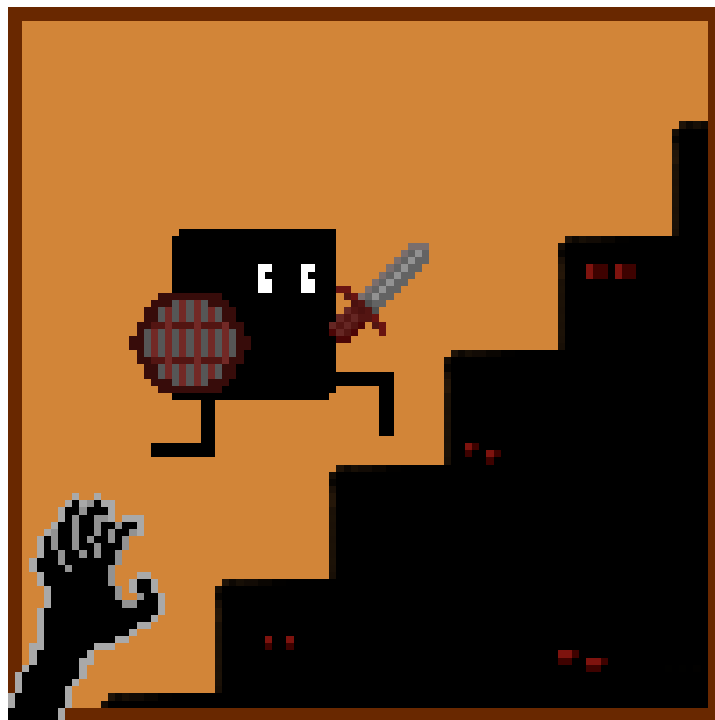


# Rapport de Projet

Orchidoclastes

(Thibaud Guignon, Erwan Doussin, Meven Cassez, Jean Bechet)

EPITA Sup (2027)



## Contents

0.0.1	Thibaud Guignon . . . . .	3
0.0.2	Erwan Doussin . . . . .	3
0.0.3	Meven Cassez . . . . .	4
0.0.4	Jean Bechet . . . . .	4
0.1	Le groupe . . . . .	5
<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Reprise du cahier des charges</b>	<b>5</b>
2.1	Le projet . . . . .	6
2.1.1	Roguelite . . . . .	6
2.1.2	Root Tower . . . . .	6
2.1.3	Inspirations . . . . .	7
<b>3</b>	<b>Rapport de la Première soutenance</b>	<b>8</b>
3.1	Jean Bechet . . . . .	8
3.2	Erwan Doussin . . . . .	15
3.3	Thibaud Guignon . . . . .	20
3.4	Meven Cassez . . . . .	23
<b>4</b>	<b>Rapport de la Seconde soutenance</b>	<b>25</b>
<b>5</b>	<b>Avancement</b>	<b>27</b>
<b>6</b>	<b>Présentations individuelles de la réalisation</b>	<b>28</b>
6.1	Jean Bechet . . . . .	28
6.2	Erwan Doussin . . . . .	34
6.3	Thibaud Guignon . . . . .	38
6.4	Meven Cassez . . . . .	41
<b>7</b>	<b>Ce qu'on en a pensé:</b>	<b>44</b>
7.1	Erwan Doussin . . . . .	44
7.2	Jean Béchet . . . . .	44
7.3	Meven Cassez . . . . .	44
7.4	Thibaud Guignon . . . . .	45

### 0.0.1 Thibaud Guignon

Mon envie de créer mes propres jeux vidéos est apparue avant même d'en avoir chez moi. Jouer quelques dizaines de minutes chez un ami a suffi à m'insuffler la passion des jeux vidéo. En découvrant la programmation en première par la spécialité NSI (que j'ai conservée en terminale avec Mathématiques), cette idée est devenue une réelle ambition dans mon esprit. Ce projet m'intéresse donc particulièrement, car il peut s'avérer être une grande avancée pour mes projets personnels et professionnels.

Au-delà de la programmation, ce projet m'a permis de mettre à profit une autre passion : la musique. Guitariste depuis bientôt 10 ans, j'ai pu m'occuper de composer et enregistrer la musique du jeu. Je me suis aussi occupé des autres sons d'ambiances.

### 0.0.2 Erwan Doussin

Épitéen pour qui la vie s'est longtemps résumé à ses études, mon éducation rime avec passion. Ainsi me suis-je dirigé vers la programmation, le choix d'en faire mon futur étant une certitude. Bien qu'ayant commencé à jouer aux jeux-vidéo assez tard, je me considère tout de même comme un fervent connaisseur du genre. Le temps que je passe à jouer a été fortement réduit cette année, pour une raison tout-à-fait inconnue, mais la flamme de mon amour pour le genre ne s'est pas éteinte. Je m'intéresse surtout aux jeux indépendants, car aucun éditeur ne peut poser des limites à l'imagination du développeur. Le rôle que j'ai tenu à cœur était de donner une identité propre au jeu, que ce soit en diversité du gameplay, des environnements ou des animations.

Ainsi ai-je eu l'honneur de servir d'artiste au groupe, fournissant les assets demandés et d'avoir le choix sur l'innovation du décors (pixel-art et animation m'intéressent tous deux autant).

### 0.0.3 Meven Cassez

Etudiant passionné d'informatique et du jeu vidéo. J'ai découvert la programmation au travers de la spécialité NSI que j'ai suivie jusqu'en terminale, mais aussi par les études suivies par mon frère. C'est après avoir découvert cette activité que de nombreux projets ont fleuri dans mon esprit, y compris celui de développer un jeu. Cela afin d'évoluer toujours plus dans ce domaine et accroître mes connaissances et facultés. De plus, étant un fan de jeu et surtout ceux de petits studios indépendants, j'ai hâte de pouvoir travailler sérieusement sur ce projet.

En plus de ma passion pour l'informatique et surtout la programmation, mon but était d'assister notre groupe dans toutes leurs tâches, ce qui fût probablement la source de tous ces maux de tête.

Au final, ce projet n'est qu'un avant-goût du possible futur que nous pourrions rencontrer en tant qu'ingénieur, ainsi qu'un bon moyen de faire évoluer nos compétences et notre esprit d'équipe.

### 0.0.4 Jean Bechet

Grand passionné tant de jeux vidéo que de jeux de rôle tels que Donjon et Dragon depuis des années, faire un roguelite est pour moi un rêve. C'est en effet l'alliance (presque) parfaite entre ces deux domaines qui ont bien plus à voir qu'il n'y paraît.

Etant un expert dans un domaine comme dans l'autre, j'espère que mes connaissances pourront apporter de l'authenticité au projet, malgré le fait que je ne fasse réellement de l'informatique que depuis début 2022.

Je pense que créer ce premier jeu a été une grande expérience pour moi car il représente un avenir que j'envisage possiblement à la sortie d'EPITA, au même titre qu'il m'a permis de développer mes compétences en travail de groupe et en informatique tant qu'en design de jeux vidéo.

## 0.1 Le groupe

Comme vous avez pu le lire, nous avons tous un point commun : l'envie de créer un jeu. Toutefois, des idées provenant de quatre grands esprits ne convergeront pas forcément.

Alors s'est installé un règlement dans notre groupe, pour améliorer l'ambiance de travail. En voici les 10 commandements :

- Tu planifieras ton travail : son avancement et son rendu.
- Tu n'éciras pas de fonction de 1000+ lignes.
- Si incompréhensible et non-documenté, ton code sera mauvais.
- Tu partageras régulièrement ton avancement sur le saint site GitHub.
- Tout travail sera pensé avec le multijoueur en tête.
- Tu aideras ton prochain et quémenderas de l'aide si besoin en est.
- Tu ne sous-estimeras aucune tâche, tout travail rendu pourra être final.
- Tu t'assureras que tes idées ne sont pas trop présomptueuses.
- Tu n'hésiteras pas à exprimer un quelconque mécontentement à tes camarades.
- Tu agiras en groupe, et le groupe agira avec toi.
- Le jeu plaira à chacun des membres du groupe.

