The Call Of The Goose

Projet de S2

Rapport de projet

Yvon Morice
Souleymane Sentici
Amin Salmi
Marie Legay

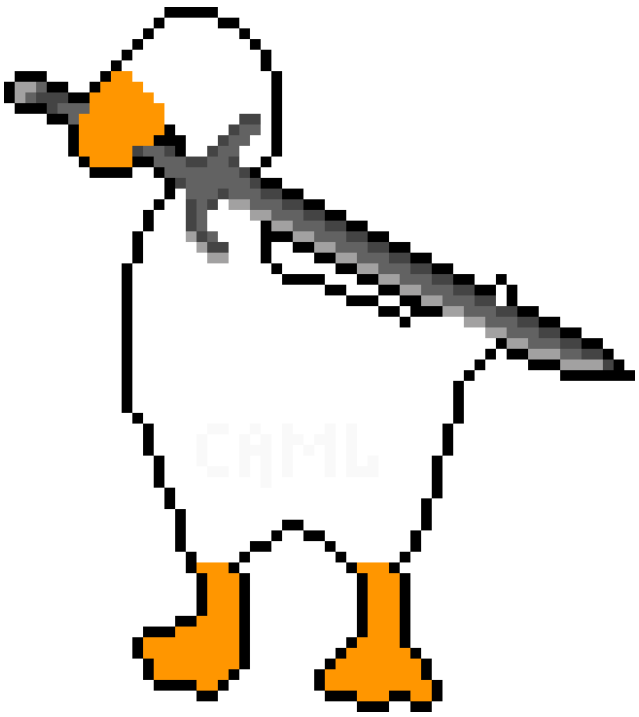


Table des matières

1 Introduction **3**

2 Bibliographie **4**

3 Le cahier des charges **5**
3.1 Le projet 5
3.2 Résumé 5
3.2.1 Présentation du jeu 5
3.2.2 Principe du jeu 5
3.3 Le groupe 6
3.4 Travail 7
3.4.1 Répartition des tâches 7
3.4.2 Prévisions 8

4 Installeur **9**

5 Site Web **10**

6 Les outils utilisés **12**

7 Éléments du jeu **13**
7.1 Menu principal 13
7.2 Menu de pause 15
7.3 Génération des niveaux 16
7.3.1 Génération des niveaux en solo 16
7.3.2 Génération des niveaux en multijoueur 20
7.4 Système de minimap 21
7.5 Personnage joueur 21
7.5.1 Caractéristiques 22
7.5.2 Les objets 23
7.5.3 Déplacements 25
7.5.4 Système d'inventaire 25
7.5.5 Système de dialogue 26
7.5.6 Interface du joueur 27

7.6 Combats 28
7.6.1 Combat en mode solo 28
7.6.2 Combats en mode multijoueur 31
7.7 Ennemis 32
7.7.1 Caractéristiques 32
7.7.2 Déplacements 32
7.7.3 En combat 33
7.8 Magasin et marchand 34
7.8.1 Le magasin en solo 34
7.8.2 Le magasin en multijoueur 35
7.9 Mode Multijoueur 36
7.10 Réalisation artistique 38
7.10.1 Musique 38
7.10.2 Sprites 41

8 Fonctionnalités abandonnées 43
8.1 Les classes jouables 43
8.2 Arbres de compétences 43
8.3 Tutoriel 44

9 Réalisation 45
9.1 Le groupe 45
9.2 Souleymane Sentici 45
9.2.1 Marie Legay 47
9.2.2 Amin salmi 47
9.2.3 Yvon Morice 48

10 Conclusion 50

1 Introduction

Notre objectif durant la dernière période de travail a été de passer d'un état de prototype à un état plus avancé, plus proche d'un jeu.

Ainsi, nous nous principalement concentrés sur le multijoueur, le système de combat ou encore les parties graphiques et sonores du jeu pour cette dernière soutenance.

L'un des principaux intérêts de ce projet était de développer nos connaissances dans le domaine de la programmation ainsi que dans le processus de création de jeu vidéo.

Un autre des intérêt de ce projet était d'apprendre à travailler en groupe sur une longue durée avec des soutenances régulières.

La création d'un jeu vidéo consiste en effet en une expérience inédite pour plusieurs élèves. Les seules expériences similaires étant pour la plupart le TPE ou des projets de classe, généralement bien moins ambitieux. L'un des intérêts de ce projet était donc également d'apprendre à être plus autonomes, à apprendre à présenter un projet ou encore à respecter des délais.

Notre groupe est constitué de 4 membres, tous en D1. Dans l'ordre nous avons, Marie Legay, Yvon Morice (chef de projet), Amin Salmi et enfin Souleymane Sentici.

2 Bibliographie

Génération de niveaux :
— https ://www.youtube.com/watch ?v=WtDXk6uuZO4
— https ://www.youtube.com/playlist ?list=PLX2vGYjWbI0SKsNH 5Rkpxvxr1dPE0Lw8F
— https ://www.youtube.com/watch ?v=hk6cUanSfXQ
— https ://www.youtube.com/playlist ?list=PLBIb_auVtBwA-qr2-WnWX0LjZXkqKu5Aj
— http ://weblog.jamisbuck.org/2011/2/7/maze-generation-algorithm-recap

Menu / Interfaces :
— https ://www.youtube.com/watch ?v=zc8ac_qUXQY

Graphismes :
— https ://blog.studiominiboss.com/pixelart
— https ://www.piskelapp.com

Site web :
— bootstrapstudio.io/

Installateur :
— https ://jrsoftware.org/isdl.php

3 Le cahier des charges

3.1 Le projet

3.2 Résumé

Notre projet est un jeu de type rogue-like, ce type de jeu se caractérise par une création de donjon aléatoire, ainsi qu'une mort définitive du joueur en cas de game over (réinitialisation de toutes les statistiques et bonus accumulés au cours de la partie). Il ne faut pas confondre ce type de jeu avec son homologue le rogue-lite, qui se caractérise lui par une mort non définitive, en effet, dans ce type de jeu, toutes ou une partie des caractéristiques sont conservées à la mort du personnage.

Plusieurs jeux du genre du rogue-like sont sortis avant. l'un d'entre eux "Rogue" ayant donné son nom au genre. Des exemples récents de jeux de ce genre pourraient être "Spelunky", "Faster than light" ou encore "Enter the Gungeon".

Pour ce qui est des rogue-lite, on peux citer comme exemples "The Crypt Of The Necrodancer", "Dead Cells" ou bien "The Binding Of Isaac".

Nous nous sommes donc inspirés de ces jeux afin de créer le notre.

3.2.1 Présentation du jeu

— **Genre :** Rogue-like
— **Style :** Action, RPG
— **Plateforme :** PC (Windows)
— **Moteur :** Unity 2019.3.0f5

3.2.2 Principe du jeu

Au lancement de la partie :

— Choix du type de partie (Solo ou Multijoueur)
— Sélection de la classe du personnage
— Lancement du jeu

Lors de la partie :
— Génération du niveau en cours de manière aléatoire
— Le jeu commence à l'étage 0 (sur 5 étages)
— Il y a un boss par étage ainsi qu'un changement d'ambiance aussi bien visuelle que sonore
— Objectif du joueur : atteindre le 5ème étage et vaincre le boss

3.3 Le groupe

Legay Marie, passionnée de jeux vidéos et de musique.

Morice Yvon, passionné de jeux vidéos qui a pour vocation de devenir développeur, si possible dans le domaine du jeu vidéo.

Salmi Amin, passionné de littérature Lovecraftienne, de cuisine et de jeux videos, qui a pour vocation de devenir développeur.

Sentici Souleymane est un passionné de jeux vidéos qui a pour vocation de devenir développeur.

3.4 Travail

3.4.1 Répartition des tâches

Partage des tâches		
Tâche	Responsable	Suppléant
Histoire	Amin	Marie
Ambiance	Amin	Marie
Bande son	Marie	Amin
Character design	Amin	Marie
Animations	Marie	Amin
Dessin 2D	Marie	Amin
Level design	Souleymane	Yvon
Gameplay	Yvon	Souleymane
Tutoriel	Souleymane	Yvon
Système de combat	Marie	Amin
Interactions avec le marchand	Marie	Souleymane
Génération des niveaux	Yvon	Souleymane
Items	Souleymane	Amin
Ennemis	Marie	Yvon
IA des ennemis	Marie	Yvon
Personnages	Souleymane	Amin
Pathfinding	Amin	Souleymane
Menu/Menu de pause	Amin	Yvon
Multijoueur	Marie	Yvon
Site web	Yvon	Souleymance

3.4.2 Prévisions

Tâche	Soutenance		
	1ère	2nde	3ème
Bande son	0	25	100
Character design	50	70	100
Animations	10	50	100
Dessin 2D	10	50	100
Level design	70	90	100
Gameplay	40	80	100
Tutoriel	10	60	100
Système de combat	30	60	100
Interactions avec le marchand	0	50	100
Génération des niveaux	40	70	100
Items	10	50	100
Ennemis	30	50	100
IA des ennemis	10	60	100
Personnages	20	60	100
Pathfinding	0	50	100
Menu/Menu de pause	50	75	100
Multijoueur	20	60	100
Site web	30	50	100

4 Installeur

(Réalisation : Yvon)

L'installeur que l'on a créé pour notre projet a été généré au travers de l'outil Inno Setup Compiler. Cet outil génère dans un premier temps un script d'installation à partir des données fournies. Ces données étant constituées principalement des fichiers à inclure, du nom du logiciel, de la version de ce dernier, de l'installeur ou encore du fichier de licence.

