

PROJET D'INFORMATIQUE S2 :

RAPPORT DE SOUTENANCE

EPITA LYON

CHEF DE PROJET : ENZO DELOYE
ALICE CARIOU
BAPTISTE PERRIER

PROJECT.EXE



Table des matières

1	Introduction	3
1.1	Présentation du groupe	3
1.1.1	Enzo Deloye	4
1.1.2	Alice Cariou	4
1.1.3	Baptiste Perrier	5
1.2	Objet d'étude	6
1.3	État de l'art	6
2	Découpage du projet	9
2.1	La physique du jeu	9
2.2	La caméra	9
2.3	Technologique et Méthodologique	9
2.4	Opérationnel	10
2.5	Interface graphique	10
2.6	Ambiance	10
3	Organisation du Projet	11
3.1	Répartition des tâches	11
3.2	Avancement du Projet	12
4	Présentation général du projet	13
4.1	Site web - Enzo	13
4.2	Joueur - Enzo	14
4.2.1	Les statistiques	15
4.2.2	L'attaque à distance et les projectiles	16
4.3	Interface - Alice/Baptiste	18
4.3.1	Les menus principaux- Alice	18
4.3.2	Le choix de l'arme - Alice	20

4.3.3	Mini Map - Baptiste	21
4.4	Graphismes - Alice	22
4.4.1	Le personnage - Alice	22
4.4.2	Les boss - Alice	22
4.4.3	Les environnements - Alice	23
4.4.4	Potions et vie - Alice	25
4.4.5	Les mobs - Alice	27
4.4.6	les armes - Alice	27
4.4.7	autre - Alice	28
4.5	Audio	29
4.5.1	La bande-son - Alice	29
4.5.2	Les bruitages - Baptiste	29
4.6	Génération - Baptiste	31
4.7	Ennemis - Baptiste/Enzo	35
4.7.1	Classe mère IA - Enzo/Baptiste . . .	35
4.7.2	Boss - Baptiste	36
5	Bilan général et personnel	38
5.1	Enzo Deloye	38
5.2	Alice Cariou	38
5.3	Baptiste Perrier	39
6	Conclusion	41
7	Bibliographie	43
8	Annexes	43

1 Introduction

Qu'est-ce que la vie ? Non, plus sérieusement, qu'avons nous fait comme projet pour ce deuxième semestre et pourquoi avons nous choisi de réaliser cela ? Notre objectif pendant ce projet était de faire quelque chose qui puisse tous nous intéresser et d'apprendre à utiliser de nouveaux outils. Nous voulions vraiment que cela nous apporte à tous de nouvelles connaissances y compris à ceux qui avaient déjà fait de la programmation auparavant.

Pour cela, nous avions choisi d'utiliser Godot ce qui nous permettait de nous démarquer par l'utilisation d'un outil moins commun que Unity et cela permettait aussi aux personnes ayant déjà utilisé Unity de découvrir un nouvel outil.

Malheureusement, la tentative d'utiliser Godot s'est soldée par un échec puisque certain ont rencontré de sérieuses difficultés à avancer avec Godot et à se renseigner à l'aide de la documentation, qui est moins développée que celle de Unity.

Cela nous a fait perdre bien trop de temps, et nous avons du prendre la décision qui s'imposait. Nous avons donc finalement décidé d'utiliser Unity comme outil afin d'avoir quelque chose de plus commun et avec une documentation plus fournie ainsi qu'une communauté plus active.

Pour l'idée du projet en lui-même, l'idée de faire un jeu semblait la plus évidente puisque c'est quelque chose d'assez visuel où il est simple de voir la progression. Cela rendait le travail à faire un peu plus ludique et permettait de voir vraiment l'objectif que nous voulions atteindre. De plus, cela permettait de profiter des capacités en dessin d'Alice. Pour le style de jeu, nous en avons parlé entre nous et comme Yann, Alice et Baptiste étaient des fans de rogue-like nous nous sommes dit que cela pourrait être intéressant à faire et ça ne nous semblait pas trop compliqué à réaliser.

1.1 Présentation du groupe

1.1.1 Enzo Deloye

Je m'appelle Enzo Deloye, je redouble ma 1er année d'epita, mais je dois principalement m'occuper du site internet pour le projet je n'avais jamais coder de site web avant c'est pourquoi je voulais me dedans pour apprendre comme un site web était coder puis car ça me servir à l'avenir de savoir en coder un. Je dois aussi m'occuper des interactions du personnage avec son environnement. J'ai toujours apprécié les jeux vidéo et l'informatique et ayant déjà codé un jeu l'année dernière je trouve ça très intéressant comme expérience humaine à réaliser et je suis très content de le faire avec Alice et Baptiste.

1.1.2 Alice Cariou

Pour commencer, je m'appelle Alice Cariou. Pour me présenter simplement, j'ai toujours été curieuse et intéressée par des choses variées, dont l'informatique. Pourtant, je n'ai envisagé de faire mes études dans ce domaine que très tard, car d'autres domaines, tels que la musique me sont également très cher.

J'ai donc fait plusieurs stages pour avoir un point de vue assez complet pour pouvoir me décider, et après avoir passé une semaine dans cet environnement, il n'a pas fallu beaucoup de réflexion pour me rendre compte que c'était cela que je voulais faire.

Un de mes objectifs au cours de ce projet est donc d'utiliser les compétences que j'ai dans d'autres domaines, tels que le dessin pour les graphismes, que ce soit la réalisation des différents environnement, la création du personnage, des mobs, des boss, les armes, mais aussi des objets avec lesquels nous allons pouvoir interagir, tels que les potions de soin ainsi que celles de mana, ou encore les coffres.

Je joue de la musique depuis une dizaine d'année, et j'aimerai pouvoir me servir de mes connaissances dans ce domaine pour la bandeson, bien que je n'ai pas encore d'expérience en réalisation de musique à l'aide de logiciels. Grâce à tout cela, j'espère réussir à rendre notre projet unique! Bien entendu, je participerai aussi un maximum à la programmation du jeu en apportant ma contribution pour l'interface graphique, dont la réalisation des différents menus, mais aussi le développement du jeu.

1.1.3 Baptiste Perrier

Je m'appelle Baptiste Perrier, j'ai 18 ans et je suis étudiant en première année dans une école d'ingénieur en informatique. J'ai toujours été intéressé par tout ce qui touchait aux sciences, que ce soit les mathématiques, la physique mais surtout à la logique que ça soit de l'électricité, de l'algorithme ou tout autre domaine d'application.

J'ai commencé à utiliser des ordinateurs vers l'âge de deux ans. Les ordinateurs ont toujours été une partie intégrante de ma vie. C'est lorsque j'étais au collège que j'ai commencé à m'intéresser au domaine de la programmation. Et c'est sur les conseils d'un proche que j'ai commencé à apprendre le HTML. Par la suite, j'ai un peu essayé de toucher à tout mais je n'ai jamais vraiment approfondi.

J'ai pu participer à mon premier projet de groupe lorsque j'étais en seconde où j'ai appris à utiliser Unreal Engine et j'ai découvert la programmation orienté objet. Malheureusement, ce projet n'a pas pu être mené à bout à cause de conflits à l'intérieur du groupe. Lors de mon année de première, j'ai pu découvrir le domaine de la cyber sécurité grâce au site Root-Me.

Je n'ai pas forcément beaucoup approfondi ce domaine mais c'est à ce jour ce que je trouve le plus intéressant dans le domaine de l'informatique car c'est un domaine où l'on touche à beaucoup de langages différents et où l'on peut expérimenter plusieurs environnements de travail différents.



-Baptiste Enfant-

Dans ce projet j'ai été chargé de m'occuper de la génération des

niveaux et du contenu des salles ainsi que de la création des Boss. J'ai aussi pu travailler sur les IA des ennemis.

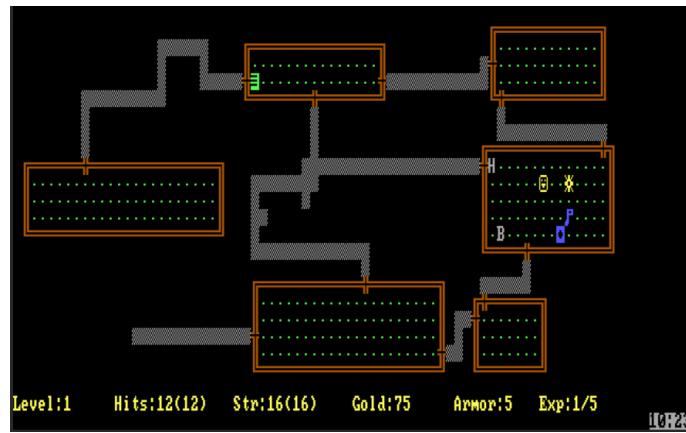
1.2 **Objet d'étude**

Notre but dans ce projet est tout d'abord d'apprendre à travailler en groupe et de travailler ensemble sur un projet commun avec une date limite et des résultats à rendre. De plus, un objectif important est de prendre du plaisir et de s'amuser en réalisant ce projet. Pour la partie individuelle, je pense que cela va nous apprendre à travailler autrement et à chercher ce qu'il nous faut pour notre projet par nous même avec peu d'aide extérieure, ainsi que de grande connaissance en programmation de jeu vidéo car, pour la plupart, nous n'avons jamais codé de jeu.