## 1 Introduction

Ce rapport de projet relate nos débuts sur Unity, donc le processus de création du jeu vidéo Root Tower, de ses débuts à son apogée.

Pour citer notre cahier des charges, "Root Tower est un roguelite, jouable en solo ou en ligne avec des amis, dans lequel les joueurs devront arriver au sommet d'une mystérieuse tour aux multiples aspects".

Nous allons donc ici écrire ce que nous avons pensé de cette expérience, évoquer les difficultés ainsi que les découvertes incongrues, dans l'avancement chronologique du projet.

## 2 Reprise du cahier des charges

Dans le cahier des charges, nous avons présenté notre projet en énonçant les différentes idées fondamentales de celui-ci. Nous avons expliqué que, inspirés par des jeux comme Hades ou The Binding of Isaac, nous voulions créer un jeu vidéo de type Roguelite, c'est à dire que le joueur collecte des objets et améliore son équipement ainsi que ses statistiques tandis que les ennemis gagnent en puissance. Les niveaux sont générés aléatoirement et se terminent par au moins un combat de boss. Mourir termine la partie, une nouvelle partie génère de nouveaux niveaux et le joueur repart de zéro, en perdant toute améliorations.

À propos de la génération aléatoire des étages, il était prévu de constituer un ensemble de salles prédéfinies, et c'est l'agencement de celles-ci qui serait différent à chaque partie.

Nous avons aussi précisé que le jeu serait jouable aussi bien en solo qu'en coop.

Mais tout d'abord, qu'est-ce qu'un RogueLite ?

## 2.1 Le projet


L'idée de ce projet nous est venu tout simplement par passion, avant de se décider par élimination. Chacun des membres de notre groupe à déjà joué à un jeux vidéo, si ce n'est plus, et faire un jeu est donc apparu pour nous comme une évidence. En nous concertant, nous avons ainsi fait une liste des noms de jeux que nous connaissions, avant de les écarter en fonction de plusieurs critères. Le premier est si chacun de nous connaît le jeu, ou bien un se jouant de manière similaire. Le second est le style de jeu, puisque recréer le dernier open world sorti sur le marché en seulement sept mois n'était, par exemple, clairement pas envisageable. Ainsi, petit à petit, nous nous sommes retrouvés avec des jeux de plus petite ampleur, et aux noms moins connus, mais avec un point commun : le plaisir que nous avons eu à y jouer. Il s'est avéré que la plupart d'entre eux étaient des roguelites. Sur cette série d'élimination, nous avons ainsi décidé de créer un jeu similaire à ces roguelites. Mais vous vous demandez sûrement ce qu'est un roguelite ?

### 2.1.1 Roguelite

Les roguelites sont inspirés du style roguelike, celui-ci consistant à explorer un donjon rempli d'ennemis en tour-par-tour. Le joueur collecte des objets et améliore son équipement ainsi que ses statistiques tandis que les ennemis gagnent en puissance. Les niveaux y sont générés aléatoirement et se terminent par au moins un combat de boss. Mourir termine la partie, une nouvelle partie génère de nouveaux niveaux et le joueur repart de zéro, en perdant toute améliorations. L'environnement est représenté à l'aide de caractères ASCII.

Les roguelites (ou roguelike-likes) reprennent la plupart de ces éléments de façon plus libre, par exemple en supprimant le tour par tour et en utilisant des graphismes plus développés, voir même en ajoutant une progression entre les parties ! The Binding of Isaac, Hades et Enter The Gungeon sont des jeux de ce style qui ont fortement inspiré nos idées pour ce projet.

A savoir que le premier rogue-like se nomme `pedit5`, ou The Dungeon. Il est développé en 1975 sur le système PLATO, et s'inspire grandement du jeu de rôle : Donjons et dragons. Il propose au joueurs d'explorer un donjon rempli de monstres qu'il doit vaincre afin d'obtenir expérience et trésors. Il reprend de Donjons et dragons le système d'attributs et les sortilèges. Pour la petite anecdote, le jeu est créé par un programmeur du système PLATO, ce qui le pousse à choisir ce nom afin de le camoufler aux yeux des administrateurs du système.


 1er roguelike.png

### 2.1.2 Root Tower


Root Tower est donc un roguelite, jouable en solo ou en ligne avec des amis. Les joueurs doivent arriver au sommet d'une mystérieuse tour aux multiples aspects, le tout dans une ambiance mystérieuse. Pour cela, ils devront parcourir les différents étages qui la composent, vaincre les ennemis et défaire les boss d'étage qui se mettront en travers de leur chemin. Chaque étage présentera un nouvel environnement ainsi que de nouveaux ennemis et au moins un boss spécifique. Au cours de leur aventure, les joueurs collecteront des objets, des améliorations passives et des armes qui leur permettront de garder le rythme avec la difficulté grandissante du jeu. S'ils meurent, la partie est terminée et ils se retrouvent à nouveau tout en bas de la tour, ne conservant ni leurs améliorations ni leurs objets et encore moins leurs armes.

### 2.1.3 Inspirations


The Binding of Isaac, paru en 2011, est le premier de ces trois jeux à être sortis. Il présente à la fois une grande variété d'objets et d'ennemis, à la fois une génération de la carte totalement aléatoire à chaque nouvelle partie, mais aussi une histoire à part entière. De même, quoique ce ne soit pas le point qui nous ait le plus intéressé, il possède son style artistique propre. Il s'ancre profondément dans la lignée des Roguelite puisqu'il est souvent considéré parmi les trois meilleurs jeux de ce style parus à ce jour. L'équilibre du jeu en lui-même est inspirant puisque, malgré le degré de difficulté élevé, il reste amplement réalisable après plusieurs heures de jeu.

 the binding of isaac.png

Vient maintenant le tour de Enter The Gungeon, sorti en 2016. Si le jeu ne rencontre pas autant de succès que Binding of Isaac, il reste très intéressant de l'étudier. Outre les ennemis et les objets toujours aussi nombreux, il ajoute des fonctionnalités particulièrement attrayantes. Tout d'abord, la roulade : une possibilité d'éviter les projectiles ennemis se caractérisant par une courte invulnérabilité. Vient ensuite le système de hub : un menu principal permettant au joueur de débloquent des objets, des personnages ou encore des accès direct aux étages, accompagné d'une monnaie conservée entre les niveaux. S'ajoute à ça le style artistique du jeu, une vue en 2D, avec l'illusion d'être légèrement inclinée.

 enter the gungeon.png

Et enfin Hadès, sorti de son côté en 2020. Il nous présente un mode de jeu revisité, se démarquant par ses dialogues, une histoire mise bien plus au premier plan, une diversité de niveaux de difficulté bluffante (32 pour être exact!) et une bande son originale encore plus appuyée. Contrairement aux autres roguelite, il est possible de choisir les améliorations de son personnage, et donc de préciser les augmentations de puissance que l'on désire.

 hades.png

## 3 Rapport de la Première soutenance

### 3.1 Jean Bechet

Présentation individuelle de la réalisation : Jean Bechet

Bonjour, pour rappel, je suis Jean Bechet. J'ai été chargé de la catégorie "Item et équilibrage" ainsi que de celle concernant l'IA, et assiste les responsables des catégories "Level implémentation" et "Réseau". De même, il faut savoir que ce projet est ma toute première occasion de créer un jeu avec Unity et que, malgré la difficulté certaine pour s'adapter au fonctionnement de cet outil, je le trouve particulièrement intéressant !