Nous avons ensuite modifié le script généré par cet outil de façon à ce qu'il corresponde plus à nos besoin. Ainsi, nous avons rajouté la possibilité de désinstaller notre jeu sans passer par le Panneau de Configuration ;

L'utilisateur passera par plusieurs étapes lors de l'installation. Dans un premier temps, l'utilisateur peux choisir entre l'anglais et le français pour l'installeur.

Il devra ensuite accepter les termes d'une licence Creative Commons (CC BY-NC-SA-4.0). Celle-ci stipule que les utilisateurs ont le droit de réutiliser les ressources du projet de manière non commerciale tant que les auteurs sont crédités et que le contenu dérivé utilise une licence du meme type.

l'installeur demande si l'on veux installer le jeu pour tous les comptes de l'ordinateur, ce qui nécessite d'être administrateur, ou seulement pour le compte actuel.

Si l'on installe le jeu pour tous les comptes, celui-ci sera installé par défaut dans le dossier "Program Files" de Windows, sinon il sera installé dans le dossier "User/AppData/Local/Programs".

Ensuite, l'utilisateur peux choisir de modifier l'endroit où le logiciel sera installé. il pourra par la suite choisir d'avoir un raccourci sur son bureau.

Enfin, l'installeur procédera à l'installation. Une fois cette dernière terminée, l'installateur montrera un guide d'utilisation du jeu puis l'utilisateur pourra choisir de lancer le jeu à la fermeture de l'installeur ou de simplement fermer l'installeur.

5 Site Web

(Réalisation : Yvon)

Le site est composé de quatres pages différentes. La première d'entre elle, la page d'acceuil, est constituée de plusieurs captures du jeu ainsi que d'une brève description du jeu. On y trouve également le logo de notre groupe. Enfin cette page contient également une barre de navigation contenant des liens vers les autres pages du site. Cette barre de navigation se retrouvera sur toutes les autres pages.

Une seconde page contient une brève présentation de chaque membre du groupe. La troisième page contient quant à elle les liens de téléchargements du projet ainsi qu'un guide d'installation. Enfin, la dernière page contient des liens de téléchargements pour les différents rapports rendus lors des soutenances ainsi que le cahier des charges rendu lors de la première soutenance.

Notre site est hébergé sur GitHub Pages, nous avons choisi cette solution car, notre site étant statique, GitHub Pages était l'une des solutions les plus simples se présentant à nous. En effet, cela nous permet de faire héberger notre site gratuitement et de le modifier de façon assez simple et rapide.

Pour le réaliser, nous nous sommes servis du logiciel Bootstrap Studio, un logiciel WYSIWYG (What You See Is What You Get), utilisable gratuitement gràce au GitHub Student Developper Pack.

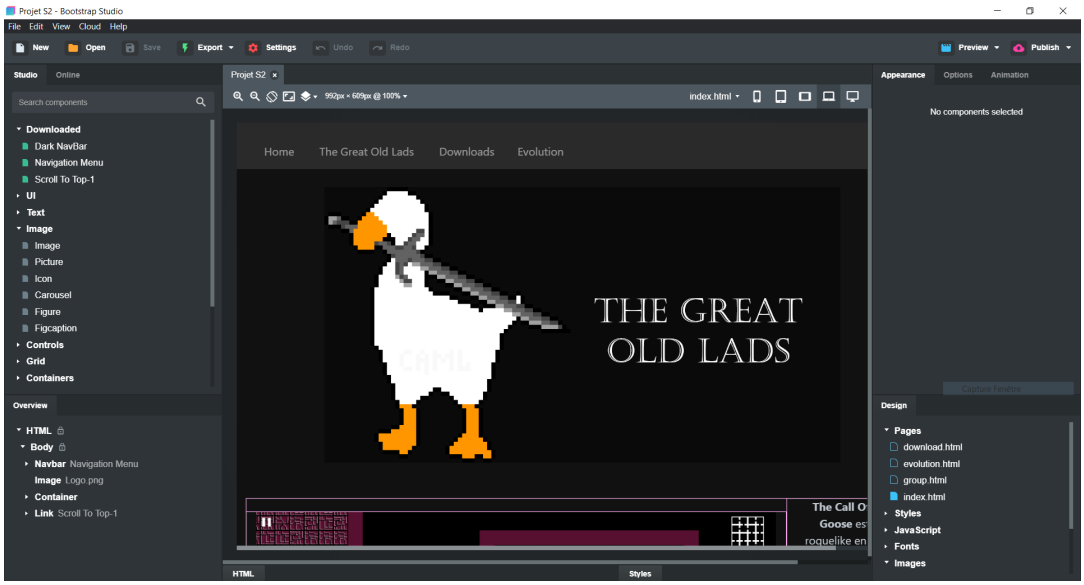


FIGURE 1 – Le logiciel Bootstrap Studio

6 Les outils utilisés

Afin de développer notre jeu, nous avons utilisé plusieurs outils. Ceux-ci étant :

1. Le moteur et framework Unity, qui nous as permis de réaliser la grande majorité de notre jeu.

2. Le logiciel de gestion de versions Git, nous avons chois d'utiliser cet outil car il s'agit d'un outil souvent utilisé dans le milieu de l'informatique, que ce soit de manière professionnelle ou non. Les différents TPs réalisés au cours de l'année nous ont également permis de commencer à apprendre à utiliser cet outil.

3. Le framework de Photon Unity Networking, nous ayant permis de réaliser la partie réseau et multijoueur de notre jeu.

4. L'IDE Rider. Cet environnement de développement ayant utilisé au cours de nos différents TPs était celui auquel nous étions le plus habitués et constituait donc un choix logique pour réaliser notre projet.

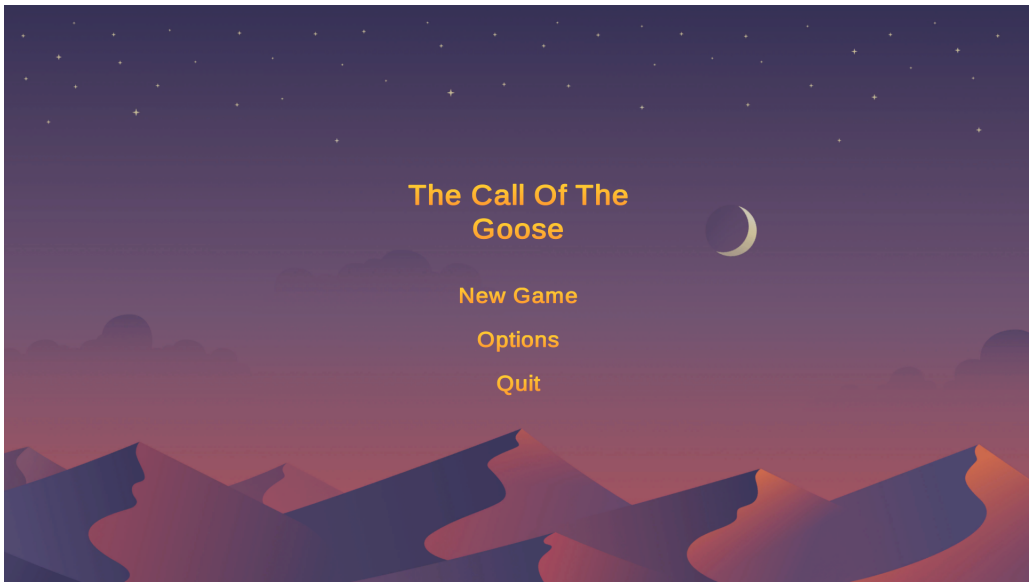
5. Le site Overleaf, nous ayant permis de réaliser nos rapports.