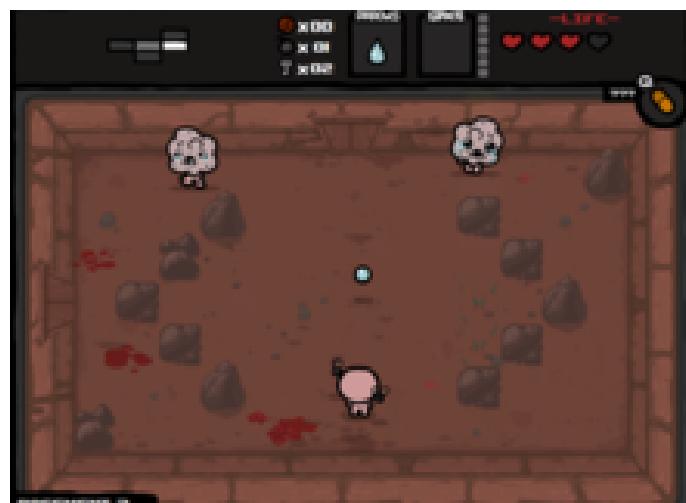
1.3 **État de l'art**

Nous avons dans l'idée de créer un jeu Rogue-like. Le premier jeu de ce type étant évidemment Rogue, créé une quarantaine d'années en arrière. Au niveau des jeux plus récents de ce type, on peut citer Dead Cells, Neon abyss ou encore Biding of isaac. Nous comptons nous inspirer de ce dernier pour la 2D vue du dessus et son principe de salle.

Pour ce qui est de la gestion de l'évolution du jeu, nous pensons plutôt nous inspirer de Soul Kinght. Et enfin, pour ce qui est de l'ambiance du jeu, nous envisageons plutôt quelque chose ressemblant à Don't Starve. Au cours du projet, ce type de jeu pourras nous permettre de nous confronter à la génération procédurale ainsi qu'à la gestion de la difficulté.



-Rogue-



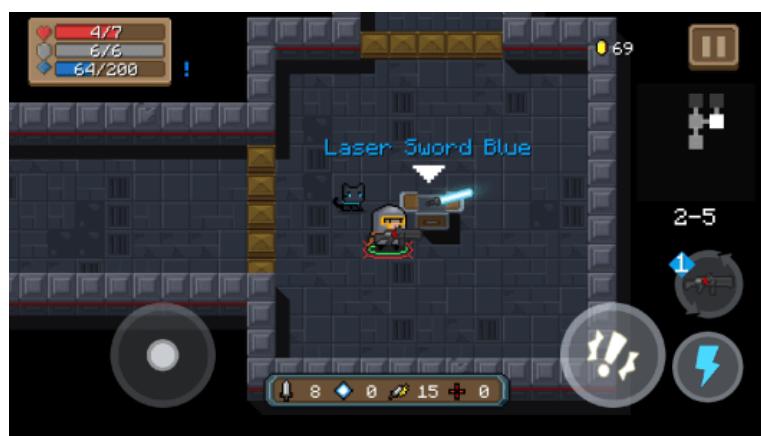
-Binding of Isaac-

Le jeu sera donc basé sur un principe de salles qui s'enchaînent. Dans chaque salle, le joueur devra affronter une vague d'ennemis suite à quoi il aura accès aux salles suivantes, et ceci jusqu'à ce qu'il atteigne le boss de l'étage.

L'objectif final étant de réussir à atteindre le boss du 3ème étage et de le vaincre. Après chaque victoire, la partie suivante verra sa difficulté augmenter, pour permettre aux joueurs d'avoir une difficulté adaptée à leur niveau pour qu'ils aient une meilleure expérience du jeu. Ainsi le jeu n'a pas réellement de fin.



-Don't Starve-



-Soul Knight-

2 Découpage du projet

2.1 La physique du jeu

Notre jeu sera en 2D, donc il n'y aura pas besoin de gravité, surtout dans un rogue like. Cependant, notre personnage ne pourra pas traverser tout et n'importe quoi, comme les ennemis ou roches dans certaines salles. Il aura des mouvements assez basiques qui seront dans les quatre directions et les diagonales. De plus, aussi bien les ennemis que notre personnage pourront tirer des projectiles qui auront une portée et une dynamique propre à chacun d'entre eux

2.2 La caméra

Comme dit précédemment, notre jeu sera donc en 2D. Pour permettre une vue claire et complète de celui-ci, nous avons choisi de positionner la caméra au dessus de chaque salle. Dans le cas dans lequel la salle serait trop grande, elle suivrait le joueur pour plus de visibilité .

2.3 Technologique et Méthodologique

Pour la conception de notre jeu, nous aurons besoin de plusieurs application qui faciliteront la tâche dans la conception de ce dernier mais aussi pour nous organiser. Tout d'abord, nous utilisons la plateforme « Discord » qui nous permet de discuter quand chacun est chez lui pour parler du projet mais aussi pour s'envoyer nos parties écrites ou pour discuter de nos idées. De plus, on utilise aussi tous ce qui est réseau social pour communiquer et aussi s'envoyer des fichiers (images, texte et autres). De plus, on a créé le logo de notre jeu à l'aide du logiciel « Krita ». Pour confectionner notre jeu on utilisera Godot, mais aussi sans doute un éditeur graphique tel que « Blenders ». Tout ceci en langage C#. Enfin, nous allons utiliser l'application « Trello » qui nous permet de nous organiser et de nous fixer des objectifs au fur et à mesure du temps. Enfin, nous utiliserons Overleaf pour faire nos rapport de soutenance, en plus du rapport de projet et du cahier des charges. Pour ce qui est de mettre en commun notre projet, nous utiliserons bien sûr Github.

2.4 Opérationnel

Nous avons prévu aucun coût pour le développement et nous n'avons pas prévu pour l'instant de commercialiser notre jeu.

2.5 Interface graphique

Le but est que le joueur ne se fatigue pas à chercher à comprendre comment tout fonctionne. Que ce soit les menus ou les fonctionnalités, tout doit être compréhensible dès le premier regard. Pour cela, tout doit être simple et bien positionné.

Pour commencer, parlons de l'écran de titre : nous auront accès à différentes proposition, qui seront de : créer une nouvelle partie, charger la dernière sauvegarde, quitter, mais aussi les paramètres, que ce soit le volume sonore, de la musique ou des bruitages, ou encore les contrôles. Il y aura également un menu pause, dans lequel le joueur pourra également accéder aux paramètres, quitter le jeu, et voir les informations le concernant, comme son équipement par exemple.

2.6 Ambiance

Comme dit précédemment, nous recherchons une ambiance dans le style de Don't Starve, donc assez intrigante mais intéressante. Pour la créer, nous comptons jouer sur l'ambiance sonore dans un premier temps, avec une bande son assez mystérieuse, pour plonger le joueur dans le jeu, sans le déconcentrer.

Et cette ambiance évoluera pour être plus épique lorsque viendra le temps d'affronter un boss. Bien entendu, les graphismes iront également dans ce sens, avec un environnement mystérieux, et un personnage à l'apparence peu rassuré, et l'utilisation de couleurs plutôt ternes, par exemple.

3 Organisation du Projet

3.1 Répartition des tâches

En ce qui concerne les taches de Yann, elles ont été répartie entre Alice et Baptiste le temps que Yann ne revienne.

TABLE 1 – Répartition des tâches

Tâches	Enzo	Baptiste	Yann	Alice
IA			X	
Generation de map		X		
Interface graphique (menu , terrain , etc..)				X
Bandé-son				X
Interations machine/joueur	X			
Développement du jeu (stage, niveaux,leveling, etc...)		X	X	X
Site	X			

3.2 Avancement du Projet

TABLE 2 – Avancement du Projet

Timing	1ère Soutenance	2nd Soutenance	Soutenance finale
Création et conception entitée	15-30%	80%	100%
Interface graphique (menu , terrain , etc..)	20-30%	65-75%	100%
Interactions machine/joueur	25-35%	60-80%	100%
Développement du jeu (stage, niveaux ,leveling , etc...)	25-30%	70-80%	100%

Pour accéder à une version plus avancée de la répartition des tâches ainsi que du planning et pour observer l'avancement du projet :

<https://trello.com/invite/b/Yhhf0G5B/5c20bca70e7d53d51b55bf66hello-world>

4 Présentation général du projet

4.1 Site web - Enzo

Pour la réalisation du site web sachant que je n'ai jamais codé de site web, je me suis fait aider par un tutoriel qui présente comment coder un site morceau par morceau.