Voici les pourcentages d'avancement attendus pour les catégories dont j'étais responsable ou assistant :

- Item et équilibrage (Responsable) : 100 %
- IA (Responsable) : 100 %
- Level implémentation (Assistant) : 100 %
- Réseau (Assistant) : 100 %

Évidemment, pour cette dernière soutenance, l'entièreté des points se doivent d'être prêt et complets à 100

Dans l'ordre, les parties sur lesquelles je me suis attardé sont les suivantes :

1. Level implémentation (Avec Erwan)
2. Réseau (Avec Meven)
3. Item et équilibrage (Avec Meven)



#### 4. IA (Avec Meven)

Si j'ai choisi cet ordre pour travailler dessus, c'est tout simplement par nécessité d'implémentation et par demande d'achèvement. Il m'était difficile de créer un réseau sans avoir un niveau de prêt pour l'essayer et des objets pour tester la répartition entre joueurs.

Item et équilibrage :

Depuis le début du projet, les tâches qui m'incombaient dans la création des item et de l'équilibrage de ceux ci étaient les suivantes :

- Créations d'objets actifs : un type d'objet nécessitant l'interaction du joueur pour provoquer un effet. Par exemple, une potion de soin à utilisation unique permettant de régénérer la santé. Il en existe deux types différents : ceux à utilisation multiple et ceux à utilisation unique.
- Création d'objets passifs : en détenir un dans l'inventaire provoque un effet constant sur le joueur tant qu'il le possède. Par exemple, l'orbe de feu permet d'infliger plus de dommages.
- Création d'objets aux effets temporaires : utiliser l'objet permet d'obtenir un effet temporaire qui se désactivera après un certain temps. Ces objets peuvent être à utilisation unique ou multiple.
- Création du système d'inventaire : ramasser les objets, les stocker, les relâcher, les utiliser, la limite de taille d'inventaire, l'affichage de l'inventaire...
- Création du système de coffre : Ouverture du coffre, génération d'objets aléatoires parmi les objets existants, fonction pseudo-aléatoire.

L'entièreté de ces tâches ont pu être amplement réalisées dans le temps imparti.

Je pense que la création d'objet est la partie qui m'a le plus plu. En effet, il s'agissait de mes premiers pas en matière de pixel art, la création d'objet m'a donc beaucoup plu. J'ai poursuivi en m'intéressant aux UI pour créer l'inventaire. Cette notion particulièrement intéressante de Unity m'a permis de créer 4 emplacements s'affichant en bas de l'écran du joueur et pouvant contenir des items. Puis, j'ai créé deux items différents, ou tout du moins leur affichage sur la map. En plus d'un collider, ils contiennent un script permettant au joueur de les ajouter à l'inventaire en cas de collision.

Une fois ajoutés à l'inventaire, il faut donc pouvoir les enlever ! C'est ici que j'ai créé un script pour leur permettre de réapparaître sur la carte lorsque la croix est cliquée, mais aussi un permettant d'utiliser l'objet "potion de soin" lorsqu'il est cliqué ! J'en ai profité pour y ajouter les effets correspondant (regain de points de vie) et faire de même au passage pour l'orbe de feu.

Une fois ces tâches faites, évidemment accompagnées de tutoriels, de forum et bien d'autres ressources pour m'y aider, j'ai décidé de créer le système de coffre. La partie la plus complexe fut le script de pseudo random, notamment par le peu de personnes expliquant comment en créer un en C #, privilégiant tous l'option de la classe Random. Mais à force de recherches, d'essais et d'erreurs, j'y suis parvenu. L'ouverture du coffre et l'apparition d'objet était ensuite une réelle partie de plaisir, notamment grâce aux connaissances que j'ai acquises au préalable.

Pour la seconde soutenance, les objectifs avec les items étaient assez faible et l'avancement nécessaire était en réalité bas. L'objectif était d'expérimenter différentes interactions avec le joueur. C'est pour ça que les fonctionnalités choisies sont :

- Augmenter la vitesse de déplacement
- Permettre de réutiliser un objet plusieurs fois
- Placer un gameobject en utilisant un item

it Seul la dernière fonctionnalité n'a pas été entièrement implémentée en raison d'un défaut sur l'effet du gameObject sur la map. Cependant, cet objectif n'étant que secondaire nous avons décidé que nous pouvions nous en passer sans que cela n'affecte l'avancement nécessaire aux item.

Pour la dernière soutenance, les objectifs avec les items étaient assez complexe et l'avancement nécessaire était en réalité assez élevé. Ici, des fonctionnalités et type d'utilisation pour les objets ont été implémentés :

- Augmenter la vitesse d'attaque
- Augmenter les dégâts
- Avoir des effets temporaires

it Tous ces objectifs ont pu être remplis et implémentés. La réelle difficulté venait d'implémenter ces notions parmi les fichiers déjà existant sans créer d'erreur ni qu'ils ne gênent le déroulement normal d'une partie sans ces item.

Concernant l'équilibrage, j'ai dû commencer à me pencher dessus, notamment avec les item réutilisables. La problématique suivante se posait : comment implémenter un objet à utilisation unique aux côtés d'un objet réutilisable tout en le rendant intéressant auprès du joueur. Forcément au long terme un objet à utilisation unique est moins intéressant qu'un objet à utilisation répétée. Cependant, l'avantage de la potion de soin comparé au gland est qu'il permet de se soigner intégralement, pouvant sauver de situations critiques. De même, en le rendant plus commun que le gland, le joueur hésite moins à l'utiliser que le gland.

Toujours en termes d'équilibrage, il m'en a fallu un pour le taux d'apparition d'objet, et j'ai décidé de rendre le jeu aussi généreux que cruel. Même si le nombre d'objet paraît grand, la difficulté envers les ennemis suffit largement à compenser cette générosité. Le nombre de coffre par étage n'est pas défini par avance et peut autant être nul qu'élevé.

Evidemment, je parle ici avec le pronom "je", mais l'entièreté du travail à été réalisé en collaboration avec Meven.

Level implementation :

Les tâches qui m'incombaient dans le "Level implementation" étaient les suivantes :

- Création des assets : par asset, j'entend des blocs de pixels de taille 32x32 nous permettant d'afficher les salles. Il s'agit du décor dans lequel évolue le joueur.
- Créations des différentes room possibles : Il s'agit de la création des asset de tous les types de room que le joueur pourra rencontrer, ainsi que de la bossroom (permettant d'accéder à l'étage suivant) et des chest room (là où se trouve les coffres).
- Création du premier étage type : Assemblage type du premier étage visant à montrer la disposition générale des salles.
- Création du système de caméra : Déplacement de la caméra avec le joueur de salle en salle.
- Apparition des créatures, des coffres et du boss sur la map.

- Limitation de la taille de la map afin d'éviter les générations infinies, les étages trop grands et permettre l'utilisation des salles à trois entrées. Ainsi, le labyrinthe plus semblable à une suite de couloir peut enfin devenir un véritable labyrinthe.
- Ajout de décoration aux salles afin de leur donner un visuel plus agréable.
- Ajout du second étage.
- Ajout de la fin du jeu.

Ici, contrairement aux items, les tâches ont été réparties en fonction de l'aisance et des capacités de chacun. Erwan s'est occupé de la création des assets ainsi que des différentes room possibles tandis que je me chargeais de l'étage type ainsi que du système de caméra. La raison à cela est tout simplement l'expertise dans le domaine du pixel art que possède Erwan, et l'intérêt que je porte aux caméras.

En effet, j'ai choisi ici de m'imposer une difficulté : ne pas utiliser Cinemachine, l'extension pourtant populaire de Unity. Outre les limites qu'il m'imposait, je trouvais ça plus intéressant de m'en charger afin de reproduire le système des caméras d'anciens jeux tels que les anciens Zelda. A coup de documentation Unity, j'ai donc pu en créer un satisfaisant. Ce système permet, lorsqu'une porte d'une salle est franchie, de faire avancer la caméra à la salle suivante, avec en prime un défilement continu pour y parvenir.