6. Le logiciel Piskel, nous ayant permis de réaliser les différents sprites de notre jeu.

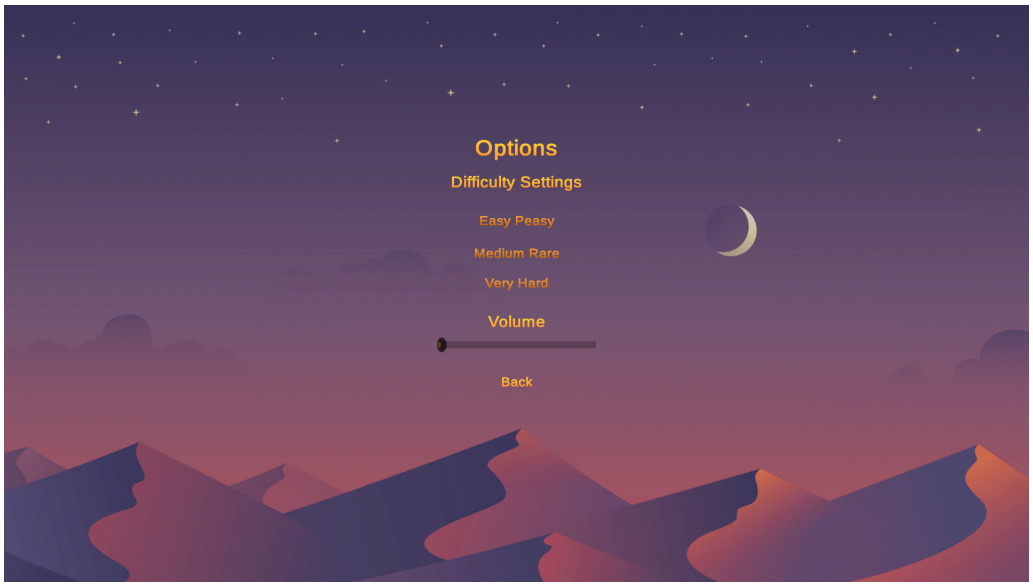
7 Éléments du jeu

7.1 Menu principal

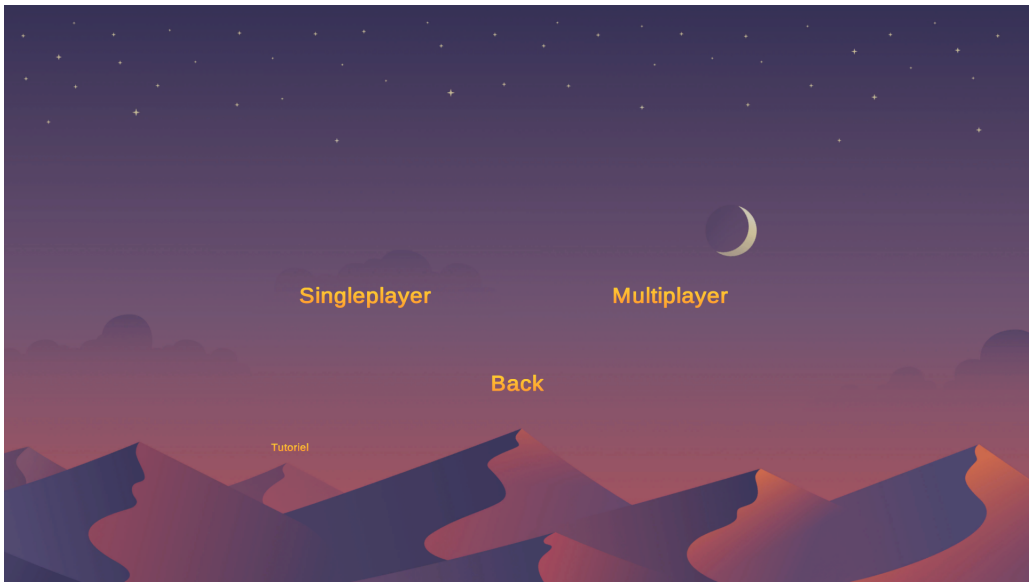
(Réalisation : Amin)



La toute première chose que nous avons réalisé était le menu principal, il consiste en un menu où l'on choisit entre lancer une partie, ce qui amènera le joueur à un écran où il pourra sélectionner le mode de jeu, modifier les options du jeu ou quitter le jeu.



Le menu d'option permet de choisir entre 3 niveaux de difficultés et changer le volume sonore du jeu.



Le menu où l'on choisit le mode de jeu permet de choisir entre le mode solo ou le mode multijoueur.

Le bouton singleplayer lance directement une partie, tandis que le bouton multiplayer ouvre un dernier sous menu où le joueur doit choisir un pseudo, puis peut créer ou rejoindre des salles pour jouer en ligne avec d'autres personnes.

7.2 Menu de pause

(Réalisation : Yvon)

Ce menu permet au joueur de retourner au jeu ou de simplement mettre en pause le jeu. Il est constitué de deux boutons, l'un permettant de reprendre la partie, l'autre permettant de retourner au menu.

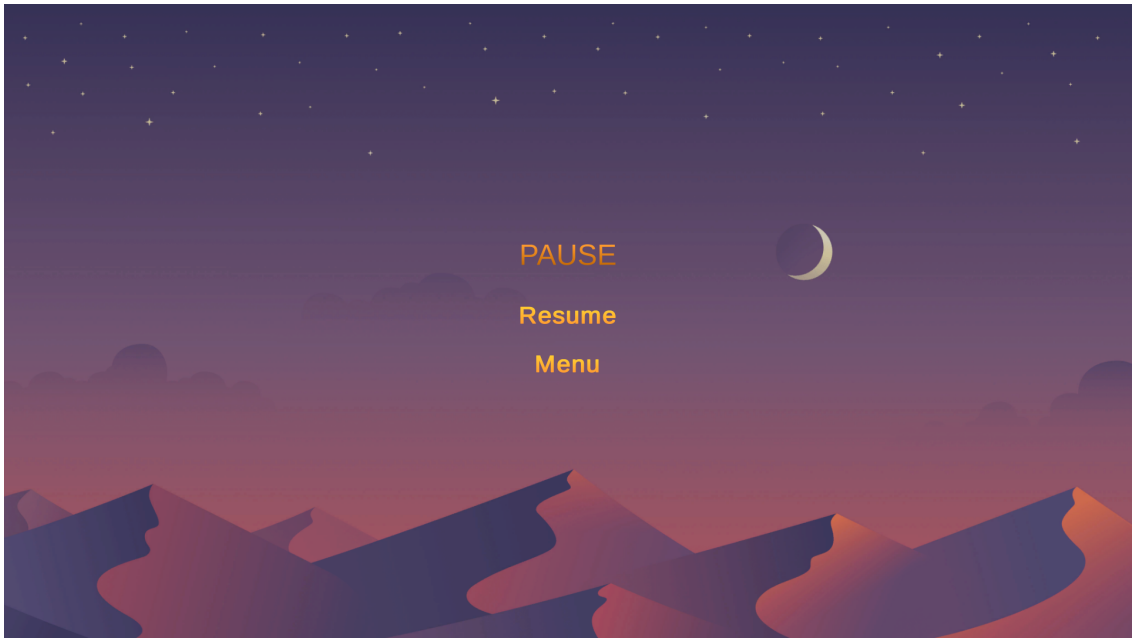


FIGURE 2 – Le menu de pause

7.3 Génération des niveaux

7.3.1 Génération des niveaux en solo

(Réalisation : Yvon)

À chaque étage, un nouveau niveau est généré. Celui-ci est créé en utilisant une variante de l'algorithme des arbres binaires. Celui-ci permet de créer des labyrinthes parfaits sans utiliser de mémoire.

En effet, l'algorithme d'origine n'utilise pas d'état pour fonctionner et peux théoriquement générer des labyrinthes sans limite de tailles, Chaque cellule étant générée indépendamment des autres. Il s'agit également du générateur de labyrinthe le plus simple et le plus rapide.

L'un de ses désavantages étant son biais. En effet, les labyrinthes générés au travers de cet algorithme auront tendance à avoir de long couloirs selon la direction utilisée pour la génération. Ces labyrinthes peuvent également être représentés sous forme de structures de données d'arbre binaires. L'exemple ci-dessous considère la salle en haut à gauche comme étant le point de départ.

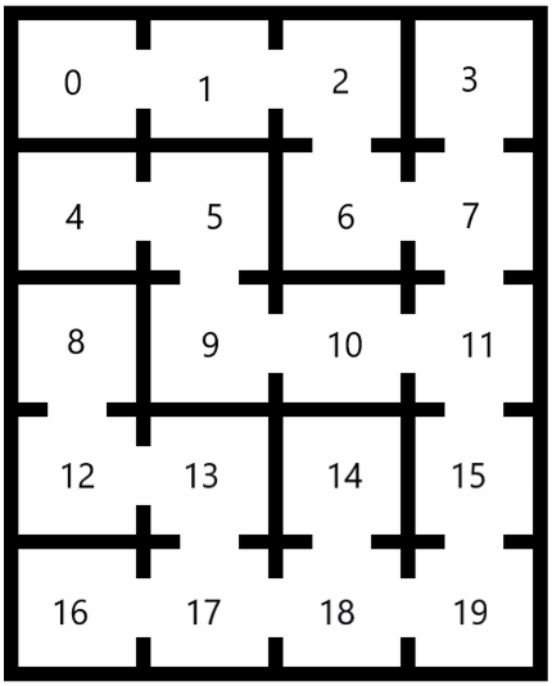


FIGURE 3 – Un exemple de labyrinthe généré par cet algorithme

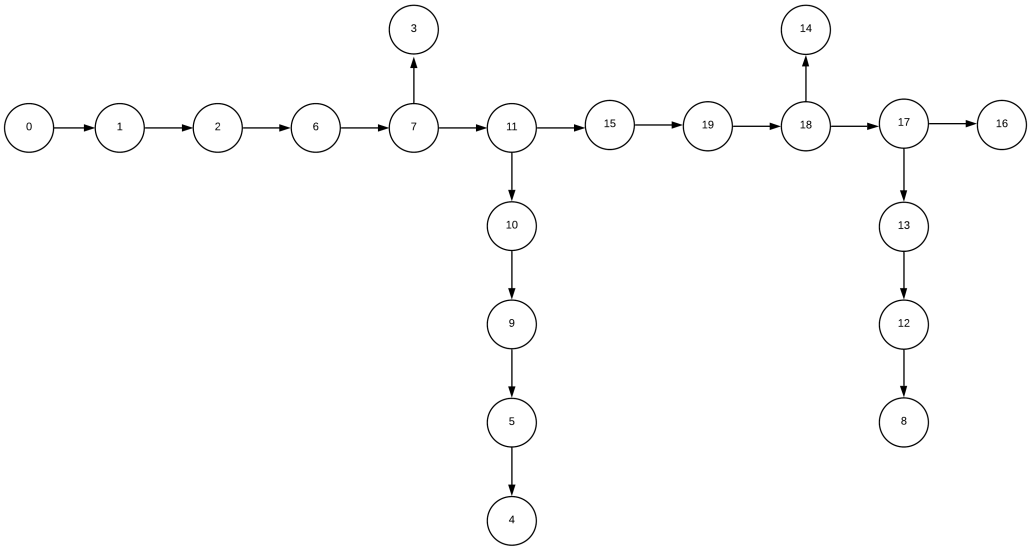


FIGURE 4 – L'arbre binaire associé

La version de cet algorithme utilisée dans notre projet utilise quant à elle de la mémoire. En effet, avant de générer la salle, on vérifie si celle-ci se trouve sur la droite ou en bas du labyrinthe.

