j'ai aussi appris à me balader entre les différents objets parents et enfants dans Unity.

3.2.2 Inventaire

Passons maintenant à l'inventaire.

Cette partie a surtout fait appel à mes connaissances de programmation orientée objet, que j'avais déjà vue grâce aux tp d'ip.

L'inventaire est composée d'une variable représentant le nombre de pièces, une variable qui est une liste d'items une autre classe d'objet, une liste de armes, encore une autre classe. j'ai donc dans mon inventaire les pièces, les items et les armes. Je peux donc vérifier le nombre d'objets ou si je les possède.

J'ai ajouté les items, c'est à dire les objets consommables, il y a une potion permettant de regagner des pv, une autre de l'armure, une autre du mana, et une autre de parer le prochain coup.

Ces objets possèdent des variables permettant de répertorier leurs effets, nombre, nom ou même description.

Il y a bien sûr eu aussi un travail au niveau visuel.

Il y a une image qui change en fonction de l'item sélectionné et un texte expliquant comment utiliser l'objet et comment sélectionner un autre item.

Pour les items, la difficulté était de gérer leur nombre efficacement pour permettre les développements futurs (sauvegarde ou magasin par exemple).

3.2.3 Sauvegarde

La partie la plus difficile pour moi était la sauvegarde.

Tout d'abord j'ai choisi de sauvegarder les fichiers sous format json, j'ai donc du dans un premier temps faire des recherches sur la bibliothèque json, encore une fois la documentation à été très utile.

Bien que cela puisse paraître idiot une autre difficulté était aussi de trouver les fichiers de sauvegarde sur l'ordinateur pour les supprimer afin d'effectuer des tests.

La particularité de la sauvegarde est que certaines variables ne pourront pas être sauvegardées. J'ai donc du revoir certains scripts pour pouvoir sauvegarder efficacement les informations que je voulais.

Une autre difficulté était d'empêcher certaines erreurs avec des objets non initialisés, j'ai donc du encore une fois revoir une bonne partie de mes scripts.

Au final j'ai fait en sorte de sauvegarder la position du joueur, son inventaire, sa santé et enfin l'heure avec l'intensité de la luminosité actuelle. Une erreur avait été créée avec le cycle jour/nuit, lorsqu'on sauvegardait la nuit et que l'on chargeait la sauvegarde cela mettait la luminosité comme le jour.

3.2.4 Magie

La partie magie tournera autour de deux fonctionnalités.

Tout d'abord la magie est utilisable en échange de mana qui est une valeur qui diminue lors de l'activation d'un sort. On regagne du mana avec le temps ou grâce aux potions.

La première difficulté était donc de faire un système de gain de mana avec le temps. C'était relativement facile mais cela requiert l'utilisation et la vérification de plusieurs variables.

Le premier sort est un bouclier. Lorsque la touche d'activation du bouclier est pressée, l'image du bouclier est activée et le personnage est invincible aux coups normaux. Il peut toutefois prendre des dégâts via la méthode ignorant l'armure.

Le deuxième sort est une boule de feu partant du joueur et progressant

en ligne droite en face de la position de base du joueur.
Une boule de feu est créée puis détruite à chaque utilisation.

3.2.5 Pièges

Afin de complexifier le jeu j'ai ajouté des pièges que j'ai dessinés.
Le premier piège est un objet qui se déplace selon une trajectoire que l'on peut déterminer, on prend des dégâts en rentrant à son contact.
La difficulté était que les points formant la trajectoire soient pris en compte correctement.
Le deuxième piège est un interrupteur qui s'active lorsque le joueur passe dessus.
lorsque ce piège est activé, une lance se dirige horizontalement vers un point d'arrivée défini.
La difficulté était de récupérer les différents composants de l'interrupteur sans rentrer en conflit avec les composants des autres interrupteurs dans la scène.
Un autre piège est le pic. C'est un des pièges les plus simples, il inflige des dégâts tant que le joueur reste dans la zone du pic.
Le champignon est dans le même principe que le pic mais cette fois-ci bénéficie d'une animation et inflige des dégâts en ignorant l'armure du joueur.
Le dernier piège est le canon.
le canon tire un boulet vers le point d'arrivée défini, il y a un certain délai et, le boulet peut pousser le joueur tout en lui infligeant des dégâts.

3.2.6 Premier Boss

Pour cette partie je me suis servi d'une partie du programme d'Alexandre concernant l'intelligence artificielle.
Le boss va chercher le joueur puis se déplacer dans sa direction jusqu'à atteindre une distance limite. Lorsque la distance limite est atteinte le boss va rester immobile.
Le boss possède trois façons différentes d'infliger des dégâts au joueur.
Le premier moyen est de rentrer en contact avec le boss.
Le deuxième est de rentrer en contact avec la boule guidée lancée par le boss à intervalles réguliers.
le dernier moyen est de rentrer avec le laser tiré par le boss en ligne droite.
En parallèle avec le boss j'ai implémenté des rectangles apparaissant et disparaissant pour fermer la zone du boss lorsque le joueur rentre dedans et la rouvrir lorsque le boss meurt.

Pour cette partie, le plus difficile était de gérer les différentes attaques du boss et son déplacement. Encore une fois j'ai du changer la sauvegarde pour sauvegarder le fait que le boss a été vaincu.

3.3 Derniers Ajouts

Cette partie regroupe tout ce qui a été fait depuis la deuxième soutenance.

3.3.1 Barre de PV du boss

Comme cela avait été dit à la deuxième soutenance il manquait la barre de vie du premier boss.

Je l'ai donc ajoutée sous format de slider, c'est à dire la taille de la jauge de pv du boss dépend de son nombre de pv restant.

3.3.2 Le Magasin

Un autre point important est le magasin.

Cet élément permet de gagner des items et des armes en échange de somme d'argents prédéfinies.

Dans cette partie il a donc fallut gérer l'inventaire du joueur et le mettre à jour et donc encore modifier la sauvegarde évidemment, qu'est-ce qu'on s'amuse! On peut ouvrir l'interface du magasin en interagissant avec, une fenêtre s'ouvre et on peut déplacer un curseur pour choisir l'arme que l'on veut au moyen d'une touche sur le clavier.

Une autre touche permet de changer de page et acheter des items. Lorsqu'on change d'item ou d'arme le nom et le prix de ce qui est sélectionné est mis à jour.

Ici la grosse difficulté était de bien gérer les différents images pour donner l'impression que le curseur bouge et afficher les bon objets à acheter. Il y a aussi eu un bug d'affichage sur le canvas lié à la résolution choisie dans les paramètres.