Une fois cela fait, j'ai créé une map type comme nous souhaitons en réaliser avec le générateur aléatoire à l'avenir. Evidemment, cette map n'a pas été créée totalement aléatoirement. Bien que le système de génération ne soit pas implémenté, une version papier de la logique utilisée existe déjà. C'est en la suivant et en utilisant un dé pour l'aléatoire que cette carte à été créée. Les seuls éléments placés volontairement ont été les salles de coffre et la salle de boss.

Génération de la carte :

Dès le départ, nous savions que la génération de carte risquait d'être la partie la plus complexe. En effet, nous souhaitons une génération possédant les caractéristiques suivantes :

- Aléatoire et changeant à chaque niveau
- Capable d'implémenter les salles d'elle même au bon endroit
- Capable de générer des ennemis dans les salles attendues
- Capable de créer des salles possédant des caractéristiques particulières ou non (coffre et boss)
- Capable de faire apparaître les joueurs

Malgré leur simplicité au premier abord, les implémenter représente un défi en soi. Tout d'abord par le nombre d'implémentations possibles. J'ai tout d'abord pensé à créer un arbre qui, de lui même, serait capable de décider de si la salle possède ou non un certain nombre d'entrées. De même, j'aurais pu ajouter un type d'ennemi propre à chaque nœud de l'arbre, ainsi que le nombre d'ennemis pouvant apparaître et leur position.

Contrôler le nombre de salles pouvant apparaître et faire en sorte que seules les feuilles puissent

être des salles de coffre ou de boss aurait été bien plus simple. Cependant, je ne maîtrise que peu les arbres, et nous n'avons pas réellement étudié les arbres autres que les arbres binaires. C'est pourquoi, tant par manque de connaissance que de maîtrise, j'ai décidé d'écarter cette option.

Éloignant de cette manière l'option la plus aisée à envisager, je me suis penché un peu plus sur une méthode récursive. Générer la première salle et, à partir de celle-ci créer une boucle qui fait apparaître de nouvelles salles en se basant sur la position initiale. Problème avec cette option est de générer des ennemis différents en fonction des salles sans les attribuer directement. Le code n'aurait absolument pas été optimisé et se serait plus approché d'un brut force.

De même l'emplacement, l'orientation et les collisions entre les salles n'était pas optimale, loin de là même. La moitié des caractéristiques que nous attendions n'étaient même pas atteintes, ce qui ne rendait pas cette option viable, voire même la pire d'entre toutes. Mais travailler avec cette récursion m'a permis d'envisager d'autres options et d'explorer des idées qui m'ont été utiles pour la solution finale. Par ailleurs, m'exercer à utiliser les salles de la sorte m'a permis d'être bien plus à l'aise avec celles-ci et à gérer les cas de génération infinie, chose bien compliquée avec Unity puisqu'il ne comprend aucun outil pour s'en échapper dans l'application en elle-même.

C'est ainsi que j'en arrive à la méthode pour laquelle j'ai opté : une génération aléatoire itérative sans utiliser d'arbres. Ma manière de procéder était d'abord inefficace : générer d'abord les cul de sac (salle à une entrée possédant soit un boss, soit un coffre, soit aucun des deux) et de créer un algorithme capable de les regrouper. Outre la difficulté du code, il générerait des cartes inintéressantes dans lesquelles l'exploration était limitée et remplie de bug, d'erreur de génération de cartes et j'en passe.

Décidant alors de procéder à l'inverse, je m'inspirais de la méthode récursive et me basait sur une unique salle d'apparition. À partir de celle-ci, cinq objets attachés : un en son centre et un pour chacun des centres des salles à apparaître autour. Les autres salles possèdent toutes deux ou une porte.

Dans le cas de deux portes, le point au centre reste afin d'éviter les doubles générations et un second point apparaît là où la salle suivante est censée apparaître. Dans le cas d'une porte, seul le point en son centre est conservé. De cette manière, chacun des points n'étant pas au centre se voit attribuer un script de génération qui va faire apparaître la salle suivante jusqu'à arriver à un cul de sac. Une fois la génération entièrement aléatoire achevée, un second script vient parcourir à nouveau l'entièreté des salles exceptée la salle d'apparition et faire apparaître des ennemis. Dans la salle la plus éloignée du joueur, la créature qui apparaît est un boss. Les coffres apparaissent eux de manière entièrement aléatoire dans les dernières salles au bout des chemins. Cependant, travailler avec cette option m'aura apporté un avantage et un défaut :

- Avantage : utiliser une liste permet de travailler avec des probabilités, permettant de prévoir une taille moyenne pour la carte et évitant des générations trop grandes ou trop petites tout en conservant des aspects de salles pouvant différer.
- Défaut : utiliser des salles à trois entrées génère des donjons bien trop grands et devient incontrôlable, forçant l'abandon de l'utilisation de salles à trois entrées.

Malgré cela et la diminution en complexité du donjon (le joueur fait face à quatre chemins différents

seulement), certains aspects la rendent réellement intéressante, notamment lorsque deux salles s'entrecroisent, créant une intersection et ne figeant pas le donjon dans un éternel cycle à quatre chemins.

C'est pour cela que pour la troisième soutenance j'ai décidé de revoir le cœur même de cette implémentation afin de remettre une idée abandonnée par le passé : les salles à 3 entrées. Les ajouter permet de complexifier le labyrinthe mais la raison qui m'avait poussé à abandonner était tout simplement la taille du labyrinthe. Ne possédant pas encore de limite à la génération et me basant uniquement sur les probabilités, il était difficile de considérer une génération accessible sans possibilité de map infinie ou trop grande. C'est pour cela que j'ai décidé d'ajouter une limite à cette génération, empêchant toute autre salle que le cul de sac d'apparaître à partir d'une limite fixée arbitrairement à vingt. Une fois les salles générées, les créatures apparaissent dedans exceptée pour les salles de coffre et la salle la plus éloignée du joueur, celle qui contiendra le boss.

Réseau :

Les tâches qui m'incombaient dans le réseau étaient les suivantes :

- Possibilité que deux joueurs puissent jouer en même temps.
- Séparation des caméras entre les deux joueurs.
- Séparation des fonctionnalités telles que l'inventaire entre les deux joueurs.
- Indépendance des dégâts.
- Gestion des deux joueurs par les IA.
- Changement de room.

L'entièreté de ces tâches ont pu être réalisées dans le temps imparti, malgré des difficultés majeures pour les réaliser. À l'avenir, les tâches qui m'incomberont seront :

- Implémenter un mot de passe pour accéder à la room.
- Adapter les fonctionnalités à venir.

Cette partie a été pour moi la plus complexe. La raison majeure à cela était la nécessité de tout merger correctement à chaque push et de faire en sorte que les fonctions réalisées fonctionnaient tout de même. C'était une des raisons justifiant la réalisation quasi complète nécessaire, et nous avons bien fait d'en demander tant. Outre la complexité d'apprentissage et de compréhension de photon, c'est cette problématique de devoir vérifier la validité de chaque script qui nous a été particulièrement chronophage. En ajoutant à ça la génération de map, la génération d'ennemi et les bugs rencontrés durant la première soutenance, la partie réseau a encore une fois su bien nous occuper.

Malgré les difficultés rencontrées, moi-même et Meven avons pu tout de même atteindre les objectifs fixés initialement, non sans difficulté. Nous les avons même dépassés ! En effet, la partie réseau ne manque quasi de rien, si ce n'est les fonctionnalités à venir et un possible mot de passe pour accéder aux rooms.