Si celle-ci se trouve tout à droite du labyrinthe, on la force à avoir une entrée vers le bas. Si elle se trouve tout en bas, son entrée se trouvera vers la droite. Enfin si celle-ci se trouve dans le coin en bas à droite du labyrinthe elle aura une entrée vers la gauche et vers le haut (créées par les salles précédemment citées) mais aucune vers la droite ou le bas. Les autres salles auront quant à elles une chance sur deux d'avoir une entrée vers la droite ou vers le bas.

Pour générer notre niveau, on commence par la salle en haut à gauche du labyrinthe et, selon les principes données ci-dessus, on ouvre une entrée vers la droite ou vers le bas. On passe ensuite à la salle immédiatement à sa droite et on répète le processus jusqu'à avoir atteint la salle la plus à droite de la ligne. Pour celle-ci on ouvre uniquement une entrée vers le bas puis on passe à la ligne suivante.

On répète ce même procédé jusqu'à la dernière ligne. Une fois cette dernière atteinte, chaque salle hormis la dernière de la salle sera ouverte sur la droite. La dernière salle se trouvant dans le coin en bas à droite, elle aura uniquement des entrées sur la gauche et vers le haut.

Ainsi, chaque salle sera obligatoirement reliée soit au bord droit, soit au bas du labyrinthe, la salle en bas en droite reliant ces deux bords. Les couloirs auront également tendance à relier plusieurs salles sur la même ligne ou la même colonne.

FIGURE 5 – Un exemple de niveau généré

Une fois toutes les salles créées, on en choisit trois au hasard en s'assurant que aucune d'entre elles n'ait déjà été choisie. Ces trois salles serviront de points d'apparition pour le ou les joueurs, le boss ainsi que le marchant de l'étage.

Enfin, on générera un nombre aléatoire de monstres dans chaque salle en dehors des 3 choisies auparavant selon sa taille qui sera assignée aléatoirement lors de leur création. On aura ainsi entre 0 et 3 (exclu) monstres dans les plus petites salles, entre 2 et 4 (exclu) monstre dans celles de taille moyenne et enfin, entre 3 et 5 (exclu) monstres dans les salles les plus grandes.

La génération des niveaux en solo avait été commencée pour la première soutenance et a été finalisée pour la dernière soutenance avec l'ajout du magasin, du boss et plusieurs corrections de bugs.

7.3.2 Génération des niveaux en multijoueur

(Réalisation : Yvon)

La génération des niveaux lors d'une partie en multijoueur se déroule de manière légèrement différente. En effet, on ne peux pas demander aux deux client (il y a deux joueurs lors d'une partie en multijoueur), de tout deux générer un niveau en meme temps, les chances que ces niveaux soient les mêmes étant extrêmement faibles.

Ainsi, nous avons choisi de ne faire générer le niveau que chez le client maître (celui qui a créé la salle), puis d'envoyer les caractéristiques des salles au deuxième client, ces caractéristiques étant composées de la taille de la salle ainsi que ses entrées (représentées par 4 booléens) afin de pouvoir le recréer à partir de ces données.

Ensuite, le client master instancie plusieurs entités au travers du réseau (via Photon). Ces entités sont les différents monstres, le boss, ainsi que le marchand.

Enfin, le client master génère son joueur dans la salle prévue à cet effet, puis envoie une commande à l'autre client au travers d'une RPC (Remote Procedure Call) lui disant de générer son propre joueur à la position voulue.

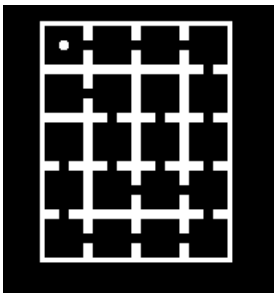
La génération de niveaux en multijoueur fut quant à elle entièrement développée durant la période séparant la deuxième soutenance de la dernière soutenance.

7.4 Système de minimap

(Réalisation : Yvon)

Nous avons également implémenté un système de minimap permettant au joueur de se repérer plus facilement dans le niveau. Celle-ci n'est disponible qu'en mode solo, afin de donner une difficulté supplémentaire en multijoueur, le but de ce mode étant d'être le premier à arriver au dernier étage sans mourir.

La minimap est constituée de deux éléments principaux, les icônes de salles, et celle du joueur. Les icônes de salles sont mis à jour à chaque génération de niveau afin de pouvoir s'y repérer, l'icône du joueur est quant à elle mise à jour à chaque frame. Lorsque celui-ci change de salle, son icône ira dans la salle correspondante de la minimap.



La minimap du jeu

7.5 Personnage joueur

7.5.1 Caractéristiques

La classe implémentant le joueur a peu changé au fil des soutenances. Elle était déjà proche de sa forme finale lors de la première soutenance. Le personnage joueur possède de nombreuses caractéristiques, une majorité de ces dernières pouvant évoluer au fil de la partie à l'aide d'objets.

Il possède évidemment des points de vie, mais également du mana et de l'endurance, qui lui permettent de lancer des attaques différentes en combat. Celles-ci sont tout aussi importantes, car ces attaques étant plus puissantes, elles permettent d'économiser les points de vie du joueur en

raccourcissant les combats.

Le joueur possède deux inventaires, l'un contenant des reliques, et l'autre des consommables. Il rencontrera ces deux types d'objets au cours de sa partie et il sera capital pour lui de les gérer correctement afin de survivre et gagner.

Durant la partie, le joueur accumule de l'or, que ce soit en tuant des ennemis ou en trouvant des bourses perdues. Cet or lui permettra d'acheter des objets au magasin du jeu, présent à chaque étage et offrant un choix d'objets aléatoire.

Il possède également une statistique d'attaque, dictant le nombre de dégâts qu'il peut infliger. Celle-ci augmente également grâce à des objets.

Enfin, le joueur a une caractéristique d'armure. Celle-ci diminue les dégâts qu'il reçoit, chaque point d'armure enlevant un point de dégât normalement reçu à chaque attaque et est également affectée par les objets obtenus par le joueur.

7.5.2 Les objets

(Réalisation : Souleymane)

Les sprites des différents objets du jeu

Prévus dès la première soutenance, ils n'ont été finalisés que suite à la deuxième. Ils se déclinent en deux catégories majeures :

1. Les reliques, des objets augmentant certaines statistiques du joueur au cours de la partie.
2. Les consommables, des objets utilisables de manière uniques et permettant au joueur de régénérer ses différentes caractéristiques, comme les points de vie ou le mana.

Nous pensions d'abord créer une banque d'objets pour stocker les différents objets du jeu, mais nous avons fini par opter pour des Scriptable Objects, facilement créables et utilisables depuis Unity.

Les Scriptable Objects Items que nous avons crée permettent de stocker différents caractéristiques que possèdent tous les objets. parmi eux, nous retrouvons :

1. Son nom, qui servira pour son affichage dans les boites de dialogue ou encore dans l'inventaire.
2. Son sprite, qui est celui affiché à l'intérieur de l'inventaire du joueur.
3. Son prix en or, qui sera celui que le joueur devra payer pour cet objet si celui-ci est disponible dans le magasin.
4. Sa description, qui est un court texte décrivant l'utilité de l'objet.

Les objets sont ensuite séparés entre la classe consommable et la classe relique qui diffèrent par leur methode Use(), qui dicte l'action qui surviendra lorsque l'on clique sur l'objet dans l'inventaire.

Tandis que les consommables peuvent être utilisés en cliquant dessus, les reliques permettent de regarder les statistiques qu'elles procurent au joueur.

Ces objets sont très importants pour la survie et la progression du joueur. les consommables lui permettant de régénérer son endurance et son mana.

Les reliques, quant à elles, lui permettent de rester à niveau avec les différents monstres qu'il croisera et d'avoir une chance de se mesurer au boss de l'étage.

7.5.3 Déplacements

Le déplacement du personnage joueur à été implémenté avant la première soutenance, et n'a pas nécessité de changements majeurs depuis.

Le jeu enregistre les inputs du joueur sur les touches de déplacements (ici ZQXD ainsi que les flèches), et fais avancer le joueur à une vitesse prédéfinie dans cette direction à l'aide de la méthode FixedUpdate(). Celle-ci permet, contrairement à la méthode Update() classique, de ne pas tenir compte des performances du jeu ou de l'ordinateur sur lequel celui-ci est utilisé, et de faire se déplacer le joueur à vitesse constante.