Le jeu

Le personnages et les monstres

Installation

Bienvenue sur le site web du jeu "Projet.exe".

Voici la présentation des membres du projet.



Pour commencer, je m'appelle Alice Cariou. Pour me présenter simplement, j'ai toujours été curieuse et intéressée par des choses variées, dont l'informatique. Pourtant, je n'ai envisagé de faire mes études dans ce domaine que très tard, car d'autres domaines, tels que la musique me sont également très cher.



Je m'appelle Baptiste Perrier, j'ai 18 ans et je suis étudiant en première année dans une école d'ingénieur en informatique. J'ai toujours été intéressé par tout ce qui touchait aux sciences, que ce soit les mathématiques, la physique mais surtout à la logique que ça soit de l'électricité, de l'algorithmique ou tout autre domaine d'application.



Je m'appelle Enzo Deloey, je redouble ma 1^{re} année d'EPITA, mais je dois principalement m'occuper du site internet pour le projet. Je n'avais jamais codé de site web avant c'est pourquoi je voulais me dedans pour apprendre comme un site web était coder puis car ça me servir à l'avenir de savoir en coder un. Je dois aussi m'occuper des interactions du personnage avec son environnement.

-Site Web-

J'utilise donc visual studio code pour mon site. Avec pour landing page une présentation du groupe de projet ainsi que une bref description du site..

J'ai par la suite mis le logo du projet, puis créé le menu qui va pouvoir diriger l'utilisateur vers différentes parties du site. N'ayant pas trop d'idées, je me suis inspiré du site internet du jeu très connu "League of Legends". Le menu est donc composé de 3 parties, "le jeu" qui à pour but d'expliquer au joueur comment jouer au jeu tout simplement. Ensuite, il y aura "Le personnage et les monstres", cette partie va expliquer et détailler chaque monstre du jeu qui va être implémenté par

la suite.

Pour le personnage je vais expliquer son "Lore" autrement dit son histoire dans notre jeu ainsi que ses statistique et son style de jeu pour que le joueur puisse comprendre comment jouer correctement. Dernièrement, il y a une partie pour l'installation et la désinstallation du jeu qui accompagné d'un texte expliquant clairement comment installer et désinstaller le jeu. Ainsi que un lien pour télécharger les cahiers des charges et le rapport de soutenance

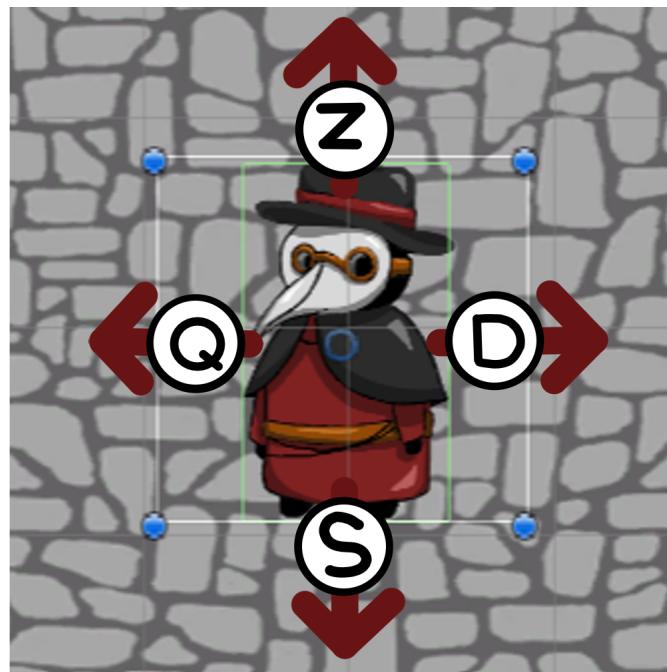
Pour une partie du css du site je me suis fais aider par un framework nommé Bulma qui est très simple à prendre en main et qui m'a permis de combler ma lacune en terme de style afin de rendre le site plus jolie plus facilement.

4.2 Joueur - Enzo

A l'origine, c'était Yann qui avait commencé à s'occuper de cette partie. Il a donc entreprit de générer un Player. Nous étions encore sous Godot, et comme c'était sa première utilisation d'un tel moteur, il a fallu qu'il se renseigne sur internet. Avec l'aide de la documentation fournie directement sur le site de Godot, il avait commencé à apprendre le fonctionnement d'un Player et il avait ainsi démarré sa création.

Il a créé le Player comme étant un "KinematicBody2D", ce qui permet de considérer le Player comme un corps pouvant se déplacer sur une surface plane et pouvant interagir avec son environnement. Je lui ai ajouté une zone de collision qui superposent à l'image l'affichant afin qu'il puisse détecter et agir lors d'une collision.

Ensuite, il avait géré les déplacements du joueur : pour contrôler le personnage il lui a donc ajouté un script simpliste de conditions le faisant bouger lorsque les touches Z,Q,S,D (Haut, Gauche,Bas, Droite) sont pressées. Ses déplacement sont gérés par des vecteurs qui dépendent de variables, sa vitesse peut donc être modifié par d'autres classes.



-Déplacement du joueur-

Pour le joueur j'ai donc décomposé cette partie en plein de sous parties pour que cela soit plus simple à réaliser. Pour commencer, on aura la partie des statistiques du joueur, comme la mana et la vie. Ensuite on aura en fonction du choix de l'arme qui sera détaillé plus bas par Alice, différents scripts associés au joueur qui lui permettront d'attaquer à distance ou au corps-à-corps.

4.2.1 Les statistiques

Pour cette partie là j'utilise donc 2 statistiques majeures qui sont le mana et la vie. Quand une partie sera lancée on initialisera la mana et la vie à leur maximum grâce à 2 variables maxMana et maxHealth qui seront bien sur prédéterminées à 100 points de mana et 6 points de vie, et à partir de cela on a pu définir plein de fonctions qui utiles pour le jeu, comme par exemple la possibilité de se soigner en faisant attention bien sur à ne pas dépasser le maximum autorisé, que Alice a réalisé.

Il y a aussi une fonction de gain de mana et une autre qui renvoie un booléen pour dire si le joueur est autorisé à pouvoir tirer et on utilisera donc ce booléen plus tard. Cette partie là était plutôt simple à réaliser et j'ai bien aimé la faire car celle-ci ne demander pas trop de connaissance poussées sur unity et j'ai donc pu la réaliser sans trop m'aider de tutoriels déjà réalisés sur youtube par exemple. Je n'avais que à penser à toutes les petites méthode utile pour que le personnage puisse faire tout ce que je veux sans trop de problèmes.

4.2.2 L'attaque à distance et les projectiles

L'attaque à distance de notre joueur ne sera possible que dans un seul cas : c'est quand le joueur aura choisi le bâton comme arme principale. J'ai donc créé un script qui va s'occuper de tout ça. Premièrement, il nous falloir une variable qui sera la force du projectile qui va nous permettre de le déplacer, de plus il va nous falloir un GameObject qui sera notre projectile.

Ensuite à chaque frame nous allons regarder si le clic gauche de notre souris est enfoncé et si il a assez de mana pour executer l'action de tirer un projectile, nous pouvons savoir cela avec la fonction décrite avant qui nous dit si le montant de mana restant au joueur est suffisant pour pouvoir tirer.

Si ces conditions sont remplis nous allons pouvoir récupérer la position de la souris sur l'écran et l'utiliser pour créer un vecteur 2 entre la souris et la position du joueur et grâce au rigidbody2D du projectile nous allons pouvoir lui donner une force qui déplacera le projectile dans la bonne direction. A la fin de ce script nous appellerons une méthode qui fait partie des statistique du joueur pour enlever 1 point de mana au joueur.

Bien sur nous n'allons pas laisser le joueur sans défense si il se retrouve à court de mana. Si le joueur essaye donc d'attaquer avec son bâton alors qu'il n'a plus de mana une attaque au corps-à-corps basique ce déclenchera dans la direction de la souris mais elle sera bien évidement plus faible que l'épée car dans le cas contraire ça n'aurais pas beaucoup de sens par rapport à l'épée.

Revenons à notre projectile car il nous reste des points à vérifier pour que tout ce déroule bien. Première chose à vérifier avec nos projectiles c'est qu'il ne dérive pas à l'infini ce qui pourrais réduire les performances du PC si le joueur en tire (vraiment) beaucoup.