3.3.3 Les Armes

Le principe des armes est assez simple, on associe à un dessin une zone de dégâts et on fait bouger l'arme lorsque la touche d'attaque est pressée. Ici la difficulté était d'équiper l'arme lorsque le bouton correspondant est pressé ou de changer l'indice de l'arme à équiper lorsque le bouton correspondant est pressé.

Un autre problème était de faire la rotation de l'arme pour simuler le coup.

Il y a donc 5 armes différentes. L'épée, la lance le couteau, l'arc et enfin la hache.

elles ont toutes des fréquences d'attaques et des dégâts différents.

le couteau attaque le plus rapidement mais fait le moins de dégâts tandis que la hache est la plus lente mais fait le plus de dégât.

L'arc est différent, il fonctionne sur le même principe que la boule de feu, il crée des flèches qui partent en ligne droite devant le joueur.

3.3.4 Réglage de erreurs

J'ai aussi du régler différents erreurs.

Par exemple j'ai fais en sorte que les coffres ne puissent pas être rouverts après avoir sauvegardé et chargé la sauvegarde.

J'ai réglé le mouvement et l'attaque du premier boss pour que ce soit plus précis.

J'ai aussi remarqué un erreurs lors de l'utilisation d'un item qui fait que les nombres des items changent.

Je m'étais juste trompé sur les variables des nombres d'items.

3.3.5 Deuxième boss

Le deuxième boss est cette fois immobile.

Il possède deux attaques, un pic se déplaçant horizontalement partant du boss, et des blocs tombant du plafond.

Le boss fait aussi apparaître un mur devant lui qui possède des points de vie et disparaît lorsqu'il en a plus.

Ici la difficulté était de gérer le mur et les blocs, pour pouvoir faire apparaître des blocs aléatoirement et faire disparaître le mur lorsqu'il le faut.

3.3.6 Troisième boss

Le troisième boss possède deux phases, une phase d'attaques à distance, et une phase d'attaque au corps à corps.

Gérer le fait qu'un boss possède deux phases à été le plus dur.

gérer les dégâts et les petites animations du boss était aussi une difficulté supplémentaire.

3.3.7 Quatrième boss

Le dernier boss cette fois est immobile.

Il lance une attaque de zone en deux temps, d'abord elle progresse en

ligne droite vers le joueur ensuite elle explose, augmentant ainsi la zone de dégât.

Gérer cette attaque à été difficile, surtout dans la gestion des zones de dégât.

Sa deuxième attaque est de lancer en ligne droite une balle qui rebondi jusqu'à toucher le joueur ou une attaque du joueur.

Ici le rebond a nécessité des recherches..

Enfin la dernière particularité du boss et qu'il peut se soigner lorsque sa main est attaquée.

3.4 Conclusion

Je vais maintenant conclure ma partie par mon point de vue sur les points importants de ce projet.

3.4.1 Gestion des outils

Je pense qu'un point assez important est bien sûr la gestion des outils que l'on a en main.

Il faut donc savoir faire des recherches mais aussi expérimenter aussi soi-même. Je dirais donc que la mise en place du projet n'était pas si facile que ça entre la prise en main de git, Unity et de la programmation C#.

Cependant je peux dire que maintenant j'ai pris l'habitude, et que dans le domaine de la gestion des outils, j'ai gagné de l'expérience.

3.4.2 Gestion du temps

Un des plus gros problème de ce projet est évidemment la gestion du temps.

Il faut prendre en compte que les cours les tp et les évaluations ont toujours lieu en parallèle du projet. Cela a donc considérablement réduit le temps disponible de travail sur le projet.

Je pense aussi avoir gagné de l'expérience dans ce domaine, même si pour moi la question de la gestion du temps restera assez complexe.

3.4.3 Travail de groupe

Le dernier point à aborder est le travail de groupe.

Nous avons eu plusieurs besoin de mettre en relation nos différents travaux pour pouvoir avancer, comme la sauvegarde du cycle jour/nuit pour moi.

Dans ces cas là, avoir un outil comme git est très utile, il permet à la

fois de voir le travail des autres mais aussi de travailler ensemble et individuellement.

3.4.4 Avis

Je finirais donc par dire que de manière générale j'ai trouvé le projet intéressant et que je pense avoir acquis pas mal d'expérience malgré les difficultés rencontrées.

3.4.5 Prise en main de la programmation dans unity

Un autre point abordé lors du début du projet est la programmation. En effet Unity a une manière de fonctionner bien à lui. L'intégralité du gameplay et de l'interactivité dans Unity est basée sur trois principes : les GameObjects, les composants et les variables.

Tout objet d'un jeu est un GameObject : c'est le cas des personnages, de la map, des accessoires, bref, de tout.

Les GameObjects ne peuvent rien accomplir seuls. Pour qu'il puisse interagir et faire des actions particulières, il lui faut des propriétés, obtenues par l'ajout de Composants (Components).

Les Composants définissent et contrôlent le comportement des GameObjects auxquels ils sont attribués. Un simple exemple est l'ajout d'un Composant Rigidbody à un objet pour qu'il tombe, ou même l'ajout d'un collider pour gérer les collisions de l'objet.

Les Composants ont de nombreuses propriétés modifiables, ou variables, qui peuvent être ajustées dans l'inspecteur, dans l'Éditeur Unity et/ou via un script. Nous pouvons dans l'exemple ci-dessus, régler la friction des collider, la transparence des sprites, qui sont les apparences des objets.

Ici, la difficulté était juste de s'approprier le fonctionnement de la programmation dans unity, finalement les tutos et la doc ont permis de gérer les différentes classes et méthodes de unity.

3.4.6 Déplacements du perso

Finalement les déplacements du personnage ne sont pas si compliqués, il faut utiliser les propriétés de Rigidbody pour les déplacements horizontaux. Il faut stocker le mouvement du joueur dans une variable pour pouvoir faire tourner le personnage quand la vitesse est négative et jouer l'animation de course.

La plus grosse difficulté est d'implémenter le saut. Pour cela plusieurs possibilités existent.

Celle que nous avons retenue est de vérifier si le personnage est au sol ou non via une variable. Enclencher le saut passera par la vérification de cette variable.

En plus du saut normal un deuxième saut plus grand a été implémenté (via une seconde variable notamment)

Un autre déplacement particulier mais plus difficile à gérer est la montée aux échelles.

Cette fois nous nous servons d'une box collider sur l'échelle pour savoir si le personnage est a portée. Si oui le joueur peut maintenant appuyer sur E pour activer la possibilité de monter verticalement au niveau de l'échelle.

3.4.7 Gestion de la barre de vie et des dégâts

La barre de vie est gérée avec 9 images de coeurs : chaque coeur se voit attribué un demi-coeur en dessous de lui puis un coeur vide, ce qui offre la possibilité de cacher le coeur complet pour ne voir qu'un demi-coeur puis le demi-coeur pour voir un coeur vide.