7.5.4 Système d'inventaire

(Réalisation : Yvon et Souleymane)

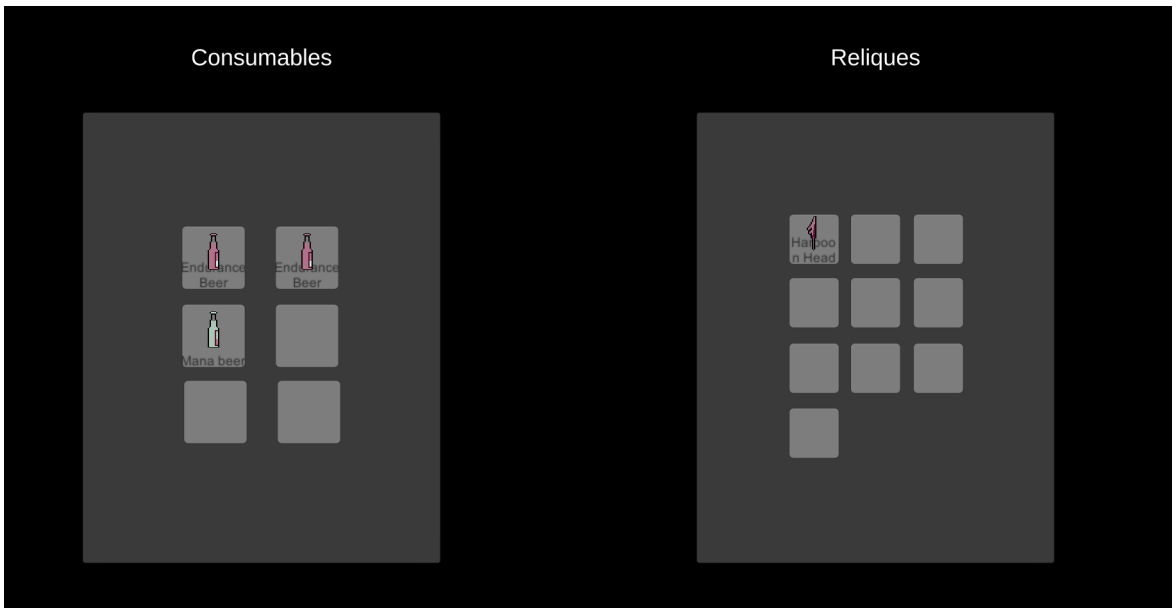
Nous avons implémenté le système d'inventaire, permettant au joueur de stocker des objets de plusieurs types qu'il pourra trouver durant la partie. Les deux types d'objets étant les reliques et les consommables. Celui-ci à été commencé et finalisé pour la dernière soutenance.

Le joueur peut l'ouvrir en appuyant sur la touche 'E', ce qui lui ouvre une interface séparée en 2. Cet deux parties contiennent chacune une in-

dication du type d'objet en haut (Consommables ou Reliques) ainsi qu'un panneau contenant plusieurs slots dans lesquels se trouvent les objets. La partie consommable ne peux contenir que jusqu'à six objets, tandis que la partie reliques peut en contenir jusqu'à dix.

D'un côté, le joueur peut voir les consommables qu'il possède actuellement, et en utiliser en cliquant dessus avec la souris. Ceci consommera l'objet, et écrira son effet dans le système de dialogue du jeu.

De l'autre, les différentes reliques en possession du joueur sont affichées, et il peut vérifier les statistiques qu'elles procurent en cliquant dessus, ces informations s'affichant dans le système de dialogue.



L'inventaire du jeu

7.5.5 Système de dialogue

Le système de dialogue est composé de 3 éléments. Le nom de l'objet décrit, sa descritption ainsi que d'un bouton 'continue' permettant de

passer à la suite de la description, si cette dernière est finie, le dialogue se ferme.

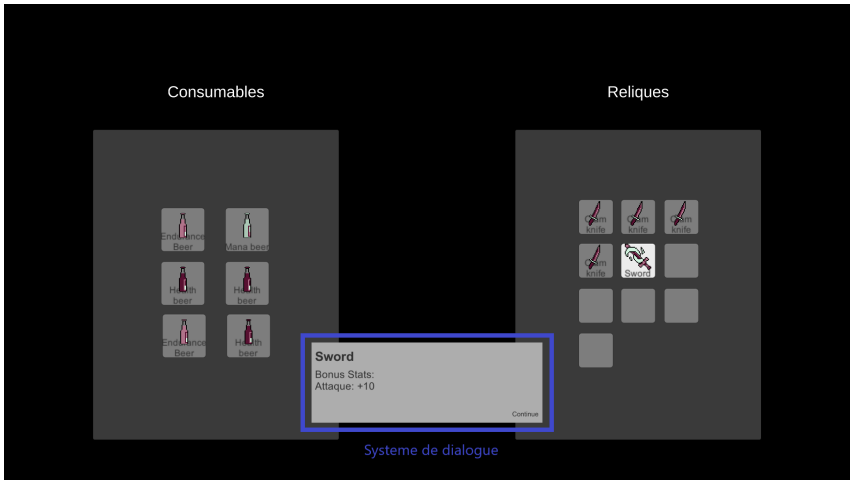


FIGURE 6 – Le système de dialogue

7.5.6 Interface du joueur

(Réalisation : Yvon)

L'interface du joueur est composé de deux éléments principaux, un bouton de pause ainsi que la minimap dont on a expliqué le fonctionnement plus haut. Le bouton de pause permet d'accéder au menu de pause.

Celle-ci se présente de la manière décrite par l'image ci-dessous.

FIGURE 7 – L'interface du joueur

7.6 Combats

7.6.1 Combat en mode solo

(Réalisation : Souleymanne et Yvon)

Pour la dernière soutenance, nous avons finalisé le système de combat du jeu, qui n'était pas fonctionnel auparavant. Un combat a lieu lorsqu'un ennemi rentre en contact avec le joueur, ou lors des combats de boss prévus à la fin des étages.

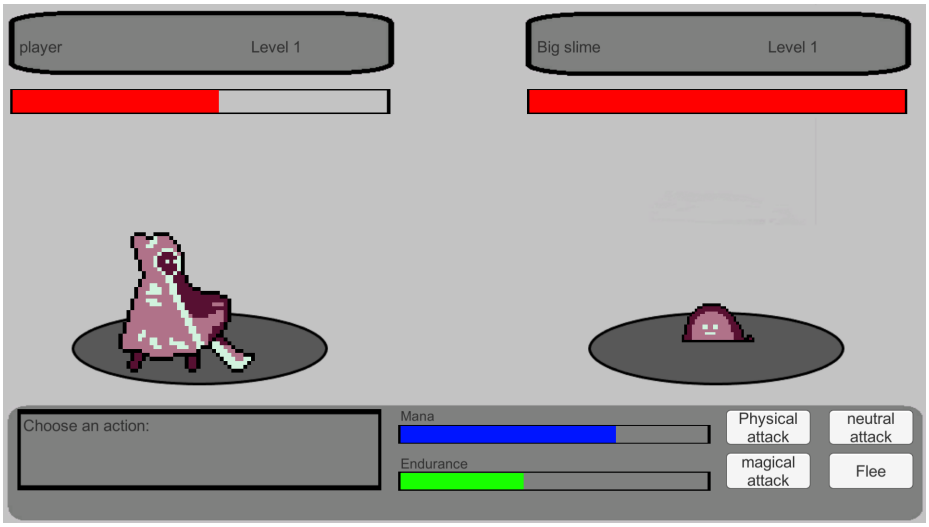
Lors d'une rencontre, un ennemi est choisi aléatoirement parmi ceux présents dans le jeu. Le joueur a alors le choix parmi 4 options :

1. Une attaque neutre, qui ne coûte aucune ressource mais qui n'est pas très efficace.
2. Une attaque magique, plus efficace mais coûtant de la mana a chaque utilisation.

3. Une attaque physique, aussi efficace que la magique mais coûtant cette fois-ci de l'endurance.
4. La fuite, si le joueur est en mauvaise posture ou possède déjà tout ce dont il a besoin, il peut éviter les combats et préserver ses ressources pour le boss en fuyant les rencontres. Celle-ci est a double tranchant. En effet, elle fais tout de même disparaître l'ennemi rencontré, ce qui prive potentiellement le joueur d'objets qu'il aurait pu obtenir en se battant. Il a également une chance sur 5 d'échouer à s'enfuir si il combat un ennemi classique et échouera à tout les cas si il s'agit d'un boss. Il se fera attaquer immédiatement après par l'ennemi qu'il est en train de combattre en cas d'échec

Le joueur et l'ennemi échangent ensuite des coups jusqu'a la mort de l'un d'entre eux. En cas de victoire, le joueur aura une chance de trouver un consommable et une relique aléatoirement sur le cadavre de l'ennemi, avec un taux de 50% et 40% respectivement. Il recevra également de l'or, en quantité variable selon le type d'ennemi vaincu.

En cas de mort cependant, le jeu étant un Rogue-Like, le joueur aura un écran de Game Over et sera renvoyé à l'écran titre du jeu, et devra recommencer depuis le début.



L'interface de combat

L'interface du système de combat est composé de plusieurs éléments. Il est notamment composé d'une barre contenant un dialogue dans lequel s'affiche différentes indications pour le joueur, telles que s'il s'agit de son tour, s'il a réussi à fuir ou le type d'attaque qu'il a effectué.

Cette barre contient également les barres de mana et d'endurance, permettant au joueur de savoir combien il lui en reste, ainsi que les boutons lui permettant d'agir, à savoir les boutons d'attaques et de fuite.

En haut de l'interface, il y a également 2 rectangles, l'un contient les informations de l'ennemi, à savoir son nom (typiquement : slime, big slime, etc), son niveau ainsi que sa barre de vie, l'autre rectangle contient lui les informations du joueur (également son nom, son niveau et sa barre de vie).

Enfin, l'écran de combat contient deux cercles sur lesquels se positionnent les deux combattants dont les animations changent lorsque le combat commence.

Pour implémenter la notion de tours, nous avons utilisé une énumérations d'états, appelés States, qui nous a permis d'effectuer des actions

différentes selon la situation du combat.
Le combat commence avec l'état Start, qui actualise l'interface, avec les points de vie actuels des entités et place les bons sprites à leur place.