Pour cela nous allons utiliser un Collider2D que nous allons attacher à tous nos projectiles et nous allons regarder si celui-ci rentre en collision avec quelle chose d'autre que le joueur lui-même, ça seraient un peu bête de détruire le projectile si il touche le joueur car le projectile par du joueur, donc si il rentre en contacte avec n'importe quoi d'autre que le joueur nous allons regarder si cette chose comporte un script nommé Enemyrecievedamage (qui se charge de gérer la vie des mobs).

Si oui nous allons donc lancer la méthode DealDamage qui prend en paramètre les dégâts que nous voulons infliger au monstres dans notre cas les projectiles seront lancés que avec le bâton donc ils infligeront 2 de dégâts. Par la suite nous savons que le projectile est rentré en contacte avec quelque chose et avons vérifier si c'était un monstre il nous reste plus qu'a détruire le GameObject.

4.3 Interface - Alice/Baptiste

Je commencerai par préciser que je n'avais jamais pris part à ce genre de projets auparavant. Il me restait donc tout à apprendre et la majorité des problèmes que j'ai rencontrés en début de projet étaient liés à mon manque d'expérience. Il est important de rappeler que jusqu'à la première soutenance, nous utilisions Godot. J'ai eu beaucoup de mal à faire fonctionner des choses qui fondamentalement étaient simples, le moindre changement amenait des problèmes que je ne maîtrisais pas. C'est pourquoi le passage sous unity m'a été très bénéfique. J'ai trouvé unity bien plus abordable, et avec plus de documentation, ce qui le rendait plus adapté à notre projet. Comme je l'avais prévu, bien que j'ai dû tout recommencer de zéro pour cette partie, j'ai gagné énormément de temps pour la suite du projet.

4.3.1 Les menus principaux- Alice

La première chose dont je me suis occupée est les menus. Ma problématique était alors de les rendre les plus facilement utilisables par l'utilisateur, tout en faisant en sorte qu'ils restent agréables à regarder. J'ai commencé par le menu principal, qui n'a pas été très difficile à mettre en place. Il met simplement en avant le logo et le nom du jeu, avec des boutons pour lancer une nouvelle partie, charger la partie en cours, accéder au menu settings et un dernier pour quitter. Cela a été instructif et m'a permis de me familiariser avec unity pour la suite.



-Main Menu-

Il me restait encore beaucoup de choses à comprendre sur le fonctionnement des HUD, et j'ai bloqué sur des choses assez basiques, mais que je ne connaissais tout simplement pas. A cause de cela, les éléments n'étaient jamais là où je pensais qu'ils seraient, lors des changements de scènes par exemple.

Après m'être renseignée correctement, tout semblait aller, mais un bug est apparu, et les éléments et les canvas semblaient changer de taille et d'emplacement de manière aléatoire, étant parfois entièrement retournés, ne laissant qu'une croix rouge à leur emplacement. Cela rendait le tout tout simplement inutilisable. A cause de cela, nous ne pouvions même pas build le projet sous peine de tout recasser, et de tout devoir remettre manuellement, ce qui a été une perte de temps considérable, en particulier entre les 2 premières soutenances.

Suite à de longues recherches, aucune explication n'a pu être trouvée pour ce problème. J'ai trouvé un moyen de faire fonctionner les choses, simplement en changeant un paramètre pour forcer les choses, mais cela n'explique en rien pour quelle raison tous les éléments bougeaient de cette manière. Mais au moins tout fonctionne.



-Paused Menu-

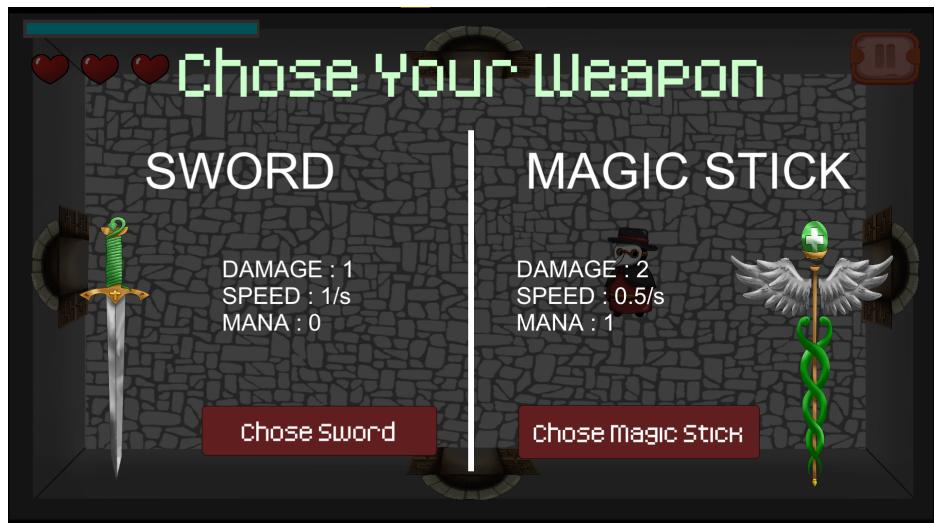
Mis à part ce bug qui a été mon problème principal dans ce projet, car rien ne pouvait fonctionner si il persistait, j'ai rencontré quelques difficultés, comme avec le bouton pause : je pensais que forcer la timescale à 0 était suffisant mais bien que nous ne pouvions plus bouger, les ennemis, eux, le pouvait. Mais ce problème a été assez rapidement réglé : j'ai créé une simple variable booléenne, si le jeu était en pause

ou non, que j'ai mis en condition aux mouvements des ennemis dans leur FixedUpdate. Mis à part tout cela, je n'ai pas rencontré d'autre problèmes dans les menus game over et settings.

4.3.2 Le choix de l'arme - Alice

Pour finir, j'ai créé un menu permettant au joueur de choisir son arme en début de partie. Il ne s'ouvre qu'une fois, lorsque le bouton "new game" est cliqué. Il donne le choix entre nos deux armes : une épée et un bâton magique, et affiche leurs caractéristiques. Sachant que le joueur a 6 vies et 100 points de mana : l'épée inflige 1 point de dégâts aux ennemis, peut attaquer une fois par seconde et ne consomme pas de mana. Le bâton magique, lui, inflige 2 points de dégâts, peut porter une attaque une fois toutes les 2 secondes, et consomme 1 point de mana à chaque attaque, sachant qu'il récupère 3 mana tous les 3 ennemis tués.

Dans l'éventualité où le joueur n'aurait plus de mana et pas de potions, il pourrait tout de même attaquer avec le bâton au corps à corps, en faisant 1 point de dégât avec la même vitesse d'attaque qu'auparavant. La seule "difficulté" que j'ai rencontré avec ce menu a été de pauser le jeu, mais cela n'avait rien de différent de ce que j'avais rencontré avec le bouton pause donc cela a été vite réglé.



-Choose Your Weapon Menu-

4.3.3 Mini Map - Baptiste

Pour la Mini map je dois simplement récupérer la matrice que j'ai générée au début de la partie et afficher en haut à droite de l'écran une carte avec des carré gris pour représenter les salles avec un symbole de coffre en plus si il s'agit d'une salle des coffres, un cranes si il s'agit de la salle de boss ainsi qu'un point blanc pour savoir où est le joueur

4.4 Graphismes - Alice

4.4.1 Le personnage - Alice

La première chose que j'ai créée pour ce jeu a été le personnage. En réalité c'est même l'idée du jeu elle-même qui est venue du design de ce petit médecin du moyen-âge, qui porte donc tout son équipement : masque, chapeau, cape, tunique etc. J'ai décidé de le faire assez peu détaillé pour ne pas avoir à détailler trop d'éléments dans les graphismes par la suite, ce qui m'aurait fait perdre beaucoup de temps.

Dans les faits, il est évident que j'ai passé bien plus de temps que ce que j'avais initialement prévu dans les graphismes, mais j'ai fait en sorte que cela n'influe pas sur le reste du travail que j'avais à faire. Juste après j'ai pu dessiner le logo du jeu, qui représente le même personnage, simplement avec plus de détails.



-personnage-

Pour rappel, le but du jeu est de voyager chronologiquement dans différentes époques pour mettre fin à la pandémie mondiale qui y fait rage. A commencer par la peste noire, au moyen-âge, puis le choléra, au XIXe siècle, et enfin le covid de nos jours, avec toujours un environnement correspondant à la bonne époque.