Il y a trois manières de perdre des coeurs : toucher un ennemi, appuyer sur h (pour les tests, cela disparaîtra dans la version finale) et enfin de tomber dans le vide.

3.4.8 gestion de la deathzone

Quand le personnage tombe, il verra une animation d'écran noir puis réapparaîtra là où se trouve le point PlayerSpawn.

De plus il perdra un coeur. Cette zone est en fait un collider que l'on peut ajuster, si le joueur rentre dedans alors l'animation d'écran noir se joue et le personnage est replacé au niveau de PlayerSpawn et perd un

coeur.

3.4.9 Gestion des Checkpoints

Les checkpoints sont des collider, qui permettent de déplacer PlayerSpawn à leurs position, lorsque le personnage rentre dedans et appuie sur E.

Le personnage regagne tous ses coeurs lors de cette action, c'est pour l'instant le seul moyen de regagner de la vie.

Le principe de nos checkpoints est de pouvoir les réutiliser à chaque fois que l'on veut. Cela permet donc les retours en arrière et ainsi une plus grande liberté.

3.4.10 Création de portails

Cette fois deux types de portails on été implémentés. Il y a d'abord le portail dans la même scène et ensuite le portail qui mène à une scène différente.

Le portail dans la même scène est bien sûr le plus simple, il s'inspire du principe de la deathzone.

Donc lorsque le personnage rentre dans le collider du portail et appuie sur E, cela active l'animation de l'écran noir. Ensuite le personnage est téléporté au second portail. Le principe marche dans les deux sens.

Maintenant, le portail vers une autre scène possède quelques ressemblances.

Il téléporte le joueur lorsqu'il appuie sur E en étant en contact avec le collider du portail. Il y aura encore une fois l'animation d'écran noir. La principale différence avec le portail normal est que ce portail doit posséder une méthode qui sauvegarde certaines données : personnage, caméra, santé ... On appelle donc DontDestroyOnLoad.

3.4.11 Ouverture de coffres

Cette interaction particulière est plutôt simple, elle est constituée d'une animation d'ouverture de coffre, qui s'active lorsque le personnage est en contact avec le collider du coffre et le joueur appuie sur E.

Cette interaction fera appel à l'inventaire et sûrement aux items de notre jeux, qui ne sont pas encore mis en place.

La création de l'inventaire des items puis leur sauvegarde s'annonce être

la partie la plus difficile. Il ne restera plus que les PNJ à ajouter après cela.

3.4.12 Problèmes et difficultés

Cette partie se termine sur les problèmes et difficultés rencontrés, même s'il n'y a pas eu de très gros problèmes.

La principale difficulté au début était la prise en main de Unity. Elle a ensuite été la capacité à se retrouver dans le projet et à suivre de manière précise chacune des actions effectuées.

En effet les problèmes rencontrés étaient principalement dus à des erreurs d'inattention ou des fonctionnalités activées qui rentraient en conflit avec certains autres objets.

Le problème le plus embêtant sont quand on veut vérifier et faire plusieurs choses en même temps. Dans ce cas, il faut bien séparer les actions et les variables.

4 Alexandre

Je suis le responsable du groupe de ce projet. Je me suis vu confier les tâches de l'organisation, la mise en place d'une intelligence artificielle et de l'implémentation de la fonctionnalité multijoueur.

4.1 L'intelligence artificielle

Après m'être occupé à organiser le projet, je me suis penché sur la mise en place de l'intelligence artificielle.

Il s'agit de la principale fonctionnalité que j'ai présentée lors de la première soutenance.

4.1.1 Description

Dans ce type de jeu, l'intelligence artificielle n'a pas le besoin d'être très poussée. En effet, nous avons prévu pour les mobs de base de les faire suivre un chemin généré entre la position de l'ennemi et celle du joueur. Le mob ne pourra cependant pas quitter une certaine zone, le but étant de ne pas regrouper tout les ennemis de la carte sur le joueur (cela pourrait peut-être devenir un mod du jeu Hardcore). Le but sera aussi d'éviter que les mobs tombent dans le vide en allant chercher le joueur.

4.1.2 Implémentation

Pour implémenter l'intelligence artificielle, nous avons choisi d'utiliser le package externe d'Unity 'Astar'. Il comprend des scripts de base pour la création de chemins (un pathfinder pour la zone de recherche et un seeker pour utiliser cette zone). Il comprend aussi des scripts plus avancés et automatisés mais nous ne les avons pas utilisés.

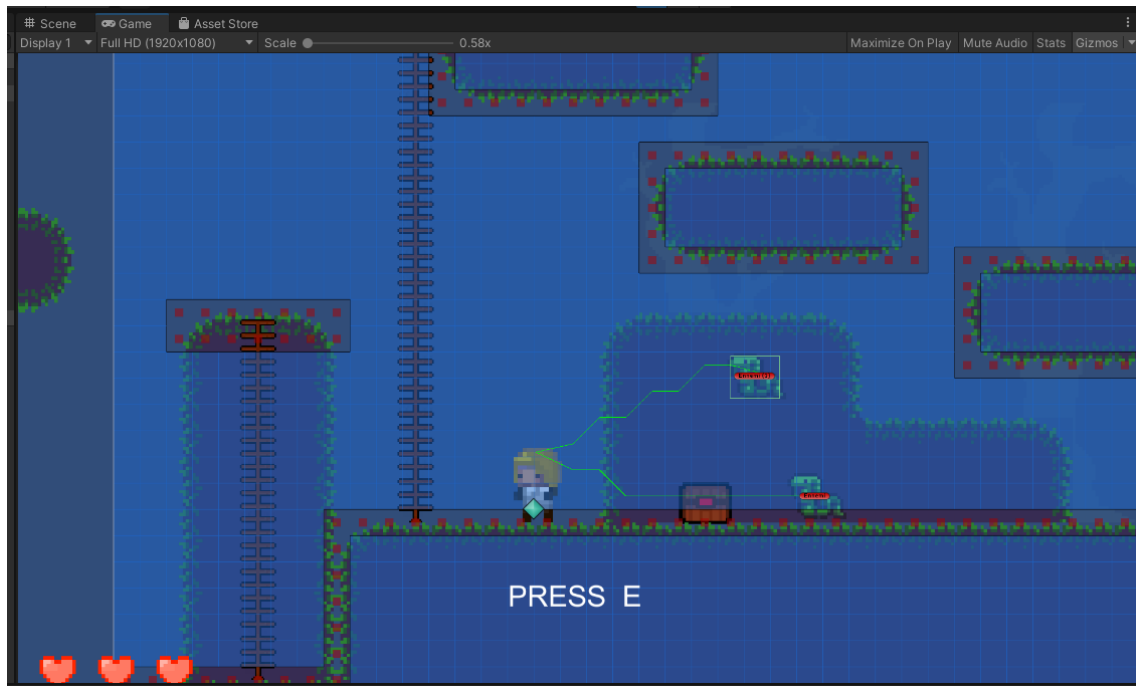


FIGURE 2 – Aperçu du Pathfinding

4.1.3 Difficultés

Lors de l'implémentation, nous avons rencontré beaucoup de difficultés.