IA :

Les tâches qui m'incombaient concernant l'IA étaient les suivantes :

- Creation d'un modele d'ennemi animé.
- Suivi du joueur par l'ennemi.
- Dommages infligés au joueur.
- Apparition aléatoire dans les salles
- Santé, mort et nombre de créatures
- Implémentation des boss
- Attaques spéciales pour les boss

L'entièreté de ces tâches ont pu être réalisées dans le temps imparti, malgré la complexité. La seule exception est le confinement dans une seule salle. La raison à cela est simplement une décision commune déterminant que les IA devait être capable de suivre le joueur entre les salles, ne lui laissant aucun répit.

La création de l'IA est aussi intéressante que complexe. Tout d'abord en raison du peu de temps que nous avons à cause du réseau, mais aussi car créer une IA n'est pas une mince affaire. C'est pour ça que nous nous sommes contentés de créer un algorithme qui permet aux ennemis de suivre le joueur, ainsi que la possibilité des dégâts. L'ennemi va suivre le joueur, plus ou moins rapidement en fonction de sa classe, et tenter d'entrer en contact avec lui pour lui infliger des dégâts. Le joueur, de son côté, va devoir les frapper avec son arme afin de leur infliger des dégâts et de les éliminer.

Le boss est une créature particulière. Lorsque le joueur entre dans la salle où il est présent, il entre en activité. Son mode d'attaque est assez simple à comprendre : il fait apparaître des attaques magiques dans la salle de manière rapide et régulière afin de blesser le joueur. Il n'y a pas de moyen de les prévoir, il s'agit d'attaques surprises. Le boss a 20 points de vie, bien plus que n'importe quelle créature du jeu. A sa mort, il fait apparaître l'accès à l'étage suivant. C'est donc une étape essentielle d'une partie pour le joueur, s'il ne le bat pas, il ne pourra pas avancer plus loin.

Le second étage possède des créatures différentes, bien plus rapides et mortelles pour le joueur. Même si cet étage ne possède pas de Boss, la rapidité des créatures suffit à le rendre bien plus difficile que le premier.

### 3.2 Erwan Doussin

Bonjour, c'est Erwan Doussin qui vous écrit à présent.

Avant d'écrire à propos de mes tâches, je me dois de vous expliquer mon rapport avec le projet. Tout d'abord, il s'agit d'une expérience agréable: la création d'un jeu vidéo est une envie de longue date pour moi, mais je dois admettre préférer un jeu sur Unity que sur Pygame, la bibliothèque python de création de jeux vidéo.

Je digresse; toujours est-il que je peux écarter les difficultés auxquelles j'ai fait face pour dire que j'apprécie ce travail de groupe.

L'ordre dans lequel j'ai abordé mes travaux est le suivant :

- Menu Interfaces (Avec Thibaud)
- Core Gameplay (Avec Thibaud)

- Graphismes (avec tout le monde)
- Level Design (avec Jean)

Je suis donc responsable des menus, du Level-Design et des graphismes, ainsi qu'un assistant dans la partie Core Gameplay.

Cela signifie que mon travail est essentiel à beaucoup de rôles : par exemple, menus et salles doivent avoir été créés et implémentés avant leur utilisation dans le multijoueur.

Aussi, je m'efforce de faire toute partie graphique le plus rapidement possible, car travailler est bien agréable pour mes camarades avec un rendu visuel plutôt qu'un placeholder (un substitut à l'aspect simpliste, en somme).

Et voici les pourcentages d'avancement qui étaient attendus dans lesdites catégories:

- Menu et Interface (Responsable) : 50 %
- Core Gameplay (Assistant) : 40 %
- Graphismes (Responsable) : 25 %
- Level Design (Responsable) : 25 %

Quoique je n'ai pas fini ces différentes parties dans cet ordre.

En effet, la partie graphique, si ce n'est son implémentation, est de loin la moins importante d'un point de vue pédagogique, mais est ironiquement proportionnellement la plus chronophage : je me suis donc retrouvé à chercher des idées, designs et comportements pour les ennemis plus en cours que chez moi, où je me concentrais sur l'apprentissage d'Unity.

En rétrospective, c'est plutôt l'initiation au domaine du jeu vidéo qui fût notre principal ennemi jusqu'à cette première soutenance. Le fait que nous ayons choisi de créer un jeu en 2D nous facilite la tâche sur certains points, mais il y a tant de types de jeu en 2D que nous avons trouvé peu de guides concernant la création d'un roguelike sur Unity. Enfin, au moins nous créons un jeu avec notre propre esthétique.

A présent, voici en détail le travail que j'ai effectué pour cette première soutenance:

#### Menu Interfaces

La partie qui pourrait sembler la plus simple m'a causé bien des soucis. En effet, notre menu a connu différentes versions avant d'être ce qu'il est, et est toujours loin d'être terminé. C'est la première chose que j'ai essayé d'implémenter dans Unity, et je devais d'abord comprendre le fonctionnement de l'éditeur, cela m'aurait évité bien des soucis !

J'ai opté pour un menu principal assez simple, avec trois boutons basiques (jouer, paramètres, quitter). Les paramètres n'étaient pas ma priorité jusqu'à maintenant, car le jeu est loin de la version finale que nous lui avons imaginé.

Il est à noter que je n'ai pas encore touché aux menus du multijoueur, car Meven avait besoin d'une interface claire et concise.

À propos d'interfaces, l'HUD a été un véritable calvaire. L'inventaire n'était pas facile à représenter, surtout lorsque Jean a commencé à implémenter des objets (potion, orbe...).

Nous avons opté pour un inventaire depuis lequel nous pourrions jeter les objets avec une croix. La dernière partie de l'HUD, la barre de vie du joueur, n'est pour l'instant qu'à moitié implémentée, en intégralité par Thibaud.



Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Les paramètres devront être fonctionnels.
- Une refonte de l'écran de chargement, si possible avec animation.
- Le menu principal devra refléter le nouvel étage.

#### Core Gameplay

Ma principale implication dans cette section concerne la définition des mouvement. Nous avons en effet tenté d'inclure plusieurs types de mouvements :

- Les mouvement directionels basiques.
- Un mouvement qui freine le personnage lorsqu'il se tourne sur lui-même.
- Un "dash", c'est-à-dire une impulsion du personnage pouvant servir d'esquive.
- Des déplacements rapides, un "mode course" en somme.

Pour l'instant, nous n'avons implémenté que les mouvements directionnels, mais avons déjà recherché comment les autres types de mouvements pourront l'être, à l'exception des déplacements rapides.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Le dash au moins devra avoir été implémenté.
- Le joueur devra pouvoir utiliser des attaques secondaires.
- Le joueur devra pouvoir interagir avec les ennemis.

#### Level-Design

Le Level-Design consiste en l'imagination d'un environnement cohérent, sa conceptualisation, mais aussi en son implémentation.

Ainsi, j'ai commencé par réfléchir à un environnement qui ne lirait pas une histoire mais la raconterait visuellement. Sa réalisation graphique appartient par définition à la partie Graphismes, mais j'ai d'abord essayé de comprendre comment j'allais l'implémenter avant de la réaliser visuellement.

Ainsi ai-je utilisé tout d'abord des assets publics, pour apprendre à créer des salles.

Après avoir stylisé des "dalles" (tuiles pouvant être placées de manière répétitive), j'ai pu créer des salles : elles sont donc orientées dans toutes les directions (haut, bas, gauche, droite, avec plusieurs directions possibles à la fois).

J'ai ensuite, avec Jean, travaillé sur les hitbox des murs (déjà séparés du sol) et nous avons créé une carte, sur laquelle nous pourrions tester toute fonctionnalité, en suivant la logique de génération aléatoire dont nous avons décidé.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- L'aléatoire implémenté et fonctionnelle, ne serait-ce que pour la génération des donjons.
- La création de toutes les salles pour le prochain étage au moins.

### Graphismes

Je parle ici de toutes les images utilisées pour l'instant dans le jeu : nous nous devons d'éviter l'utilisation d'assets gratuits si nous voulons créer un jeu de toutes pièces.