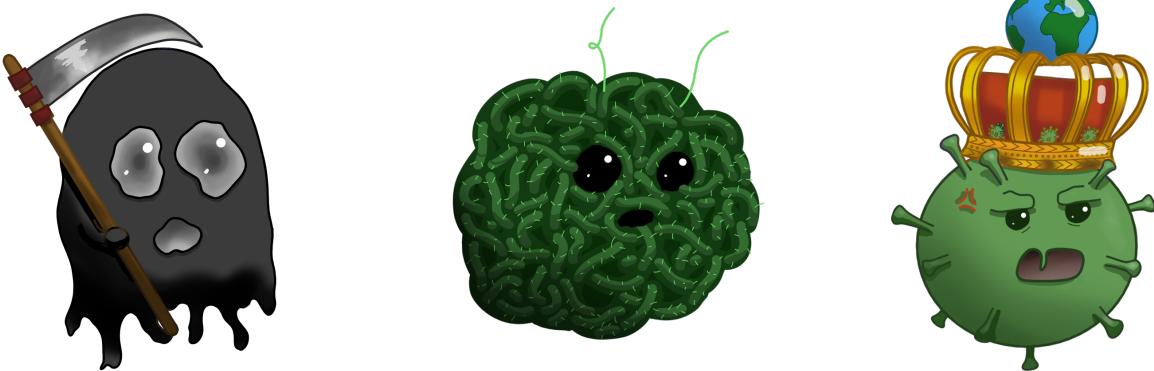
4.4.2 Les boss - Alice

Logiquement, la prochaine étape a été de créer les boss. J'ai commencé par le boss final, le covid, tout simplement parce que c'est celui

pour lequel j'avais le plus d'idées : j'ai utilisé à la fois la signification de son nom et ce à quoi il ressemble réellement, qui sont en fait lié : corona signifie "couronne", à cause de son apparence, que j'ai représenté. En plus de cela, je lui ai ajouté une réelle couronne sur la tête, qui appuie son importance. En bonus, sa couronne est surmontée d'une planète car je pense qu'on peut dire que cette maladie domine le monde.

Ensuite vient la peste noire, qui a été assez simple à représenter, bien que moins effrayante et impressionnante que ce que j'avais initialement prévu : un fantôme et une faux, deux éléments ayant un fort lien avec la mort, pour représenter la maladie ayant fait le plus de victimes au cours de l'histoire.

Pour finir, le choléra, qui est, contrairement aux autres, une infection causée par une bactérie. C'est donc un ensemble de bactéries désorganisées.



][0.2cm] -Boss-

4.4.3 Les environnements - Alice

Suite à cela, il a fallu créer les environnements des différents niveaux. Concernant la disposition de la salle, il y a un problème de perspective entre la salle vue de haut et les personnages, mobs et boss vus de côté, mais le fait est que je n'ai pas trouvé de façon de représenter les salles vues de côté sans compliquer considérablement la mise en place de la suite. De cette façon, chaque mur, et donc chaque porte est visible de la même façon.

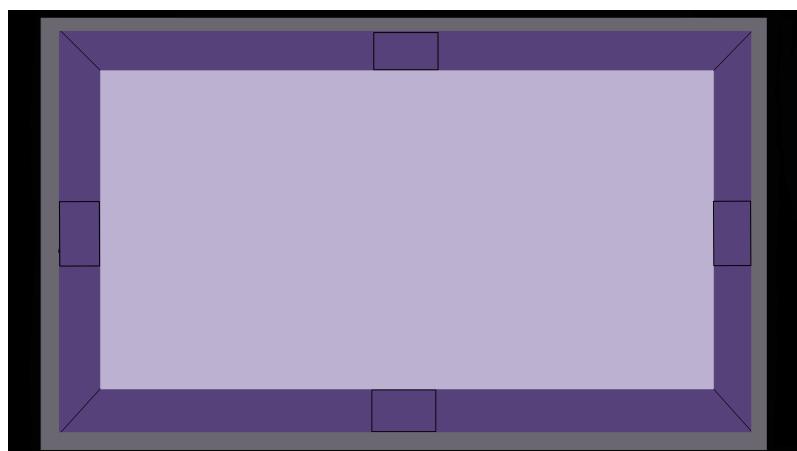
Au début, sous Godot, ce que j'avais prévu pour les environnements n'avait strictement rien à voir avec ce que j'ai finalement fait.

Sur tous les exemples que j'ai pu voir et qui ont pu m'aider, des grilles de 10x10 étaient utilisées, avec les murs tout autour, et une case au milieu pour chaque sortie, que la map soit générée aléatoirement ou non. J'aurai pu tout simplement choisir de faire les salles carrées me direz-vous. Mais plusieurs points m'ont poussé à ne pas le faire :

On avait une grande perte de place : pour que le jeu rende bien j'ai longuement réfléchi à la façon dont je devais représenter une salle : de côté, comme les objets, pour garder une perspective cohérente, ce qui semblait de premier abord être la meilleure solution, ou du dessus, pour que ce soit plus simple à représenter et ne pas ajouter des contraintes supplémentaires. C'était mes 2 possibilités.

Mais après de nombreux croquis, n'étant pas satisfaite des tests pour la première solution, je suis restée sur cette 2e idée, à la Binding of Isaac, qui était celle qui, finalement, avait le plus de chance de bien rendre, tout en rendant probablement la tâche plus facile. Le problème de cette solution étant là aussi la perte de place : les murs prenant tout de même une place importante sur les bords de l'écran, pour que les portes restent correctement visibles. On reste donc avec un problème de perspective par rapport au personnage et aux objets, mais disons que ce problème est secondaire.

Finalement, le modèle de base de la salle, avec les proportions des murs et l'emplacement des portes est le suivant :



-Salle de base-

En ce qui concerne le premier niveau, dans un décors moyenâgeux, donc surtout avec beaucoup de pierres : des pavés pour le sol et des portes à doubles battant assez imposantes.

Ensuite, pour le deuxième niveau, au 19e siècle, ce sera un environnement plutôt steampunk. Un plancher pour le sol, avec des murs abîmés et des tuyaux envahissant les murs. En ce qui concerne les portes, elles restent assez sobres, toujours dans le même style et avec un hublot.

Pour finir, le dernier niveau, plus moderne, se déroule dans un hôpital. C'est donc très blanc, avec du carrelage et des murs simples. Pour passer d'une salle à l'autre, le personnage passera cette fois par un ascenseur.

Pour chaque niveau, il a fallu faire deux modèles de portes, un premier lorsqu'elles sont fermées et un deuxième alors qu'elles sont ouvertes, pour différencier lorsque le joueur peut passer ou non.

Vous pourrez voir les différentes salles et les différentes portes en annexe.

4.4.4 Potions et vie - Alice

Passons aux potions et aux points de vie. Pour commencer, pour les potions, j'ai simplement gardé les couleurs que l'on voit en général pour que cela soit plus intuitif à jouer, c'est-à-dire rouge pour la vie et bleu pour le mana. Nous avons donc une fiole de potion de vie ronde, et une fiole de potion de mana avec plus d'angles pour qu'elles ne se ressemblent pas du tout.



-Potion de vie / Potion de mana-

En ce qui concerne les points de vies du joueur, affichés avec la jauge de mana en haut à gauche de l'écran, j'ai choisi de les représenter sous forme de coeurs, en perdant un demi-coeur à chaque dégât. Avec 6 vies, il y aura donc 3 coeurs.

Pour que cela fonctionne, j'ai simplement créé trois modèles de coeurs : vides, à moitié plein et plein. Ensuite, j'ai superposé les images des demi-coeurs sur les coeurs vides et les coeurs pleins sur les coeurs à moitié vides. Lorsque le joueur perd un point de vie et qu'il a un nombre de vies restantes multiple de 2, un cœur plein disparaît. Et lorsque le nombre de vies restantes ne l'est pas, un demi-coeur disparaît. Jusqu'à ce qu'il n'en reste plus, le menu game over est alors déclenché.

Le procédé pour gagner des coeurs est similaire, la fonction modifie les vies uniquement lorsque le nombre de vies est inférieur au nombre maximal, et ajoute un demi cœur lorsque le nombre de vies restantes est divisible par 2, et un cœur plein sinon.



-Coeur Plein / Demi Coeur / Coeur Vide-

4.4.5 Les mobs - Alice

Ensuite, il a fallu créer les mobs. Malheureusement, le temps est venu cruellement à manquer, et je n'ai pas pu en créer autant que j'aurais voulu. J'ai donc gardé mes deux mobs en les déclinant en 3 couleurs pour les 3 niveaux différents. Ce sera moins varié que prévu mais c'est le mieux que j'ai pu faire. J'ai essayé de rester dans le thème du jeu.

J'ai donc décidé de commencer par représenter un acarien, puis un tardigrade, deux espèces microscopiques. Il y aura une version marron et un peu transparente pour rappeler le fantôme, boss du niveau, puis une version verte pour rappeler le design du choléra, puis une version blanche pour le dernier niveau.

4.4.6 les armes - Alice

Comme je l'ai évoqué plus tôt, nous auront deux armes dans ce jeu. Une épée, pour pouvoir attaquer au corps à corps, et un bâton magique pour attaquer à distance.