La première était la bonne mise en place des paramètres des éléments mis en jeu. De la définition de la zone du pathfinder au éléments composant l'ennemi, certains paramètres ont du être réglé au centième près.

La seconde est certainement celle qui nous à pris le plus de temps et d'énergie à régler. Le message d'erreur indiquait qu'il ne trouvait pas de point d'arrivé. Après des heures de recherche, de débogage, de recompilation des bibliothèques d'Unity et aussi de plusieurs git rebase, l'erreur fut enfin trouvée. Elle provenait d'une ligne de code manquante dans le script du mouvement du joueur. Cette ligne de code permet la mise à jour de la position de l'objet 'Transform' rattaché au Player. Le chemin ne pouvait alors pas trouver de point d'arrivé puisque la position n'était pas réglée.

Une troisième difficulté apparues après les autres concernent les déplacements de certains monstres. Une optimisation à du être nécessaire pour corriger le fait que les monstres pouvait quasiment voler. Cette erreur venais surtout d'un mauvais usage des outils de détection du sol.

4.2 Le Multijoueur

Ensuite, je me suis préoccupé de la partie multijoueur de notre jeu. Il s'agit de la principale fonctionnalité que j'ai présenté lors de la seconde soutenance.

4.2.1 Description

Bien que de base les jeux présentant une dimension narrative ne nécessitent pas forcément une partie multijoueur, il peut tout de même être intéressant de pouvoir taper sur des ennemis avec des amis ou encore faire une course à celui qui aura tué le plus d'ennemis en un temps imparti. C'est en partie la raison qui explique que notre partie multijoueur se focalisera sur une map spécialement conçue pour, où des vagues de monstres arriveront et où le but sera de survivre le plus longtemps possible. Nous avons souhaité introduire d'autres modes de jeux mais nous n'avons malheureusement pas eu le temps. De toute manière, il s'agissait d'une partie mineure de notre jeu, cela devait passer en second plan.

Comme expliqué plus haut, notre multijoueur se focalisera sur un gameplay du genre mini-game, où les joueurs devront survivre le plus longtemps possible à des vagues de monstres. Le gameplay comprend l'adoption des principales dernières fonctionnalités comme les nouvelles armes ou l'utilisation des potions. Du côté des monstres, il peut arriver que sur certaines vagues un boss apparaisse. Un chrono et un compteur des vagues auquel les joueurs ont survécu est aussi affiché.

4.2.2 Implémentation

Afin d'implémenter notre fonctionnalité, nous avons choisi d'utiliser 'Proton', un service qui propose d'héberger des serveur accompagné de d'une API complète. Proton propose un service gratuit jusqu'à 20 joueurs, amplement suffisant pour notre projet (en tout cas pour l'instant!). Par ailleurs, ce service est plutôt populaire : il est utilisé par de très nombreux éditeurs de jeux pour énormément de jeux, ce qui le place aujourd'hui à la première place des services multijoueurs les plus utilisés.

Photon est accompagné d'un API spécifique à Unity : 'PUN' (Photon Unity Network). Celle-ci se base sur un système de lobby et de room. Pour expliquer ceci, le joueur doit d'abord se connecter à un lobby, puis, à l'aide d'un nom de room, il peut entrer dans un room déjà existante ou en créer une et la rejoindre.

Dans notre jeu, la connexion au lobby se fait automatiquement. Le joueur arrive alors sur une page de connexion qui lui demandera un pseudonyme et le nom de la room qu'il souhaite rejoindre. Si la room n'existe pas, la room sera créée automatiquement avec le nom entré. Si aucun nom de room est entré, le jeu cherchera alors à se connecter à une room aléatoirement. Si aucune room existe, il sera demandé au joueur d'en créer une.

Une fois arrivé dans la room, les joueurs et les ennemis sont instanciés et les informations concernant la position et l'état des entités présentes sur la scène de jeu sont émises et reçues par tous les membres connectés à la room. Par exemple, quand le joueur se déplace, les valeurs de sa position sont envoyées sur le serveur et redistribuées aux autres membres du serveur. De cette manière chacun peut savoir graphiquement où se situent les autres personnes présentes avec lui et l'état de leur santé par exemple.

C'est de cette manière que notre fonctionnalité Multijoueur sera implémentée.

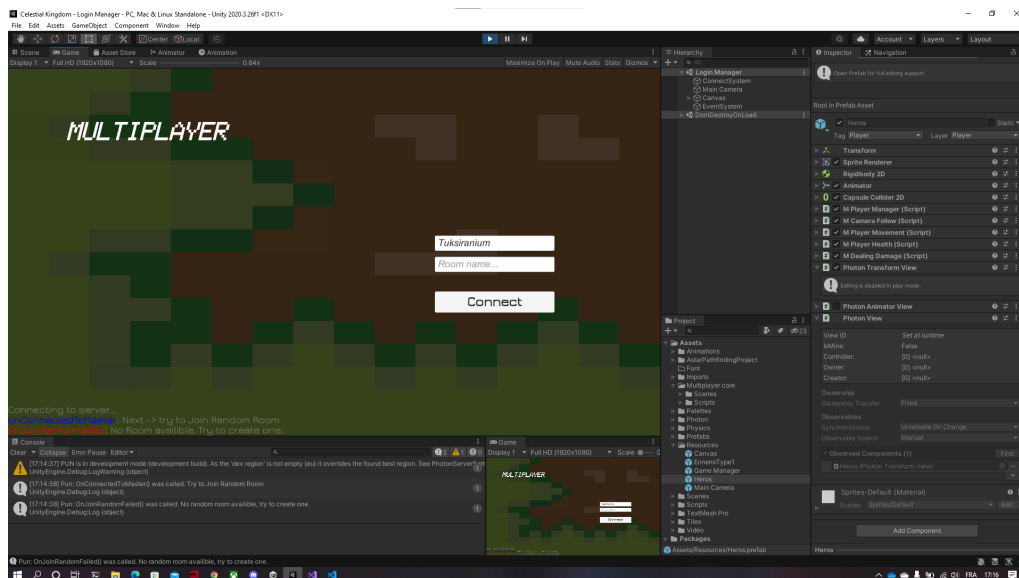


FIGURE 3 – Aperçu de la page de login quand aucune room n'existe et que on essaye de se connecter sans mentionner de room.