Viens ensuite l'état Tour du joueur, qui autorise le joueur à appuyer sur un des boutons d'actions. Suite à cette action, le jeu vérifie que l'ennemi n'est pas mort, et si ce n'est pas le cas, c'est à son tour de jouer, et ceci se répète jusqu'a la mort de l'une des entités, ou la fuite du joueur.

Selon l'état en cours, une méthode spécifique appellera une certaine coroutine. Celle-ci permet d'effectuer une action tout en rajoutant des pauses de quelques secondes. Ceci est particulièrement pratique pour gérer les messages de la boîte de dialogue du combat, pour que le joueur ait le temps de lire les messages affichés avant que d'autres ne les remplacent.

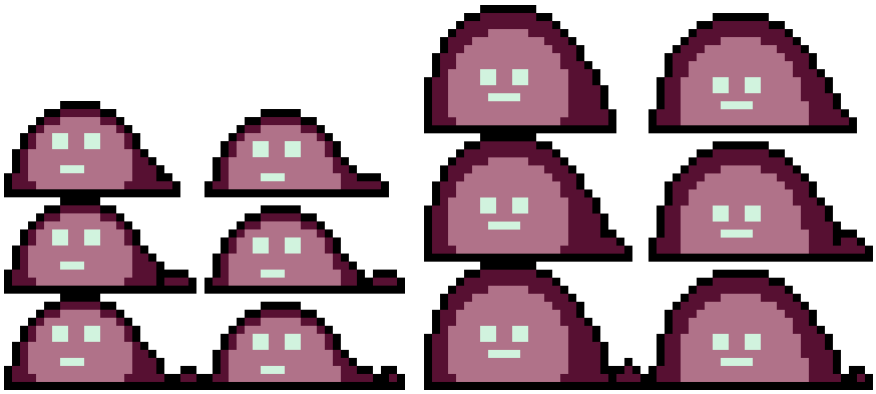
7.6.2 Combats en mode multijoueur

En mode multijoueur, chaque joueur a son propre écran de combat et un joueur ne peux pas rejoindre le combat d'un autre. De plus, les deux joueurs ne peuvent pas se combattre l'un l'autre.

Cela a été fait de sorte que les joueurs se concentrent sur leur but, à savoir arriver le premier au dernier étage et réussir à vaincre le dernier boss. Ainsi, le fait de faire combattre les joueurs pourrait rendre l'un de ces deux joueurs désavantagé.

De plus, les couloirs étant assez étroits, cela déclencherait des combats incessants.

7.7 Ennemis



7.7.1 Caractéristiques

Depuis le début du projet, les ennemis de Call of the Goose n'ont pas été très différents du joueur en terme de statistiques. La principale différence étant l'absence de mana et d'endurance, s'expliquant par l'incapacité de ces monstres à pratiquer la magie ainsi qu'une forme physique discutable.

Les ennemis ne possèdent également pas d'inventaire ou d'or. En effet, ceux-ci ont été substitués par les les différents objets que ces monstres possèdent sur eux, et que le joueur obtiendra potentiellement s'il parvient à le tuer.

7.7.2 Déplacements

Les déplacements des ennemis ont été implémentés entre la première et la deuxième soutenance, et n'ont nécessité des changements que lors de l'intégration du multijoueur.

Contrairement au joueur qui se déplace librement, les ennemis restent sur place la majorité du temps. Cependant, lorsque l'un des joueurs se trouve assez proche, celui-ci se met à se déplacer dans sa direction, dans

l'optique de lancer un combat contre lui.

Pour cela, on détermine à chaque frame quel est le joueur le plus proche d'un ennemi, ainsi que la distance qui les sépare. Si celle-ci est inférieure à une certaine valeur, l'ennemi en question aura son vecteur mouvement transformé en une ligne droite le menant au joueur.

Nous avons décidé de faire en sorte que le joueur soit plus rapide que les monstres qu'il rencontre. Puisque Call of the Goose est un Rogue-Like, le joueur doit peser le pour et le contre dans chaque situation, et décider si de potentiels objets valent le risque de perdre des précieux points de vie.

7.7.3 En combat

Les ennemis en combat se comporte basiquement comme un joueur, pouvant attaquer à chaque tour. Cependant, puisqu'ils n'ont pas de mana ni d'endurance, ils se contentent de faire des attaques neutres, utilisant leur statistique d'attaque comme montant de dégâts. Ceci à été fait pour plusieurs raisons.

Premièrement, nous ne souhaitions pas saturer l'interface de combat avec trop d'informations pour le joueur, les barres utilisées pour la mana et l'endurance étant relativement grandes.

Deuxièmement, nous pensions que cela compliquerait inutilement les combats, puisqu'ils dureront rarement assez longtemps pour que l'une des barres de l'ennemi soit vidée, et rendrait l'attaque neutre obsolète, tout en rendant les combats plus durs.

7.8 Magasin et marchand

7.8.1 Le magasin en solo

(Réalisation : Yvon)

FIGURE 8 – Le sprite du marchand

Nous avons aussi implémentés un système de marchand et de magasin. Le marchand apparaît pendant la génération du niveau et est créé dans la salle choisie pour accueillir le magasin.

On peut utiliser le magasin à tout moment (sauf en combat) tant que l'on est suffisamment proche du marchand. Pour activer l'interface du magasin, le joueur doit appuyer sur la touche 'S' de son clavier.

L'interface est constituée de quatre boutons, l'un d'entre eux permet de quitter le magasin (cette action peut également être effectuée en

appuyant sur la touche 'S' en étant à l'intérieur du magasin. Les trois autres permettent quant à eux d'acheter des objets.

FIGURE 9 – L'interface du magasin

Au début de chaque niveau, trois objets sont choisis au hasard et sont mis en vente dans le magasin. Ils sont ainsi assignés aux boutons mentionnés ci-dessus. Lorsque l'on appuie sur l'un de ses boutons, si le joueur en possède assez, le prix de l'objet sera retiré à la quantité d'or qu'il aura obtenu au travers du niveau. En dessous de chacun de ses boutons, on peut trouver une indication du prix de chaque objet.

Actuellement, une relique coute mille or et un consommable classique deux cents.

Enfin, une fois un objet acheté, celui-ci sera jouté automatiquement à l'inventaire du joueur.

7.8.2 Le magasin en multijoueur

(Réalisation : Yvon)

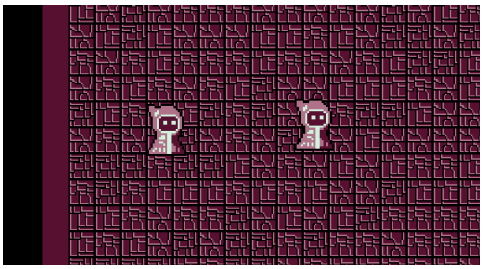
Lors d'une partie en mode multijoueur, chaque joueur a son propre magasin, ainsi même si l'un des joueurs achetait la totalité des objets du

magasin, cela ne gênerait pas l'autre joueur. De plus, les joueurs auront ainsi une chance d'acheter des objets différents.

7.9 Mode Multijoueur

(Réalisation : Marie)

Le mode multijoueur a été changé à de nombreuses reprises. Ce fut donc une partie particulièrement laborieuse. C'était, de plus, une expérience nouvelle pour chaque membre de l'équipe et a donc nécessité un investissement de la part de chacun afin de permettre la création d'un mode multijoueur fonctionnel.



2 joueurs dans la même partie

Bien que reprenant l'aventure du mode un joueur, le multijoueur nécessite quelques changements afin que le jeu reste appréciable pour tout les joueurs participants. Il a fallu donc modifier quelques aspects du jeu, notamment lors du contact entre joueurs ou les informations nécessaire au lancement d'une partie.

Par exemple, le mode solo ne nécessite aucun prérequis, autre qu'installer le jeu, tandis que le mode mutlijoueur requiert plusieurs joueurs et un pseudo (afin que chacun puisse reconnaître son personnage).

La première étape fut de choisir entre multijoueur local, avec un ordinateur host et les autres joueur clients, ou un multijoueur passant par

des serveurs.

Bien que le multijoueur local ait d'abord été envisagé et présenté lors de la première soutenance, la seconde solution fut finalement retenue et son esquisse, qui nécessitait de nombreux ajustements, avait été présenté lors de la soutenance suivante.

UNet étant devenu obsolète, cette tâche a été réalisée à l'aide du package Photon Unity Networking (PUN) qui permet de réaliser facilement la connexion à un serveur et la modification de fonctions rend leurs utilisation dans l'incorporation du multijoueur plus simple. Tout cela a donc joué un rôle important dans le choix final et la création du mode multijoueur de notre jeu.

Une fois le script nécessaire à la connexion aux serveurs effectué, des fonctions ont été ajoutées afin d'entrer un pseudo, ainsi que de créer ou rejoindre un salon, puis de lancer la partie une fois chaque joueur dans le salon, ou de le quitter si besoin.

Les pages de choix du pseudo et de sélection du salon ont ensuite été raccordées au menu afin que celles-ci soient disponibles en appuyant sur le bouton permettant de lancer ce mode.