Pour commencer, pour la base de l'épée, je me suis inspirée d'épée qui dataient de la même époque que notre personnage, c'est à dire du moyen-âge. Pour que le design colle plus à notre jeu, j'ai rajouté une croix sur la garde de l'épée, puis je me suis inspirée de la représentation du serpent s'enroulant autour d'un récipient, logo des pharmacies. La fusée de l'épée est donc le corps du serpent, sa tête arrivant au niveau du pommeau.



-Epée / Bâton Magique-

Ensuite, en ce qui concerne le bâton magique, j'ai pensé m'inspirer du caducée d'Hermès, qui a été utilisé pour représenter le secret médical, pour aller jusqu'à être le symbole de la médecine elle-même. J'ai donc représenté un bâton entouré de deux serpents et surmonté d'une paire d'ailes.

Cela reste un bâton magique, donc j'ai représenté un orbe flottant au dessus du bâton en lui-même, qui est un élément fréquemment présent dans la représentation des bâtons magiques.

4.4.7 autre - Alice

Enfin, il a fallu réaliser les coffres, qui n'ont rien de particulier, mis à part une petite croix sur le devant pour rester dans le thème du jeu. J'ai simplement chercher à rester sur quelque chose de simple, car ce n'est pas un élément très important du jeu, il ne doit donc pas vraiment attirer l'attention.

Malgré tout, je n'ai pas pu absolument tout dessiner, et il a fallu que je prenne des assets pour le menu et le bouton pause pour me faire gagner un peu de temps, qui m'était nécessaire pour avancer sur d'autres choses.

4.5 Audio

4.5.1 La bande-son - Alice

En ce qui concerne la musique du jeu, c'est moi qui devais initialement m'en occuper. Malheureusement, j'avais sous-estimé la charge de travail que les autres parties dont je suis responsable représentaient et je n'ai pas de connaissance ou de pratique en création de musique à l'aide de logiciels. Un ami a proposé de nous aider sur ce point.

Nous auront donc une musique pour le menu de démarrage, plutôt calme, peut-être un peu inquiétante, dans le but de mettre le joueur dans l'ambiance, une autre que pour le menu game over, accompagnant la déception du joueur, une musique pour chacun des trois niveaux, dans laquelle on ressens l'ambiance de chaque environnement, et enfin, une musique de boss, épique et impressionnante.

4.5.2 Les bruitages - Baptiste

Pour la plus grosse partie des bruitages j'ai simplement recherché en ligne des banques gratuite de bruitage en ligne. Sur lesquelles j'ai recherché des bruitages pour les monstres et le joueur.

J'ai trouvé des choses intéressantes, mais je n'ai pas pu trouver tout ce qui m'intéressais et parfois certain bruitage ne correspondais pas à l'ambiance voulu pour le jeu. Par exemple pour la peste noire, représentée par un fantôme, j'ai trouvé des bruitages intéressants, mais souvent trop effrayants pour coller à l'apparence de ce boss, qui inspire plus la sympathie que la terreur. J'ai donc été contraint de devoir les créer moi même.



-Audacity-

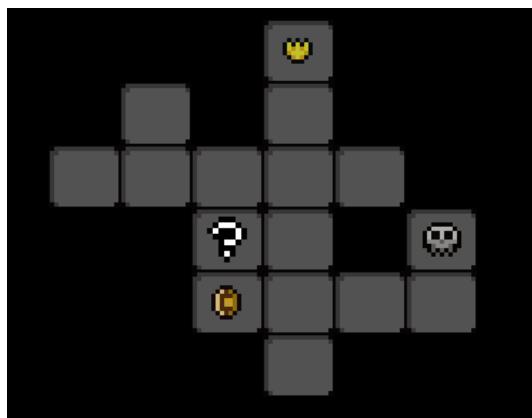
Pour créer ces bruitages je me suis servi du logiciel Audacity afin d'enregistrer des extraits et de les modifier pour créer l'ambiance et la signature sonore attendue.

J'ai donc demandé à des gens autour de moi de faire des bruits qui leur semblaient cohérents par rapport au monstre que je leur montrais et je les ai enregistré. Ensuite je modifiais l'extrait pour obtenir quelque chose ressemblant moins à une voix ou à quelque chose d'humain.

4.6 Génération - Baptiste

Lorsque j'ai commencé la partie de la génération de la carte j'avais une vague idée de la méthode que je souhaitais utiliser. Mais, ne sachant comment j'allais appliquer cette méthode, j'ai eu besoin de commencer mon travail par une grosse phase de recherche afin de me renseigner sur les moyens utilisés pour la génération des cartes dans le domaine des rogue-like.

Au cours de ces recherches j'ai pu trouver de nombreuses choses intéressantes mais surtout un site détaillant avec précision comment marchait la génération de la carte dans Biding of Isaac ainsi qu'une démonstration de cette génération. Il ne me restait donc plus qu'à trouver comment traduire cela en code et comment le mettre en œuvre de manière efficace.



-Génération-

Pour ce faire j'ai donc décidé d'utiliser un matrice de 10 par 10 que j'initialise remplie de 0. Ensuite je tirais un nombre au hasard entre 2 et 20 pour savoir à l'avance le nombre de salles qu'il allait falloir générer.

Après cela je choisissais une place parmi les 100 disponibles afin de désigner l'endroit par où j'allais commencer ma génération. Il fallait ensuite choisir un nombre entre 0 et 13 (nombres de combinaisons différentes de portes pour une salle contenant 1 à 4 portes).

Puis il suffisait de choisir une nouvelle salle pour chaque porte que possédait ma salle et réitérer cela jusqu'à ce que le nombre de salle voulu soit obtenu.

Cette méthode avait malheureusement un problème majeur. Il était possible d'arriver dans une situation où il était impossible de continuer

à générer car toutes les portes de la salle dans laquelle on se situait donnait sur des places où il y avait déjà une salle.

Pour pallier cela, il y avait plusieurs solutions. Premièrement, j'ai pensé à simplement arrêter la génération dans le cas de figure présenté précédemment mais cela été peu envisageable car on avait donc une porte qui donnait sur une salle n'en possédant pas et cela risquait de poser des problèmes lors de l'implémentation de mon algorithmes dans le jeu.

En effet, on aurait eu la possibilité d'aller d'une salle vers l'autre mais pas de revenir en arrière. Pour cela on aurait pu changer les portes de la salle concernée mais cela ne convenait pas car ça revenait à tirer un nombre au hasard pour ne finalement pas s'en servir.

La deuxième solution à laquelle j'ai pensé a été de recommencer la génération à chaque fois que j'arrivais dans un situation de blocage. Mais cette solution avait de forte chance de rendre la génération de la carte très longue et de parfois générer des boucles infinies.

J'ai donc finalement choisi de revoir la façon dont j'allais me servir de ma matrice en utilisant les connaissances récemment acquises en programmation orientée objet. Pour cela, j'ai décidé de garder ma matrice de 10 par 10 mais de remplacer son contenu. J'ai donc créé plusieurs objets Room (dont je parlerais plus loin).

Pour initialiser ma matrice je me sert de ses objets dont la valeur par défaut est null. Je commence ensuite ma génération comme pour la version précédente en choisissant mon nombre de salle et un emplacement pour la salle de départ. Le plus gros changement a lieu au niveau de la gestion des portes et des voisins. À chaque fois que je place une salle, je la définis comme ayant toutes ses portes fermées et j'ajoute tous les emplacements libres qui lui sont collés dans une liste qui me permet de savoir où je peux placer la salle suivante.

```

public static Room[,] Init_list()//Initialisation l'étage vide
{
    Room[,] stage = new Room[10, 10]
    {
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null},
        {null,null,null,null,null,null,null,null,null,null}
    };
    return stage;
}

```

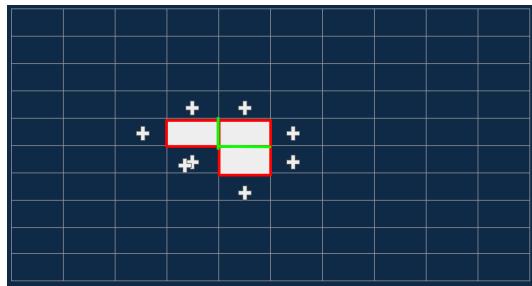
-Matrice Initiale-

Je mets à jour ses portes si elle a un voisin de manière à ce que si il a un voisin la porte concernée devient ouverte et j'en fait de même pour sa voisine. Pour finir je la retire de la liste des emplacements disponibles et je trie une nouvelle salle aléatoire dans cette même liste.