4.2.3 Difficultés

Lors de l'implémentation, nous avons rencontré trois difficultés majeurs.

La première était que beaucoup des fonctionnalités multijoueur ne sont pas compatible avec celles du monojoueur (en tout cas, de la manière dont on l'a implémentée). Nous avons effectivement commit l'erreur de ne pas nous être préoccupé de cette partie au lancement du projet. Cela a fait que beaucoup de scripts on dû être réécrit (en partie ou dans leurs intégralités). Notre manière d'instancier les objets aussi a dû être modifier. Dans la partie solo, on instancie les ennemis et le joueur directement dans la composition de la scène, alors que dans la partie multijoueur, on doit instancier les entités dans le code avec l'API de Photon pour que les données soit retransmises à tous les joueurs. La modification des scripts avait pour but aussi la synchronisation des données avec tout le monde.

La seconde était lors de l'implémentation, nous avons eu un problème de transmission de données au serveur. Nous avons mis beaucoup de temps à régler ce problème en raison du fait que très peu de problèmes du genre on été signalé sur les forums sur internet. Finalement, après avoir réécrit la moitié des scripts dédiées au multijoueur, le problème c'est résolu de lui même. Nous n'en connaissons donc malheureusement pas la cause. Nous suspectons néanmoins le fait que nous avons pour la première fois créé nos scripts en s'inspirant des fonctions issues d'un tuto proposé par Photon, car le type de jeu était bien différent du notre.



FIGURE 4 – Aperçu du jeu multijoueur sur le terrain d’essai.

La troisième est simplement le fait que vu que la partie multijoueur s’est retrouver retardée à causes de nos erreurs d’organisation, beaucoup de nouvelles fonctionnalités voyaient le jour, trop rapidement pour être facilement implémenté dans ce mode de jeu. Cette difficulté aurait sûrement pu ne pas exister si nous avions correctement défini des règles et conventions de travail entre nous, car à chaque fois que l’on souhaitais utiliser une fonctionnalité d’un autre membre du groupe, il devait souvent se trouver présent pour expliquer et aider.

4.3 Site Web

J'ai été chargé de faire le site web de notre organisation et qui présente notre jeu.

Pour information voici le lien qui y mène : <https://la-frelu.github.io/>

4.3.1 Au commencement

J'ai utilisé une fonctionnalité du site Github (<https://github.com>) qui permet à partir des fichiers déposés sur un repo git de faire fonctionner/héberger notre site internet.

4.3.2 Création

Ne possédant pas de solides compétences en langage HTML, CSS et JavaScript, mais voulant quand même créer quelque chose de moi-même, j'ai décidé de partir de zéro. Aucune template, rien. C'est pourquoi notre site est d'apparence très simpliste. Il n'est pas forcément non plus très complet mais assez pour servir de wiki afin de guider les joueurs si sont perdus.

4.3.3 Remplissage

Nous avons enfin rempli la section "notice" du site par tout le contenu d'aide aux joueurs, et j'ai enfin tenté d'adapter au mieux le site à l'image de la frelu, notre organisation.

4.4 Mon ressenti sur le projet

Faire notre propre jeu vidéo est une très bonne expérience. Cela nous à permis d'apprendre beaucoup de choses tant au niveau de l'organisation qu'au niveau technique. Nous avons commit un certain nombre d'erreurs, certaine était plus grave que d'autre, mais cela nous a permis d'en tirer de très bonne leçons.

Notre plus grosse erreur était surement la mauvaise prévision de ce que serait le côté multijoueur à implémenter. Le jeu à été développé de base sans réfléchir au difficultés d'implémentations de cette fonctionnalité. A cause de cela, beaucoup de scripts on dû être modifiés et certaines scènes on du être adaptées (notamment au niveau de l'instanciation des entités). C'est une erreur qui a causée la perte de beaucoup de temps et aussi le fait que la partie multijoueur soit assez pauvre en fonctionnalités. Nous avons donc compris qu'il est très important de se soucier de l'implémentation de toutes les fonctionnalités dès le tout début du projet et ainsi de prévoir une manière de travailler commune à tous pour que la liaisons entre ces différentes fonctionnalités se fassent sans poser trop de problèmes.

Selon moi, ce projet nous à aussi permis de mieux percevoir la notion d'organisation d'un projet. Que ce soit au niveau du planning avec la répartition des tâches qui n'a clairement pas été simple au début vu notre manque total d'expérience, où à la gestion des différents outils nécessaires (comme git) qui à demandé plus d'apprentissage que ce que l'on pensait au début. Par ailleurs, en parlant des outils, il est bon de mentionner que ces apprentissages, notamment la maîtrise de git, à été particulièrement bénéfique pour nous. Cela nous à permis d'apprécier la puissance de cet outils et de nous sauver plus d'une fois de situation assez critiques.

Enfin, si je devais mentionner une ultime satisfaction, je citerait ceci : "Ce projet m'a permit de mieux comprendre les coulisses d'un univers qui me fascine." Je m'explique. J'ai toujours aimé les jeux vidéos. Plus particulièrement, j'aime le fait que le jeu vidéo regroupe quasiment toutes les formes d'arts, et cela me fascine. De la création du déroulement du jeu, de l'histoire, du visuel graphique en passant par l'ambiance sonore et les musiques, plus le jeu se voudra complet, plus sa complexité augmentera. J'ai toujours aimé chercher sur un jeu ses limites jusqu'à dans les moindres recoins, ou me laissé emporter par l'aspect visuel de

celui-ci combiné avec la bande son. Avant d'acquérir plus de connaissances sur la manière de créer des jeux vidéos, je tentais d'imaginer tant bien que mal ce processus au fur et à mesure que je progressait dans ces mondes virtuels. Maintenant, avoir pu en créer un moi même m'a permis de comprendre beaucoup de choses, notamment le fait qu'il est quand même assez important d'avoir un minimum une âme d'artiste lorsque l'on souhaite développer un jeu vidéo. Malgré certains aspects, je pense avoir réellement apprécier ce projet.

5 Nathan

Je me suis fixé comme rôle d'être le responsable designer graphique et audio du groupe étant donné que j'avais des bases dans ces domaines contrairement au reste de mon groupe.

Cela à impliqué de me familiariser au maximum avec ces applications et également d'aider les autres membres du groupe en cas de problème.

Pour rappel, pour la création et édition visuelle :

- j'ai principalement utilisé Gimp, un logiciel d'édition visuelle que j'utilisais déjà quotidiennement avant le début du projet.
 - j'ai également utilisé Aseprite, un logiciel d'animation spécialisé dans le pixel art qui nous permettait de pouvoir créer nos propres animations.
- Pour la création et édition audio : j'ai utilisé FL Studio.