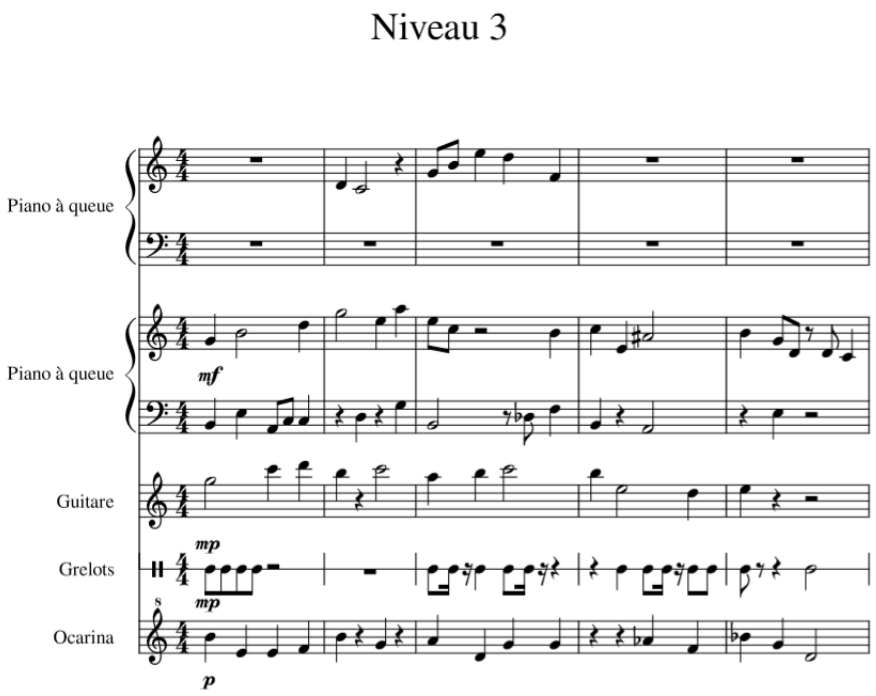
Une fois tout cela effectué, il a fallu gérer le déplacement des joueurs et par conséquent de la caméra. Cela a donc nécessité de modifier légèrement le script des mouvements afin que l'on ne puisse déplacer que son personnage. Il fallait également que chaque joueur possède une caméra qui le suive pour que chacun puisse explorer le niveau comme il le souhaite sans affecter l'autre joueur.

7.10 Réalisation artistique

Nous souhaitions réaliser un projet qui utilisait le plus de composant original possible. Cela est notamment passé par la création de courts morceau de musique ainsi que toute la réalisation graphique.

7.10.1 Musique

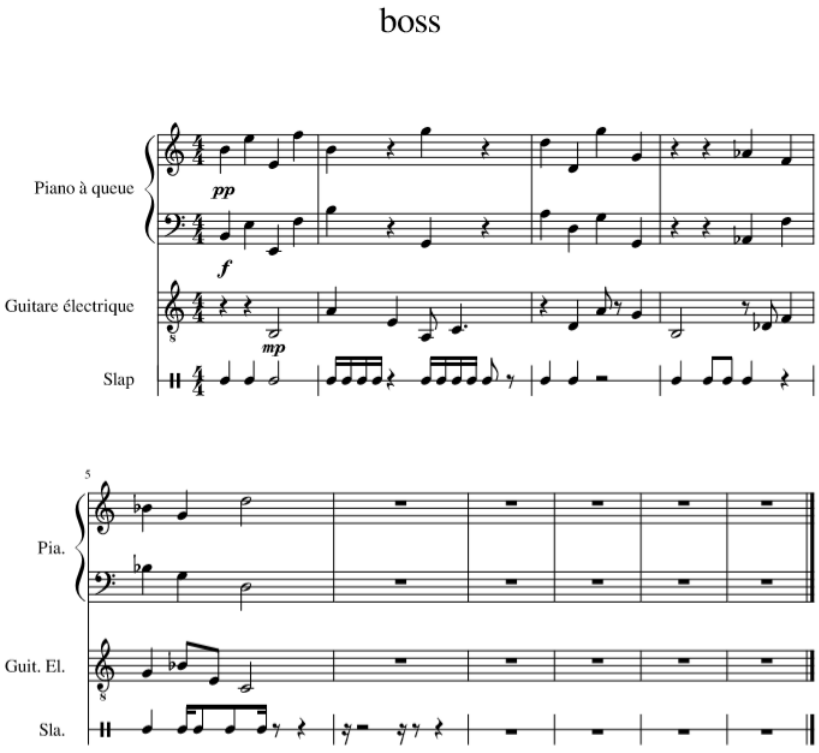
Réalisation : Marie



Début de la partition qui se joue durant le niveau 3

Le jeu comprend une musique pour chaque niveau, une pour les combats et une dernière pour les boss. Elles n'ont pas d'inspiration particulière. Bien que des versions précédentes avaient quelques morceaux de référence, elles ont été abandonnées et réécrites au fur et à mesure.

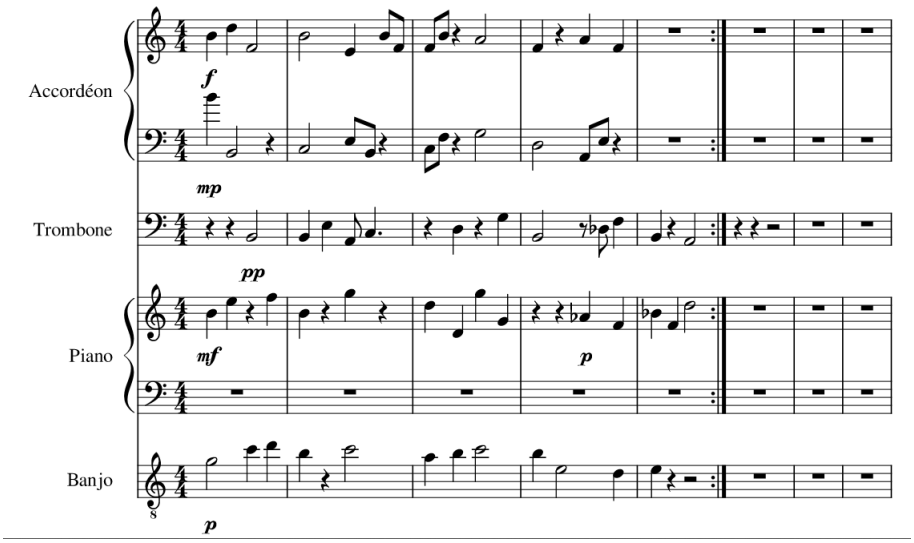
Ayant une préférence pour les instruments à cordes, cette famille est la plus représentée. Ils ont pour rôle principal de jouer la mélodie, et de guider le reste de la composition pour que cela puisse donner l'effet voulu.



Partition qui se joue durant un combat contre un boss

Les percussions ont ensuite été ajoutées pour donner un rythme à chaque morceau. Ils ont par conséquent une plus grande importance lors des combats qui sont plus rythmer, notamment grâce au tour par tour, que le déplacement dans les niveaux.

combat



Partition qui se joue durant les combats

On trouve dans certaines compositions des instruments à vent ou des choeurs afin d'apporter un léger changement et par conséquent les différencier plus facilement.

Les musiques pour chacun des niveaux sont similaires pour garder un fond sonore assez constant sans pour autant donner l'impression de toujours jouer le même niveau. La différence est principalement dans les instruments accompagnant le piano, ils varient entre chaque niveau afin d'avoir un accompagnement légèrement différent.

7.10.2 Sprites

Réalisation : Amin

La grande majorité des sprites ont été fait par nous, réaliser les sprites nous même au lieu de prendre des images sur internet nous a permis de faire 2 ou 3 choix artistiques, comme par exemple réaliser tout les sprites avec 4 couleurs ou moins (du violet (571032), du rose (B07289), du bleu (D1F3DE) et du noir(000000) pour l'outline), garder un style régulier et ne pas faire des sprites trop complexes dans un contexte simpliste, ou l'inverse.

Par exemple, tous les items rentrent dans un cadre de 32 x 32 pixels (320x320 pour le sprite ingame, plus HD) et le shopkeeper, chibi du personnage jouable et les nuages représentants les ennemis ne sont pas bizarres mis les un à côté des autres, ils rentrent tous dans le même panier, de même pour les slimes et le personnage jouable en combat, chaque sprite est au même niveau de complexité que les autres sprites qui rentrent dans le "contexte".

L'inspiration principale pour la majorité des sprites réalisés vient du jeu Celeste, l'artiste du jeu, Pedro, a réalisé un tutoriel contenant une

centaine d'articles sur Tumblr où il explique le B A BA du pixel art, en allant de la réalisation de sprites simples, à la créations d'animations fluides, en passant par les ombres et la réalisations de décors. Un des chapitres de son tutoriel parle justement des pixels arts en 2 bits, c'est à dire qui n'utilisent que 2 couleurs, c'est ce chapitre qui nous à inspiré à faire nos sprites en 4 bits.

Cependant, même si le tutoriel de Pedro a énormément aidé, ce n'est pas la première fois que je fais du pixel art, la seule chose qui est vraiment nouvelle pour moi c'est l'animation, en effet je fais du pixel art depuis quelques années pour différents médias, et j'essaye à chaque fois d'utiliser une palette de couleur spécial, d'une part pour l'esthétique, et de l'autre car je n'arrive pas du tout à gérer les couleurs "réalistes".

Le plus dur dans la réalisation des sprites était de commencer, quelles couleurs utiliser ? à quoi va ressembler le personnage principal ? quel niveau de complexité et de détails donner aux sprites ? Pourquoi les aigles n'ont pas emmené Frodon et Sam directement sur la montagne du destin ? Hotel ? Trivago.

8 Fonctionnalités abandonnées

Call of the Goose a été pour plusieurs d'entre nous notre première expérience de développement d'un jeu vidéo, ou potentiellement notre premier projet informatique, qui plus au sein d'un groupe. Il était donc à prévoir que nous surestimions notre capacité à mener à terme le projet dans les temps, et que nous aiyons été trop ambitieux concernant certains aspects du jeu.