Cette méthode fait qu'une même salle peut apparaître plusieurs fois dans la liste des emplacements disponibles mais cela n'est pas réellement un problème car cela force juste un paterne. Je dois juste faire un boucle qui supprime toutes les itérations de cette salle lorsqu'elle est sélectionnée dans ma liste.

L'image ci-dessous représente plutôt bien ce à quoi pourrais ressembler l'algorithme de génération, même s'il ne représente pas tous les cas possibles.

Ici, toutes les croix représentent un emplacement disponible, les barres rouges des portes fermées et lorsque cette barre est verte, cela signifie que la porte est ouverte. Une case blanche est donc une salle. Lorsque que je veux mettre une nouvelle salle, je prends alors l'un des emplacements marqué avec une croix et je remplace la croix par une salle avec ses portes. Dans le cas où deux croix se superposent cela signifie que cette emplacement apparaît deux fois dans la liste des emplacements libres



-Explication Génération-

En plus de générer un nombre donné de salles, je dois choisir le type de ses salles parmi trois types (Normales, Boss et Coffres).

Pour la salle de boss, il s'agit simplement de la dernière salle que je choisis, une fois que j'ai créé cette salle je dois juste sélectionner le boss en fonction du niveau dans lequel je me trouve.

Pour savoir si un salle normal où il y a des monstres ou si il s'agit d'une salle des coffres, je choisis au hasard entre l'une ou l'autre avec une probabilité d'une chance sur dix de tomber sur un salle des coffres.

Si je tombe sur une salle des coffres je dois juste y placer un coffre. Ensuite je choisis aléatoirement si il doit contenir une potion de soin ou une potion de mana.

Dans le cas des salles normales le procédé est un peu différent car je leur associe une matrice dans laquelle je vais générer les monstres de cette salle en fonction du niveau dans lequel je me trouve en faisant varier la vitesse d'attaque et de déplacement et je sélectionne aussi un modèle en fonction du niveau.

Pour la salle de départ de chaque niveau je génère une salle normale avec une liste vide de monstre dès le début.

4.7 Ennemis - Baptiste/Enzo

4.7.1 Classe mère IA - Enzo/Baptiste

Au commencement du projet, c'est Yann qui s'occupait de cette partie. A la manière du Player, il a cherché sur internet des tutoriels et de la documentation afin de comprendre et d'être capable de créer une IA.

Il a été surpris par le manque d'information pour créer des classes d'ennemis, la grande majorité des tutoriels ou sites expliquent la création de joueur, scènes, d'environnement, mais très peu d'ennemis. Pour débuter il a choisis de créer un "KinematicBody2D", comme pour le Player, afin qu'il puisse lui aussi agir avec l'environnement et se mouvoir.

Pour les déplacements de l'IA, je me suis inspiré de ce que Enzo a fait dans sa partie pour le déplacement du joueur. Le système de déplacement de l'IA est très rudimentaire, elle se déplace simplement en direction du joueur et effectue en même temps un déplacement dans une autre direction aléatoire pour essayer de donner un air moins artificiel à son déplacement.

En ce qui concerne les dégâts, lorsque les collisionneurs du joueur et du mob entrent en contact on attend une seconde et si après ce délai le joueur est encore en contact avec le mob on lui retire une vie. Lorsqu'un mob est tué, on le supprime simplement ainsi que son collider.

Après avoir fais les dégâts sur le joueur par les monstres il est temps de faire le chemin inverse pour cela on a besoin d'ajouté un script à ces monstres spécialement dédié à la réception des dégâts.

Pour cela nous devons commencer par donner de la vie au monstre qui sera déterminé par la difficulté du jeu et qui marche comme ceci nous prenons une base de 4 points de vie puis nous multiplions ce nombre par 0.5 si la difficulté est facile par 1 pour normal et par 2 pour la difficulté difficile.

Par la suite dans la fonction start nous commençons par donner la vie du monstre comme étant la vie maximale qu'il peut avoir. Il nous reste plus que 2 méthode à ajouter qui serons le fait de recevoir des dégâts et de regarder si le monstre est mort ou non.

Pour la 1^{re} rien de plus simple, on prend en paramètre les dégâts que le monstre doit recevoir et on les soustrais à la vie restante du monstre et on appelle ensuite l'autre fonction pour savoir si le monstre est mort ou non. Si c'est le cas c'est à dire que ses points de vie sont inférieur ou égale alors nous détruisons le monstre.

4.7.2 Boss - Baptiste

Pour les boss, je dois juste créer trois nouveaux objets avec un système de déplacement et des mécanismes de combat différents. Comme les seules choses que ses trois boss ont en commun sont leurs variables, j'ai décidé de ne pas créer de classe mère et simplement créer chaque objet de manière indépendante.

J'ai donc commencé mon travail sur les Boss par une partie de game design afin d'imaginer et de concevoir les combats contre ces boss.

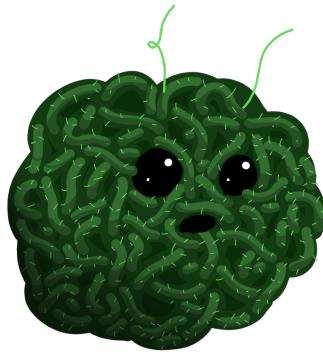
La peste



-la peste noire-

Tout d'abord je me suis occupé du Boss de la peste. Se boss à la particularité de ne pas se déplacer sauf quand il attaque. Lorsqu'il attaque, il se met à vibrer pendant une seconde puis fonce sur la position qu'avait le joueur une seconde avant lui infligeant des dégâts si il le touche.

Le Choléra



me choléra

Le boss suivant est le Choléra. Pour sa part, il se déplace pendant trois secondes vers le joueur avant de s'arrêter une seconde et générer subitement une onde choc qui, si elle touche le joueur, lui inflige un point de dégâts.

La Covid



le covid

Le dernier Boss est la Covid. Comme elle est censée être le boss le plus dur, elle invoque deux autres boss qui ont un peu moins de points de vie qu'en temps normal. Il les ré-invoque au bout de 30 secondes s'ils ont été tués. De plus, elle marque quatres zones au sol pendant une seconde et si le joueur se trouve dans l'une d'elle au bout d'une seconde elle lui inflige des dégâts et elle répète cette attaque au bout de trois secondes.

5 Bilan général et personnel

5.1 Enzo Deloye

De mon côté ce projet c'est quand même mieux passer que l'année dernière en partie grâce à Alice et Baptiste qui m'ont tout les deux "supporté" et aidé. J'ai tout de même finis de prendre en main Unity et je me suis bien familiarisé avec ce logiciel ce qui m'a permis de prendre du plaisir dessus car je ne passe plus tout mon temps à faire des alt-tab pour regarder les tutos que je trouvais sur internet.

Après cela faire le site web était intéressant et je pense que avoir appris le html/css me servira souvent dans mes études je suis donc content de m'en être occupé même si j'ai eu quelque soucis avec celui-ci notamment pour la 2ème soutenance. J'ai eu quand même un soucis majeur avec le site, c'est un manque de créativité et de connaissance sur les sites pour pouvoir le rendre ergonomique et joli à regarder mais j'ai quand même fait tout mon possible pour avoir un rendu qui me satisfasse.

De manière plus général je n'ai pas trouvé ce projet facile à réaliser en partie à cause du trou qu'a laissé Yann et d'une organisation discutable qui n'est pas vraiment mon fort, c'est pour ça que je pense que c'est un point très important que je devrais travailler pour les prochains projets que je vais devoir réaliser dans le futur. De plus une autre chose que je devrais approfondir c'est ma connaissance de git hub car j'ai toujours quelque petits soucis à le faire fonctionner correctement.

5.2 Alice Cariou

En ce qui me concerne, n'ayant jamais réalisé ce genre de projet aupara-vant, c'est l'occasion de découvrir une grande quantité de nouveaux concepts, de nouveaux logiciels...

En bref, c'est une très bonne occasion pour moi pour en apprendre beaucoup dans un domaine qui m'étais jusqu'alors plutôt inconnu. Et en plus de cela, ce projet me permet de mettre à l'oeuvre d'autres de mes passions, telles que le dessin et la musique, tout en pouvant laisser libre cours à ma créativité.

Je pense qu'on peut dire que cela n'a pas été simple. A commencer par ma propre partie qui m'a posé bien plus de problème que ce à quoi

je m'attendais. Bien que j'ai réussi à régler la plupart d'entre eux, il en reste qui persistent et ce n'est pas encore parfait.

Et si l'on ajoute à cela le fait que Yann ne soit plus là, ce qui nous amène à reprendre une partie de ce qu'il aurait dû faire, bien que cela soit très enrichissant, cela reste une charge de travail supplémentaire, qui n'était pas prévue à l'origine. La répartition du travail n'a pas été tout à fait ce à quoi nous nous attendions, mais disons que nous sommes restés motivés à mener notre projet à bien, que tous les membres du groupe soient actifs ou non.