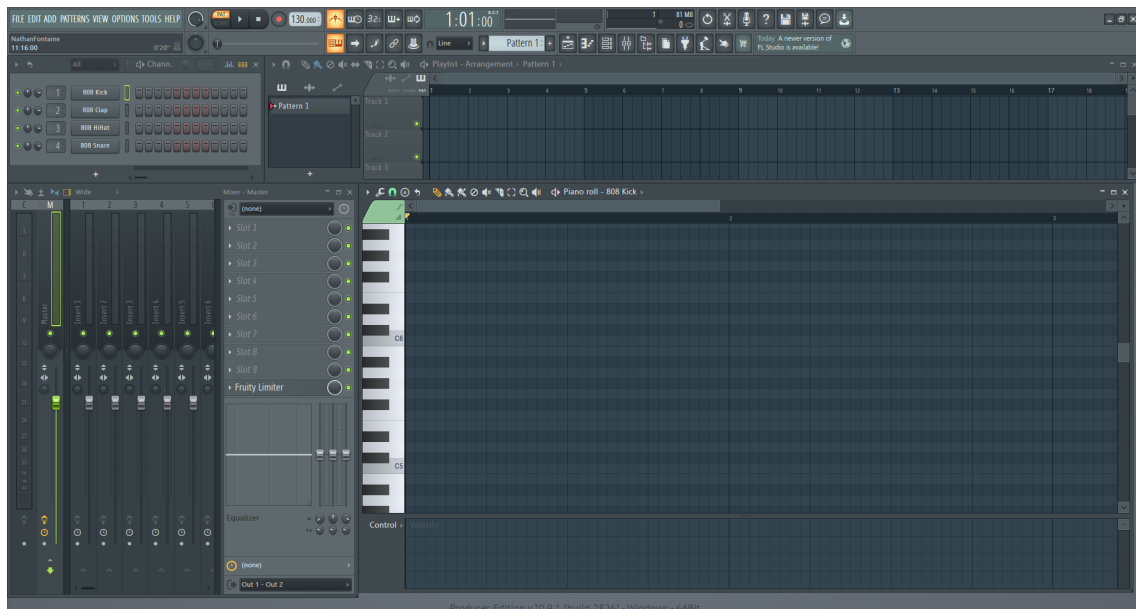


FIGURE 5 – Aperçu de FL Studio

Je vais maintenant vous résumer ce que j'ai réalisé lors des 3 soutenances de manière chronologique.

5.1 Soutenance 1

5.1.1 Menus d'affichages

J'ai tout d'abord commencé le projet par l'implémentation du Menu principal car il fallait créer une première base rapide du jeu qui soit fonctionnel.

Pour ce faire, j'ai créé un script qui permet d'intervenir entre différentes scènes et un autre script qui permet aux boutons d'avoir des actions définies.

Lors de cette soutenance, j'ai également mis en place le système de mort, que ce soit dans les scripts ou dans les menus, ainsi que le système de mise en pause d'une partie en cours.

J'avais à ce moment là seulement réaliser un aperçu de ce qu'allait être notre menu d'options.

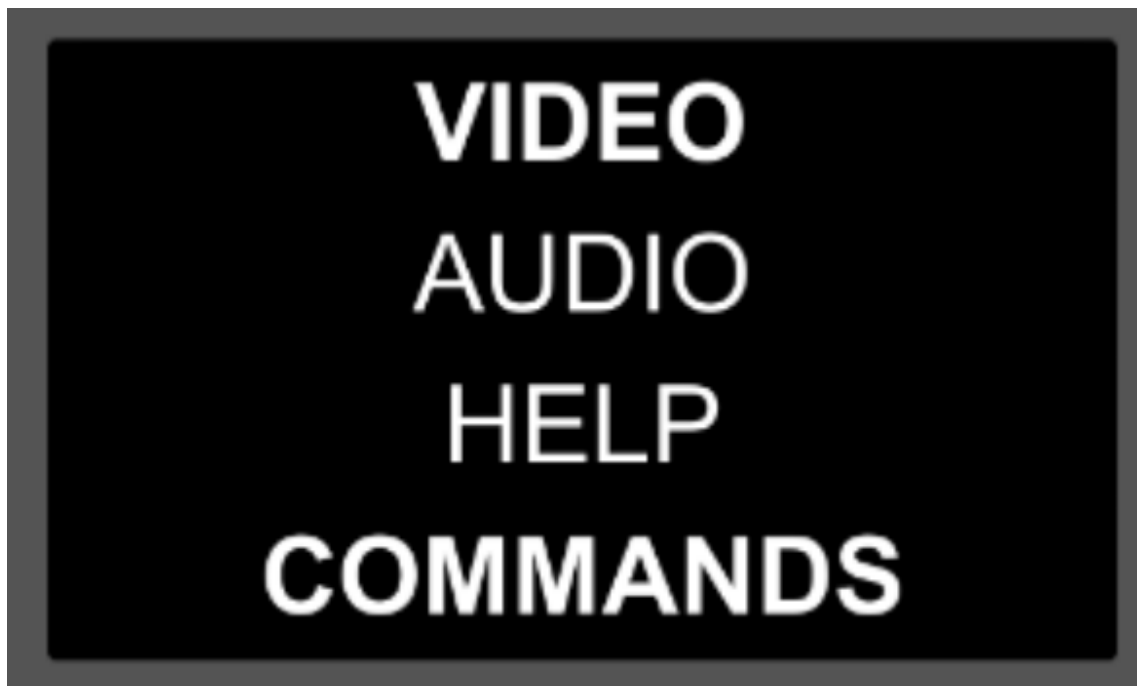


FIGURE 6 – Notre menu d'options à l'origine

5.1.2 Personnages

J'ai créé sur Gimp les premiers aperçus du personnage principal ainsi que les premiers monstres implémentés dans le jeu : pour ainsi dire, à ce moment là il n'y avait que le serpent. J'ai par la suite créé les animations du personnage principale à partir de Aseprite pour les déplacements gauche/droite ainsi que les sauts - C'est Dimitri qui les a implémentés dans le jeu.

5.1.3 Petite Conclusion

A ce moment là du projet, je n'étais encore pas très confiant dans l'utilisation de Unity ainsi que l'investissement mis dans le projet, ce qui m'a parfois valu de me retrouver dans des problèmes insolubles et de devoir changer mes objectifs.

5.2 Soutenance 2

5.2.1 Launcher

J'ai créer un launcher du jeu qui s'est avérer plutôt inutile lorsque j'ai implémenté le menu d'options, mais cela m'a toutefois permis de me donner une première idée de la création graphique d'application xaml avec du C#.