Ainsi, bien que j'ai d'abord procuré à mes camarades des images simples pour leur permettre de travailler, j'ai vite commencé à désigner les ennemis de tous les étages (dont la plupart ne sont toujours pas utilisés), et surtout tout ce qui concerne le personnage contrôlable.

Aussi, je m'efforce de faire toute partie graphique le plus rapidement possible, car travailler est bien agréable pour mes camarades avec un rendu visuel plutôt qu'un placeholder (un substitut à l'aspect simpliste, en somme).

Le milieu du pixel-art et de l'animation 2D était nouveau pour moi, et j'ai dû modifier à maintes reprises certains travaux. J'ai toutefois cherché à rendre les animations satisfaisantes, et j'ai appris plusieurs techniques du mouvement.

Les animations du personnage principal n'ont pas encore toutes été implémentées, car nous nous réservons certaines fonctionnalités qu'en cas de besoin.

En ce qui concerne l'environnement, j'ai créé des designs pouvant être répétés, et des objets pouvant être placés dessus pour casser une possible uniformité.

Dans un souci d'optimisation des fichiers, j'ai organisé les images et animations en spritesheets : toutes les images d'un environnement ou d'une animation sont donc placées dans un même fichier, faciles d'accès, prenant moins de place, et pouvant être modifiées facilement.

Ci-joint un exemple de spritesheet, comportant toutes les frames de l'animation de marché de notre premier ennemi :



Enfin, l'implémentation des images a rarement été un problème, à défaut des animations liées à certaines conditions (comme celles de déplacement).

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Une illustration grande-échelle, principal outil de promotion du jeu, qui pourrait être affichée sur le site.
- Une variation thématique des objets de l'environnement, probablement sur les robots, le tout visant un nouvel étage. Cela implique plus de 3 ennemis, 7 décors, et des tuiles répétables adaptées au contexte de l'étage.
- Des animations pour tous les ennemis qui seront alors implémentés.
- Une refonte du coffre en apparence.

En ceci consiste toute ma méthode de travail. J'ai passé l'étape de l'apprentissage, et doit maintenant m'habituer à certaines tâches répétitives.

Je ne suis que peu satisfait de notre avancée dans le projet pour cette première soutenance, car je pensais que nous aurions dépassé de loin nos objectifs. Toutefois, mes camarades m'ont fait relativiser : nous avons atteint nos objectifs, et le plus dur est passé. Nous ne devons pas nous reposer sur nos lauriers (si tant est que nous en ayons).

Je cherche donc à aider tout camarade dans le besoin, en me documentant sur les différentes sections du projet, dont celles qui ne me concernent pas, car apprendre toutes les facettes de Unity ne peut que s'avérer utile plus tard.

Il convient bien sûr d'aider mes camarades à atteindre leurs objectifs personnels.

### 3.3 Thibaud Guignon

Ce projet donne lieu à ma découverte du logiciel Unity. Malgré la difficulté que j'ai eu à le prendre en main en premier lieu, mon utilisation en est de plus en plus fluide. Concernant ma contribution au projet, je suis chargé des sections "Core Gameplay", "Menu Interfaces", "Site Web", "Graphismes" et "Sound Design".

La progression prévue dans chacune de ces sections dont je suis responsable ou assistant sont les suivantes :

- Core Gameplay (Responsable) : 40 %
- Menu Interfaces (Assistant) : 50 %
- Site Web (Assistant) : 10 %
- Graphismes (Assistant) : 25 %
- Sound Design (Responsable) : 0 %

Le pourcentage attendus jusqu'à présent de la section Site Web est assez faible parce qu'il ne fait pas partie directement de la partie principale du projet, et celui de la section Sound Design est nul car ce n'est pas une partie fondamentale du projet et est moins prioritaires que d'autres sections.

Je me suis chronologiquement attelé aux tâches dans cet ordre :

1. Core Gameplay (avec Erwan)
2. Graphismes (avec Erwan)
3. Menu Interfaces (avec Erwan)
4. Site Web (avec Meven)

Cet ordre m'est venu naturellement avec l'avancée de mon travail. J'ai d'abord créé des actions à effectuer pour le joueur, puis lié ces actions à leurs animations correspondantes. Ensuite, il fallait afficher au joueur des informations sur son personnage. Enfin, j'ai contribué à l'avancement du site web dont je me suis occupé en dernier car la progression attendue dans cette section était plutôt faible. Je n'ai même pas commencé à travailler sur le sound design, conformément au cahier des charges.

Core Gameplay :

Voici ce que j'ai effectué jusqu'à présent dans cette section :

- Création du personnage et de ses statistiques : le personnage possède des statistiques telles que des points de vie (maximums et actuels), ou une vitesse d'attaque.
- Mouvements du personnage : Récupération des directions données par le joueur pour faire en sorte que le personnage se déplace dans ces dites directions.

- Attaque avec l'épée : Lorsque le joueur appuie sur la touche d'attaque, le personnage donne un coup d'épée, tant qu'un certain temps dépendant de la vitesse d'attaque du personnage s'est écoulé depuis la dernière attaque. L'attaque inflige des dégâts aux ennemis que l'épée touche. Pour l'instant, il ne peut utiliser qu'une seule arme.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Attaque avec d'autres armes : Le personnage pourra utiliser d'autres armes qui auront des comportements différents et plus complexes, telles que des armes à distance par exemple.
- Dash: Le personnage pourra se déplacer très rapidement de façon instantanée
- Attaques secondaires : Certaines armes auront plusieurs utilisations différentes qui nécessiteront des actions supplémentaires.
- Ajout de comportements particuliers au personnage et ses actions (par exemple, attaquer plusieurs fois successivement déclenche un combo) améliorant les sensations en jeu.

Cette section est fondamentale car c'est elle qui permet au joueur de jouer au jeu. Étant la première section à laquelle je me suis attelé, la simplicité de ce que j'ai implémenté pour l'instant ne m'a pas empêché d'y consacrer beaucoup de temps. La découverte de Unity était déconcertante, mais cette étape m'a permis de mieux comprendre le fonctionnement du logiciel et d'être plus efficace par la suite. Ce qui va être nécessaire, car jusqu'à présent, les actions étaient peu nombreuses et peu complexes, mais cela va rapidement changer au fur et à mesure de l'avancement du projet.

Graphismes :

Voici ce que j'ai effectué jusqu'à présent dans cette section :

- Animation de l'attaque avec l'épée et attribution de l'animation au personnage : Lorsque le joueur appuie sur le bouton d'attaque, l'épée devient active et tourne dans un quart de cercle autour du personnage dans la direction du déplacement. Si le personnage n'est pas en train de se déplacer, il attaque vers la droite.
- -Attribution des animations de déplacement au personnage : Lorsque le joueur ne fait rien, le personnage est dans l'animation par défaut. Dès que le joueur fait bouger le personnage, les animations de déplacement (correspondant à la direction choisie) de celui-ci se déclenchent, avec une animation de transition si le déplacement est vers la droite ou la gauche.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Attribution des animations à chaque action du personnage.
- Animation de certaines armes.
- Création de certains sprites en pixel art.

Nous avons choisi de créer nous même les visuels de notre jeu, même si nous aurions pu prendre parmi les nombreux visuels gratuits trouvables sur Internet. Ce choix me convient, car il nous permet de faire ce que l'on veut. De plus, j'ai trouvé du plaisir à m'occuper de cette section.