Le changement de plateforme nous a également retardé, et m'a forcé de reprendre à 0 une partie entière de mon travail, tout ce qui concerne les interfaces, et qui ne pouvait pas être récupéré de Godot à Unity.

En conclusion, cela a été une expérience de ce qu'est un projet de groupe, avec ses bons points mais surtout ses erreurs à ne pas reproduire, ce qui nous sera bien utile à l'avenir dans nos futurs projets.

5.3 Baptiste Perrier

Pour moi, ce projet aura été très compliqué. Non pas par sa difficulté mais plus sur le plan du travail en groupe. J'ai eu des difficultés à m'organiser avec le groupe et à me mettre à son rythme.

Ma motivation du début s'est rapidement transformée en lassitude de travailler le projet. Le fait que le groupe n'était pas vraiment soudé a été une cause majeure du manque de motivation et le travail à distance n'a pas été d'une grande aide non plus.

Nous avions de grosses lacunes en organisation et la plupart des objectifs été remplis à la dernière minute et souvent mal fait et pas testé. Et ça c'est beaucoup vu lors des soutenances, les soirs qui les précédait étaient souvent très long parce qu'il fallait tout déboguer car rien ne marchait.

De plus, nous n'avions pas vraiment organisé notre gestion de Git Hub et aucune branch+ n'était jamais à jour ce qui provoquait beaucoup de conflits lorsque nous mergions à la dernière minute avant la soutenance.

Ce que j'ai fait au cours de ce projet reste quelque chose de très

intéressant et j'ai pu acquérir beaucoup de connaissances. J'ai appris à relire le code de quelqu'un et à essayer de me mettre à sa place afin de comprendre ses objectifs.

J'ai pu apprendre à m'adapter et à faire face à des conflits à l'intérieur même d'un groupe. Et même si ça n'a pas vraiment bien marché, ça a été un bon apprentissage et ça devrait m'aider pour de futur projet en groupe.

6 Conclusion

Pour la première soutenance nous avions un peu était pris au dépourvu, la charge de travail nous avait surpris et par manque d'anticipation nous n'avons pas pu rendre un travail satisfaisant. De plus, nous avions eu des difficultés avec nos outils qui étaient à ce moment-là GitLab et Godot.

Nous avions donc plusieurs fonctionnalités mais rien ne les reliait entre elles et nous n'avions donc rien de concret à présenter. Nous avions donc un menu principal, un joueur capable de se déplacer, un début d'IA qui était incapable de réagir correctement avec son environnement ainsi qu'un début d'algorithme de génération de la carte qui n'était lui non plus pas complet.

Nous avions aussi un début de site web et une base solide de model 2d. Il ressortait du ressenti global une envie de bien faire et une grande motivation à l'idée de ce qu'il y avait à apprendre et faire.

Pour la deuxième soutenance nous avions décidé de changer d'outils et de migrer de GitLab et Godot vers GitHub et Unity. Nous avions donc presque tout ce qui concernait le code du jeu à recommencer de zéro. La charge de travail a donc été bien supérieure à ce que nous avions prévu au début du projet mais au vu de notre motivation ça ne nous avait pas apeurés.

Nous avions encore une fois mal organisé notre temps et sommes arrivés à 1 semaine de la semaine de soutenance nous avons appris que nous devions finir ce projet à 3. Il a donc fallu redistribuer le travail de Yann afin d'avoir sa partie qui était nécessaire pour la soutenance.

Nous sommes donc arrivés à cette deuxième soutenance avec encore une fois beaucoup de retard. Nous avons rendu un travail qui aurait été satisfaisant pour une première soutenance. Nous avions une base jouable mais pas réellement un jeu en lui-même.

Depuis la dernière soutenance nos problèmes sont restés presque les mêmes, nous avons toujours un manque évident d'organisation même si nous communiquons plus et que notre Github est plus propre. Nous n'avons toujours pas trouver de moyen pour nous réorganiser assez bien pour compenser le départ d'un membre du groupe.

On s'est de nouveau retrouvé à 1 semaine de la soutenance avec presque tout le travail depuis la dernière soutenance à faire. De plus, n'ayant pas rattrapé notre retard pour la deuxième soutenance, il nous restait encore ce retard à rattraper et notre manque évident d'efficacité ne nous a pas permis de rattraper tout le retard que nous avons pris en changeant de plateforme.

Mais même après toutes ces difficultés, nous avons tout de même obtenu un jeu jouable, avec deux armes et donc deux façon de jouer différentes, ainsi que 3 niveaux différents générés procéduralement et possédant tous un boss différent. Certes, cela reste loin de ce que nous espérions produire à l'origine, mais nous avons tout de même su surmonter les problèmes que nous avons rencontré pour produire quelque chose.

Malgré tout cela nous avons tous beaucoup évolué pendant ce projet et avons tous appris à maîtriser des nouveaux outils. Nous avons appris à travailler dans la précipitation même si cela n'est pas une bonne méthode de travail il est bien de savoir le faire. Le travail réalisé, bien que laborieux, a été intéressant et on a pu passer de bon moment, un peu fatiguant, à avancer sur ce projet.

7 Bibliographie

- Site - Godot Engine
- Site - Krita
- Site - FL Studio
- Site - Piskel
- Site - Rider
- Site - Documentation Bibliothèque UnityEngine
- Site - lasonotheque
- Recherche d'assets
- GitHub du projet
- Site du projet
- Trello du projet
- Audacity
- Site - Explication fonctionnement génération Biding of Isaac
- Overleaf

8 Annexes



portes niveau 1



portes niveau 2



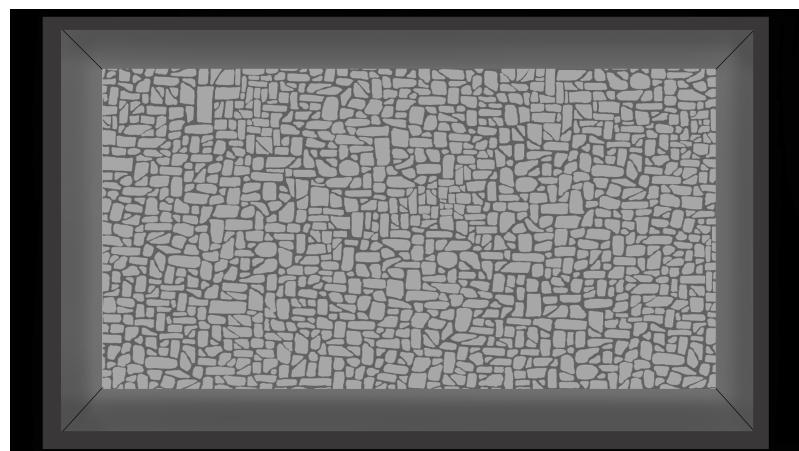
portes niveau 3



mob : acarien



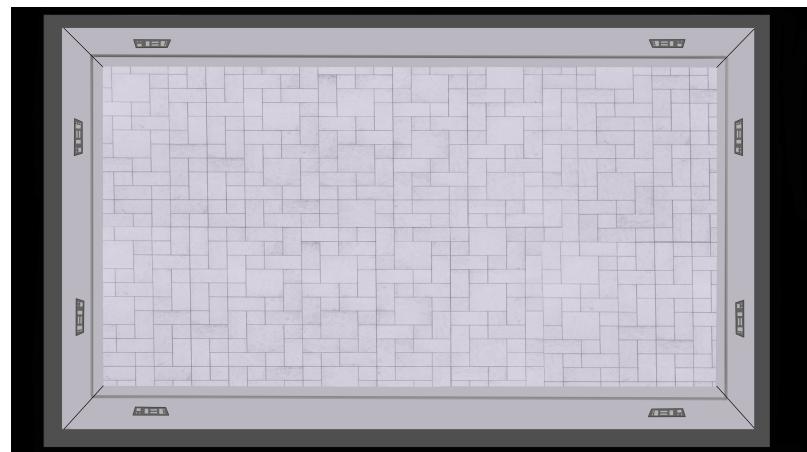
les coffres



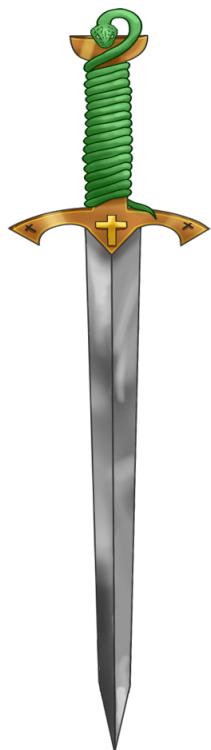
salle 1



salle 2



salle 3



épée / bâton magique



choix d'armes