5.2.2 Menu Options

J'ai créer un menu options qui permet au joueur de pouvoir faire des changements au niveau des commandes, au niveau de la résolution affichée à l'écran ainsi qu'au niveau du son. Pour ce faire j'ai du suivre de nombreux tutoriels sur internet afin de parvenir à correctement mettre en place ces options. Initialement j'avais prévu de décomposer le menu options en 4 parties : Les boutons "video", "audio", "help" et "commands". Finalement, j'ai décider de le décomposer seulement en 3 parties afin de mettre la documentation d'aide sur le site web.

La manière dont on change les touches dans le menu options à été difficile à mettre en place mais finalement après avoir longuement recherché sur la documentation et des forums j'ai finalement réussi à faire fonctionner ce système à ce moment là.

5.2.3 Personnages

J'ai un peu modifié au fur et à mesure le personnage principal pour qu'il ait un meilleur rendement dans le jeu.

A ce moment là, j'étais également parti pour faire toutes les animations du personnages selon les différentes armes qu'ils portaient ce qui est très chrono-phage pour le coup, c'est alors que Dimitri m'a signalé que l'on pouvait tout simplement mettre les modèles d'armes sur le personnage principal dans le jeu directement et gérer les armes avec des mouvements sur les objets ce qui m'a pour le coup fait un gain de temps colossal. De plus cela aurait été quasi impossible pour gérer les colliders des différents armes avec l'ennemie puisqu'elle serait directement intégrée à

l'animation du personnage, ou alors il aurait fallu changer la hitbox du personnages en fonction de l'arme qu'il possède en main ce qui aurait été un travaille de longue haleine voire inatteignable pour moi seul.

5.2.4 Travail sur les maps

J'ai mis en place un programme de génération procédurale qui nous permettait de générer des croquis qui par la suite aurait pu nous servir pour réaliser les différentes maps et niveaux du jeu.

5.2.5 Petite Conclusion

Je trouve qu'il est très compliqué de prévoir ce que l'on doit faire en terme d'objectifs et d'organisation lorsque c'est la première fois que l'on fait quelque chose du genre, parce que l'on peut facilement se compliquer la vie pour rien si l'on n'utilise pas les bons principes et procédés. C'est néanmoins une leçon importante que je peux en tirer.

5.3 Soutenance 3

5.3.1 Menu Options

J'ai finalement enlevé la possibilité au joueur de changer les touches car après en avoir à rajouter au fur et à mesure du jeu, je n'avais pas pensé au fait qu'il pouvait y avoir des conflits entre les commandes et la manière dont j'avais fait mon programme ne pouvait pas prendre en compte la dualité (voire plus) d'une commande, c'est pourquoi je me suis remis sur une interface qui explique au joueurs les touches qui permettent de réaliser tel ou tel action.

5.3.2 Personnages

Pour cette dernière soutenance, j'ai réalisé de nombreux personnages dont le boss de glace - ou de cristal, vous l'appellez comme vous voulez - (les autres bosses ont été réalisé par Dimitri car il était en avance sur ce qu'il avait à faire donc il m'a un peu aidé). J'ai également rajouté une animation d'attaque et de mort pour chacun des monstres du jeu pour donner un peu plus de "vie" aux personnages.

5.3.3 Jaquette DVD et Notice du Jeu

Étant donné qu'il faut réalisé le boitier du jeu, je m'y suis alors penché dessus et j'ai commencé par recherché les dimensions nécessaires pour le format d'un boitier de jeu puis je me suis alors mis à m'inspirer d'un

boitier déjà existant pour pouvoir réaliser le notre.

Sincèrement, je suis très fière de l'apparence final du boitier même s'il y a eu quelques défauts d'impressions notamment au niveau des couleurs. Pour la notice du jeu, je ne l'ai pas encore imprimé mais je suis entrain de la faire. Elle suivra le même principe que sur la page "help" de notre site web.

5.3.4 Installateur

Pour créer l'installateur du jeu, j'ai utilisé l'application Inno Setup qui permet de facilement créer un installateur d'une application .exe étape par étape. Cette application renvoie une exe qui dès qu'il est exécuté lance la fenêtre d'installation du jeu. En revanche, je n'ai pas créer de dés-installateur car je considère que celui de windows suffit amplement pour cette tâche et qu'il est donc inutile de faire cette tâche supplémentaire.

5.3.5 Fonds d'écrans et visuels

J'ai retravaillé tous les fonds d'écrans des niveaux, des menus ainsi que du menu principal. Cela constitue un travail assez long de dessin sur Gimp et permet d'améliorer l'expérience de jeu. J'ai également retravaillé tous les visuels des boutons en changeant notamment la typographie de Unity, cela c'est avérer un peu plus compliqué que prévu dans certains cas mais j'ai finalement réussi à mettre la même typographie dans le jeu pour tous les éléments écrits et visuelles.

5.3.6 Musique et sons

J'ai rajouter pour cette 3ème soutenance un script qui permet de gérer la playlist de musique que le jeu lancera selon ce qu'on fait et selon l'endroit et la scène dans laquelle on se trouve.

J'ai également mis en place les SFX (sounds effects) concernant les déplacements du personnages, les déplacements des monstres, les bruits d'attaques des armes, le bruit du décors environnant etc.. Par ailleurs, j'ai créer les musiques qui constituent le jeu ce qui est plutôt incroyable je trouve.

5.4 Ressentie

Je me rend compte à la fin du projet que tout ce que j'ai réalisé a servi à quelque chose et m'a permis de m'améliorer dans tous les domaines que j'ai travaillé, je me sens énormément plus productif en terme de quantité de choses que je peux réaliser mais également en qualité, par exemple je navigue très rapidement sur Gimp et je peux produire quelque chose en très peu de temps ce qui n'était absolument pas le cas au début. Je pense que ce type de projet est intéressant et permet de vraiment comprendre les méthodes d'organisations et du fonctionnement optimal d'un groupe. Même si je pense qu'on aurait tous pu faire un peu plus, je suis globalement très satisfait de ce que nous avons réalisé. Je pense également qu'à force de faire du Latex, je vais devenir fou car je suis celui qui a relu toutes ces pages avant la publication finale .. (un peu d'humour).

6 Conclusion

Merci d'avoir eu le courage d'être arrivé jusqu'ici et d'avoir lu notre rapport de soutenance.