Menu Interfaces :

Voici ce que j'ai effectué jusqu'à présent dans cette section :

- Création d'un support pour le HUD (Head Up Display, affichage tête haute en français) qui se situe en dehors de l'endroit où le joueur évolue et utilise une caméra autre que la caméra principale.
- Création d'une barre de vie qui se vide ou se remplit (et change de couleur) lorsque que le joueur perd ou regagne des points de vie. Elle se trouve dans le HUD.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Ajout de l'affichage de l'inventaire (déjà existant) au HUD.
- Ajout potentiel de certaines informations utiles au HUD.
- Affichage de certaines informations de l'autre joueur (lorsqu'il y en a un).
- Menu pause
- Menu d'options
- Boîtes de dialogues occasionnelles

Je me suis surpris à trouver cette section étonnamment intéressante. La technique permettant d'implémenter la barre de vie que j'ai découverte grâce à un tutoriel m'a permis de mieux comprendre le fonctionnement de Unity. Cependant, le visuel de la barre de vie n'a pas été créé par un membre du groupe. J'avais essayé d'en faire un qui n'a pas aboutit, alors je me suis rabattu sur celui-ci en attendant.

Site Web :

Voici ce que j'ai effectué jusqu'à présent dans cette section :

- Création d'un script python utilisant le micro framework flask pour tester le site en local.
- Création d'une page HTML présentant notre groupe.
- Ajout de style à la page d'accueil en CSS.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Création de nouvelles pages HTML présentant le projet.
- Ajout de style aux pages HTML en CSS.

J'avais déjà pratiqué du HTML et du CSS en Première et en Terminale, et c'est réconfortant de se retrouver en terrain connu, après mes bégaiements lors de ma découverte de Unity. Le CSS m'a quand même donné un peu de fil à retordre.

### 3.4 Meven Cassez

Voici ma partie, je suis Meven CASSEZ. Je suis dans ce projet le responsable Réseau et site web ainsi que l'assistant IA, item et équilibrage et pour finir sound design.

Voici donc un ordre de l'importance accordée à chaque partie et par conséquent du temps consacré:

- Réseau
- item équilibrage
- IA
- Site web
- sound design

Tout d'abord, après avoir mis en place les bases du gameplay avec mes collègues, c'est-à-dire, les déplacements ainsi que l'animation, les attaques et les collisions. Je me suis penché sur la mise en place du en ligne. Une fonctionnalité qui est une pièce maîtresse mais aussi une des plus complexes à mettre en place.

Elle s'est décomposée en cette liste d'étapes:

- Mise en place de l'asset photon
- liaison au serveur en ligne
- mise en place du lobby
- connection au lobby
- possibilité de créer des rooms et de les rejoindre
- mise en place de room privées
- synchronisation des joueurs et de leurs apparitions
- synchronisation des déplacements et des animations de mouvements
- mise en place de hud individuel
- synchronisation des caméras
- mise en place de l'apparition des ennemis côté serveur
- mise en place de l'affichage côté client

De ce fait, après avoir chercher de nombreux moyens d'héberger notre multijoueur, nous avons décidé d'utiliser Photon.

Nous avons ensuite mis en place le système de connexion au "lobby" du serveur ainsi que la création de "room" privées permettant au joueur de rejoindre la partie de leurs amis en communiquant le nom de celle-ci.

Après cette étape, qui fut relativement longue mais dans un certain sens peu complexe lorsque comparée à la suivante, je suis passé à la synchronisation de chaque joueur sur le serveur. Une étape quelque peu longue et fastidieuse, qui fut accompagnée d'un problème sur le serveur photon qui a mis plusieurs jours à être résolu.

La solution ?

La Whitelist de la géolocalisation du serveur qui avait des problèmes, et qui créait une partie sur un serveur en Europe mais tentait lors d'une connexion à cette partie de la rejoindre sur un serveur US. Une fonctionnalité de Photon fort utile dans certains cas mais qui fut un réel problème dans le notre.

Ce problème enfin résolu, nous sommes passés à la synchronisation des déplacements, des caméras ainsi que des inventaires.

Néanmoins, afin de mettre en place des bases solides sur le multijoueur et éviter les bugs, nous avons dû réduire le temps passé sur les IA, puisque dans la logique, si le serveur ne fonctionne pas même si notre IA, elle, fonctionne, nous ne pourrions même pas en profiter. Voici donc la répartition d'une grande partie de notre temps.

Pour la partie du développement sur les items ainsi que leur équilibrage, nous avons commencé par la recherche d'un algorithme de type "randomizer" afin de pouvoir mettre en place des pourcentages d'obtention des items.

Après cette partie majeure, nous avons implémenté des objets afin d'aider notre joueur dans son aventure.

Ainsi, le joueur pourra trouver durant sa quête des potions de soins et d'autres objets spéciaux. Ceux-ci sont pour le moment obtenables via des coffres. Pour plus de détails, je vous laisse le soin de lire la partie de Jean Béchet dans laquelle il fournit de plus amples détails.

Pour la partie sur l'IA nous avons pour le moment de nombreuses recherches et algorithmes en préparation mais nous avons déjà implémenté l'algorithme de "following" des ennemis.

Et pour finir, la dernière partie sur laquelle nous avons travaillé sur cette période : Le site web. Celui-ci est pour le moment du même niveau que celui d'un site web moyen dans les années 1990 mais il est voué à évoluer rapidement.

Nous avons pour le moment mis en place les différentes pages ainsi que le header présent sur toutes celles-ci et permettant de naviguer au travers des différentes pages. Il va dans un premier temps regrouper nos rapports de soutenances, dont celui-ci et cela dès la seconde soutenance.



Maintenant, je vais rapidement parler de l'avenir du projet.

Pour les améliorations à venir concernant la partie réseau, nous espérons pouvoir mettre en place un système de mot de passe avant de rejoindre les parties afin d'éviter les connexions de joueurs inconnus ou simplement contre le gré de l'hôte.

Nous souhaitons également : mettre en place le système de validation pour changer d'étage (les étages n'étant pas encore implémentés); ajouter une distinction entre chaque joueur afin de pouvoir différencier facilement le joueur n°1 du joueur n°2; gérer les objets lâchés par les monstres à leur mort.

Pour la partie item équilibrage, nous aurons donc pour objectif d'implémenter le fait que les monstres puissent lâcher des objets à leur mort et possiblement d'ajouter de nouveaux objets.

Maintenant pour la partie IA, c'est là que réside la plus grosse partie du travail, il faudra faire en sorte que les IAs puissent attaquer le joueur le plus proche ainsi que réaliser l'implémentation côté serveur.

Quant à la partie site web, nous mettrons en place la totalité des pages et commenceront à passer le site dans le même style que l'univers de notre jeu.

Et pour finir avec la partie sound design, nous essayerons de mettre en place des sons sur les animations de base, c'est-à-dire, lors des attaques, de l'ouverture de coffres ou lors de l'utilisation d'une potion de soin.

## 4 Rapport de la Seconde soutenance

Une fois de plus, c'est github, notre outil de travail collaboratif, qui nous a causé le plus de tort. Ainsi, les membres du groupe ont push plus ou moins fréquemment selon les jours, et les merges n'ont pas toujours facilité nos rendus. La veille avant l'écriture de ce rapport, nous nous sommes réunis afin de mettre en commun tout ce que nous n'avions que sur des instances personnelles du projet.