Ceux-ci ont donc soit été tout simplement abandonnées, ou elles ont été repensés afin d'être plus réalisables.

8.1 Les classes jouables

Nous pensions au départ créer 4 classes jouables dans le jeu. Nous avions fini de les designer, cependant, n'ayant pas implémenté le système de combat, nous ne les avions pas intégrées au jeu. Cependant, au vue du développement tumultueux de celui-ci, nous avons été contraints de le simplifier, en ne laissant qu'un choix limité d'attaques.

De plus, nous pensions créer plusieurs classes avant d'avoir choisi le type de combats qu'incluerait le jeu(ici au tour par tour). Des classes différentes n'auraient que peu d'intérêt avec un système de combat aussi simple.

8.2 Arbres de compétences

Encore une fois une idée que nous avions eu dès le début du projet, mais qui finit par ne plus avoir de sens au sein du jeu final. Les arbres de compétences devaient être unique a chaque classe, et être remplis en montant de niveau. Puisque nous n'avions plus de classes différentes et qu'un arbre de compétences pour des combats au tour par tour serait redondant, nous avons décidé d'abandonner cette idée, ainsi que celle des

niveaux devenus inutiles.

8.3 Tutoriel

Nous voulions créer un tutoriel pour le jeu, accessible depuis le menu principal. Celui-ci aurait dû expliquer les bases du jeu, comme le système de combat, ou celui d'objets.

Il était peu essentiel et laissé pour la fin du projet, mais le retard pris au long de celui-ci ont fait qu'il était difficile de le finir dans les temps.

De plus, il était prévu durant une période où le jeu était en cours de design, et à été simplifié depuis. L'intérêt d'un tutoriel s'est donc amoindri avec le temps.

9 Réalisation

9.1 Le groupe

Ce projet nous a permis à chacun d'apprendre plusieurs choses. Nous avons ainsi pu découvrir une façon de réaliser un mode multijoueur ou encore de réaliser un niveau dans la manière d'un roguelike, c'est-à-dire de manière aléatoire. Nous avons également pu apprendre à utiliser le framework Unity et celui de Photon.

Au cours de la réalisation de ce projet, nous avons rencontré des bugs de diverses natures tels que des bugs de collision, des différences entre la version compilée du jeu et la version jouable depuis l'éditeur de Unity ou encore des problèmes avec le multijoueur. Plusieurs ressources telles que Unity Answers, Stack Overflow ou bien encore les documents de référence de Photon et du C# nous ont permis de les corriger en grande partie.

Nous sommes généralement satisfaits du projet. Après une fin de développement relativement compliquée sur plusieurs aspects, nous avons fini par aboutir à un jeu contenant la majorité de ce que nous prévoyions, et de manière fonctionelle.

Au cours de ce semestre nous avons dû dans la mesure du possible nous organiser pour avancer à côté du reste des cours. Cela nous a permis de nous entrainer à respecter une deadline, tout en nous donnant un avant-goût de ce que représente le travail d'un développeur.

9.2 Souleymane Sentici

J'ai rêvé pendant longtemps de créer un jeu-vidéo, ceux-ci étant ma plus grande passion. Il est donc normal que ce type de projet soit celui qui me tienne le plus a coeur. Le fait de pouvoir découvrir au moins en partie l'envers du décor de ce monde qui me fascine est l'une des raisons pour lesquelles jai décidé de m'inscrire à l'Epita.

J'ai apprécié le fait de travailler en groupe, bien que cela puisse être à double-tranchant. Cela rend évidemment la charge de travail individuelle moins élevée, puisque l'on peut diviser les tâches.
Cependant, certaines parties du projet peuvent en ralentir d'autres si elles sont dépendantes l'une de l'autre. Ceci peut donc poser des problèmes si un manque d'organisation ou une baisse de productivité de l'un des membres à lieu.

Certaines parties du projet ont posé plus de problèmes que d'autres. Le multijoueur et le système de combat, en particulier, ont du être recommencés de zéro à plusieurs reprises, et en passant par plusieurs mains. Ceci a inévitablement retardé certaines fonctionnalités clés, et le projet en général.

Ce projet m'a fait découvrir Unity, dont je ne connaissait pas le fonctionnement. Bien que très pratique, surtout pour la création d'une interface utilisateur, j'ai trouvé qu'il était peu fiable par moment. Il nous est souvent arrivé d'avoir des problèmes de préfabs qui étaient introuvable, ou de scripts qu'il ne détectait plus pour aucune raison.

Ce genre de problème à l'air récurrent si l'on en croit le nombre de posts sur des forums trouvables à ce sujet, cependant les réponses proposées pour corriger les problèmes sont très nombreuses et ne marchent que rarement.

Le fait de ne pas pouvoir travailler ensembles à cause du confinement durant une grande partie du projet à posé quelques problèmes. En effet, le fait de ne pas pouvoir se retrouver pour travailler ensemble à rendu plus difficile l'organisation et l'avancement du projet, puisque chacun devait travailler de son côté pour l'intégralité de sa tâche.

Il fallait donc avoir confiance en le fait que chacun ferait son travail en temps et en heures, ce qui n'étais pas toujours le cas.

9.2.1 Marie Legay

Ce projet fut l'occasion de comprendre le fonctionnement de certains mécanismes communs dans les jeux vidéos ainsi que de réaliser la quantité importante de travail à fournir en annexe lors de leur création, notamment pour la communication avec d'autres personnes, internes ou externes au projet, qui n'ont pas forcément les connaissances ou le temps de tout voir afin de parfaitement assimiler le projet dans son intégralité.

Le travail en équipe a des avantages mais aussi des inconvénients. En effet, travailler à plusieurs permet de réaliser plus que seul, car chacun possède des compétences différentes. à ses points forts et faibles qui se compensent par ceux des autres, mais le projet passe donc par plus de personnes, qui ne réfléchissent donc pas de la même manière.

Le code en est donc impacté puisque chaque personne le fait différemment, notamment par le raisonnement effectué pour accomplir une tâche ou résoudre un problème.

Mais ces différents raisonnements sont aussi une force qui permet de corriger des bugs puisqu'un autre point de vue peut permettre d'avoir un regard extérieur et donc de trouver plus facilement ce qui coince.

Ce fut une occasion de découvrir le fonctionnement de Unity et me familiariser avec les prefabs, bien que quelque peu hasardeux à certains moments, ainsi qu'approfondir mes connaissances sur le multijoueur et l'enchaînement des scènes.

9.2.2 Amin salmi

Travailler sur un tel projet m'a appris beaucoup de choses, la seule expérience que j'avais dans la création de jeux vidéos était avec un shoot them up que j'ai réalisé sur Scratch en 2 semaines lors d'un voyage linguistique.

Ce projet m'a aussi permis de redécouvrir le travail en équipe, en effet, on est loin de l'exposé de 5ème où il suffit de copier coller des paragraphes de Wikipedia, ici on doit travailler autour des forces et des faiblesses de chaque personnes, communiquer, trouver des solutions, être créatifs, faire des recherches poussées et apprendre.

Ce projet m'a surtout permis de me rendre compte que, comme on dit en anglais, "team works make the dream works", et que ce n'est pas tout seul qu'on atteint un bon résultat, mais c'est justement grâce aux membres de notre équipe.

Au final, je me suis plus occupé de l'aspect visuel qu'autre chose, mais même si je n'ai pas autant touché au code que Marie, Souleymane, ou Yvon, je pense que le style graphique, et donc l'ambiance, est l'un des éléments les plus important dans un jeu vidéo, et je suis content d'avoir développé cette partie du projet, car j'aspire à devenir game designer.

9.2.3 Yvon Morice

Travailler sur ce projet m'a permis d'apprendre plusieurs choses. Dans un premier temps, cela m'a permis de mieux appréhender le processus de création d'un jeu vidéo.

Ayant déjà essayé de réaliser quelques jeux basiques, principalement en Java et en Python, tels que Pong, Tetris ou des Platformers 2D du même genre que Mario, je connaissais quelques bases, ce qui m'a permis de démarrer plus facilement dans la création de ce projet.

Le fait de réaliser ce jeu avec un groupe nous a permis de réaliser certaines parties plus rapidement mais cela a également apporté quelques inconvénients tels que la difficulté d'organiser le travail dans certaines situations ou d'obtenir le travail des uns et des autres au moment attendu.

Le confinement a également participé à l'apparition de ces problèmes. Le fait de ne pas pouvoir se voir rendait la coordination parfois plus compliquée et certaines personnes tendaient à se déconnecter, parfois pendant de longues périodes, sans donner de nouvelles ni envoyer leur travail.

Cela a bien évidemment rendu la création de notre jeu plus complexe par moments, mais un travail sérieux a permis de rattraper une grande partie des retards accumulés.

J'ai également pu apprendre à utiliser l'éditeur de Unity ainsi que son framework et celui de Photon pour la partie multijoueur. Certaines parties de l'éditeur ont parfois été source de bugs, telles que la différence entre la version du jeu compilée et la version jouée dans l'éditeur ou la dissociation de certains fichiers, typiquement des scripts qui ne sont plus associés à certains prefabs.

Enfin, ce projet a été plutôt intéressant, il m'a appris plusieurs choses et m'as permis de m'entrainer à réaliser un type de projet me tenant à coeur.

10 Conclusion

Durant cette dernière période de travail, nous nous sommes concentrés sur la finalisation du mode multijoueur, l'implémentation du système de combat ainsi que l'ajout d'objets, d'ennemis, de boss ainsi que d'un magasin au travers duquel le joueur pourrait obtenir certains objets.