Nous vous prions d'agréer nos plus sincères salutations.
soussigné, FRELU ;)

7 Annexes

7.1 Impression DVD



FIGURE 7 – Notre jaquette DVD

7.2 Personnages

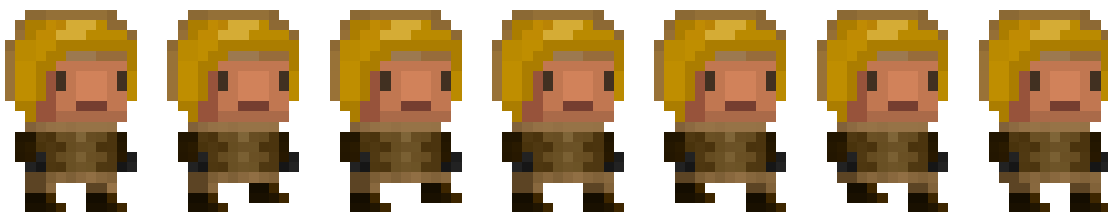


FIGURE 8 – Animations déplacement

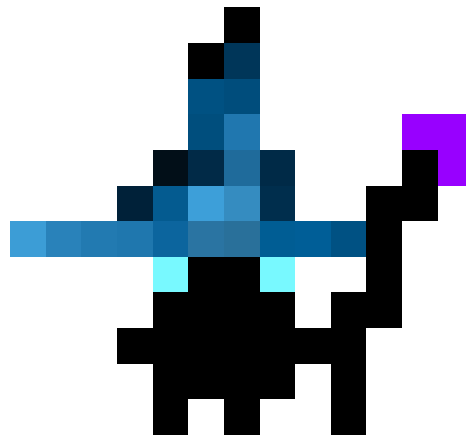


FIGURE 9 – Un sorcier

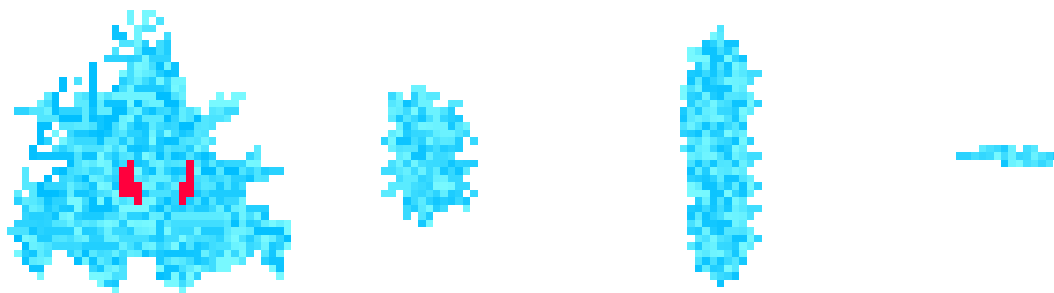


FIGURE 10 – Le boss de glace avec ces compétences



FIGURE 11 – L'animation d'une torche

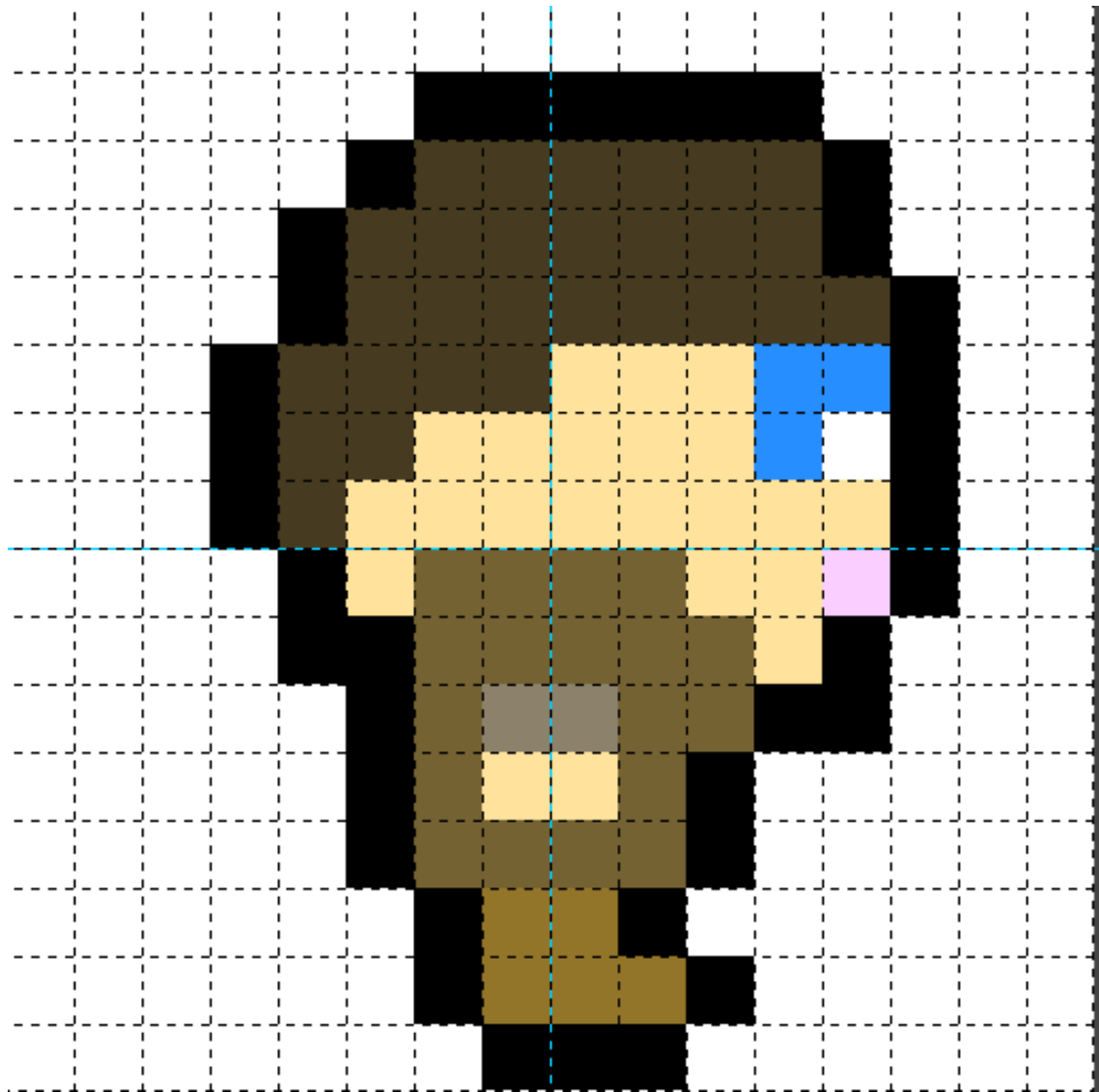


FIGURE 12 – Notre personnage à l'origine (version alpha V.1.0)



FIGURE 13 – Notre personnage avec son épée (à l'origine)



FIGURE 14 – Exemple de fond d'écran