- Méthode de travail en groupe Un changement majeur : nous nous aidons à présent dans tous les domaines, dont ceux qui ne nous concernent pas. Nous connaissons tous à présent plutôt bien Unity, mais seulement dans les domaines que nous avons travaillé. Cela implique que lorsque nous ne savons pas bien manipuler certaines fonctionnalités de Unity ou ne comprenons pas certaines fonctions, mais en étant les experts du groupe dans d'autres catégories. Par exemple, tout le monde a souhaité aider Meven dans le réseau, bien qu'il en soit toujours le principal connaisseur. Le réseau étant une catégorie difficile à manier, les membres du groupe ont surtout cherché à faciliter l'implémentation du multijoueur et à le déboguer, connaissant les possibles causes de problèmes dû à une absence de considération du multijoueur lors de leur implémentation.
- Direction artistique (inchangée depuis la dernière soutenance) Ayant choisi un style de jeu 2D, nous avons pris la décision de présenter le jeu avec du pixel art. Certains d'entre nous n'avaient aucune expérience dans le domaine contrairement à d'autres. C'est pour ça que le choix du

site à été plutôt équilibré. Il se devait d'être assez complexe pour que les meilleurs d'entre nous ne soient pas restreints et suffisamment accessible pour les débutants. Après moult essais de sites, nous avons finalement décidé de choisir Piskel. Nous l'avons choisi pour son ergonomie, sa simplicité d'utilisation et les nombreuses options offertes gratuitement, mais surtout pour son outil permettant de dégrader un pixel de couleur sur des teintes plus claires ou plus foncées, nous donnant la possibilité de créer des dégradés et des motifs complexes plus simplement. Pour la majorité des sprites nous avons choisi le 32x32 comme dimension, le 64x64 nous semblant trop complexe et détaillé pour un jeu comme le nôtre. Seuls certains items seront réalisés dans des tailles différentes.

## 5 Avancement

Notre principal soucis a d'abord été de corriger les bogues de la dernière soutenance. Ainsi avons-nous principalement travaillé la semaine d'après la première soutenance, avant de faire une pause à la reprise du cycle scolaire.

Ensuite, nos priorités étaient de conceptualiser l'étage suivant et de réaliser les fonctionnalités essentielles du jeu : tous ont passé des soirées et nuits sur le projet, que ce soit sur la génération aléatoire, le multijoueur ou l'animation et les graphismes.

L'avance que nous avons pris pour la première soutenance nous a aussi permit de nous concentrer sur l'ajout de fonctionnalités secondaire, comme les items ou les déplacements ennemis. Nous pensons que la quantité de travail offert pour cette soutenance satisfait à peine nos attentes exprimées dans le cahier des charges, mais le plus d'ur reste derrière nous : nous pouvons à présent nous concentrer sur la diversification du contenu, plutôt que sur les fonctionnalités principales, qui sont suffisantes, si ce n'est à perfectionner.

## 6 Présentations individuelles de la réalisation

### 6.1 Jean Bechet

Présentation individuelle de la réalisation : Jean Bechet

Bonjour, pour rappel, je suis Jean Bechet. J'ai été chargé de la catégorie "Item et équilibrage" ainsi que de celle concernant l'IA, et assiste les responsables des catégories "Level implémentation" et "Réseau". De même, il faut savoir que ce projet est ma toute première occasion de créer un jeu avec Unity et que, malgré la difficulté certaine pour s'adapter au fonctionnement de cet outil, je le trouve particulièrement intéressant !

Voici les pourcentages d'avancement attendus pour les catégories dont j'étais responsable ou assistant :

- Item et équilibrage (Responsable) : 60 %
- IA (Responsable) : 60 %
- Level implémentation (Assistant) : 65 %
- Réseau (Assistant) : 75 %

Si ces pourcentages ont été décidés si haut, c'est tout simplement car nous avons pris en compte le fait que nous avons déjà découvert Unity pour la première soutenance et que nous avons déjà passé la majorité de notre temps à expérimenter, à regarder des tutoriels et à découvrir l'outil plutôt que de créer réellement notre jeu, ce qui aujourd'hui n'est plus nécessaire. L'expertise acquise, nous nous sommes réellement lancés dans le jeu, avançant bien plus que lors de la première soutenance.

Dans l'ordre, les parties sur lesquelles je me suis attardé sont les suivantes :

1. Level implémentation (Avec Erwan)
2. Réseau (Avec Meven)
3. Item et équilibrage (Avec Meven)
4. IA (Avec Meven)

Si j'ai choisi cet ordre pour travailler dessus, c'est tout simplement par nécessité d'implémentation et par demande d'achèvement. Il m'était difficile de créer un réseau sans avoir un niveau de prêt pour l'essayer et des objets pour tester la répartition entre joueurs.

Item et équilibrage :

Les tâches qui m'incombaient dans la création des item et de l'équilibrage de ceux ci étaient les suivantes :

- Créations d'objets actifs : un type d'objet nécessitant l'interaction du joueur pour provoquer un effet. Par exemple, une potion de soin à utilisation unique permettant de régénérer la santé.
- Création d'objets passifs : en détenir un dans l'inventaire provoque un effet constant sur le joueur tant qu'il le possède. Par exemple, l'orbe de feu permet d'infliger plus de dommages.
- Création du système d'inventaire : ramasser les objets, les stocker, les relâcher, les utiliser, la limite de taille d'inventaire, l'affichage de l'inventaire...
- Création du système de coffre : Ouverture du coffre, génération d'objets aléatoires parmi les objets existants, fonction pseudo-aléatoire.

L'entièreté de ces tâches ont pu être réalisées dans le temps imparti. A l'avenir, les tâches qui m'incomberont seront :

- Ajouter un plus grand nombre d'objets.
- Équilibrer ces objets ainsi que leurs probabilités d'apparition.
- Permettre l'obtention d'objets par l'élimination d'ennemis.
- Plus grande variété d'armes.

Je pense que la création d'objet est la partie qui m'a le plus plu. En effet, il s'agissait de mes premiers pas en matière de pixel art, la création d'objet m'a donc beaucoup plus, quoique je n'avais à en créer qu'un de chaque type, ainsi qu'un affichage pour l'inventaire. J'ai poursuivi en m'intéressant aux UI pour créer l'inventaire. Cette notion particulièrement intéressante de Unity m'a permis de créer 4 emplacements s'affichant en bas de l'écran du joueur et pouvant contenir des items. Puis, j'ai créé deux items différents, ou tout du moins leur affichage sur la map. En plus d'un collider, ils contiennent un script permettant au joueur de les ajouter à l'inventaire en cas de collision.

Une fois ajoutés à l'inventaire, il faut donc pouvoir les enlever ! C'est ici que j'ai créé un script pour leur permettre de réapparaître sur la carte lorsque la croix est cliquée, mais aussi un permettant d'utiliser l'objet "potion de soin" lorsqu'il est cliqué ! J'en ai profité pour y ajouter les effets correspondant (regain de points de vie) et faire de même au passage pour l'orbe de feu.

Une fois ces tâches faites, évidemment accompagnées de tutoriels, de forum et bien d'autres ressources pour m'y aider, j'ai décidé de créer le système de coffre. La partie la plus complexe fut le script de pseudo random, notamment par le peu de personnes expliquant comment en créer un en C #, privilégiant tous l'option de la classe Random. Mais à force de recherches, d'essais et d'erreurs, j'y suis parvenu. L'ouverture du coffre et l'apparition d'objet était ensuite une réelle partie de plaisir, notamment grâce aux connaissances que j'ai acquises au préalable.

Pour la seconde soutenance, les objectifs avec les items étaient assez faible et l'avancement nécessaire était en réalité bas. L'objectif était d'expérimenter différentes interactions avec le joueur. C'est pour ça que les fonctionnalités choisies sont :

- Augmenter la vitesse de déplacement
- Permettre de réutiliser un objet plusieurs fois
- Placer un gameobject en utilisant un item

it Seul la dernière fonctionnalité n'a pas été entièrement implémentée en raison d'un défaut sur l'effet du gameObject sur la map. Cependant, cet objectif n'étant que secondaire nous avons décidé que nous pouvions nous en passer sans que cela n'affecte l'avancement nécessaire aux item.

Concernant l'équilibrage, j'ai dû commencer à me pencher dessus, notamment avec les item réutilisables. La problématique suivante se posait : comment implémenter un objet à utilisation unique aux côtés d'un objet réutilisable tout en le rendant intéressant auprès du joueur. Forcément au long terme un objet à utilisation unique est moins intéressant qu'un objet à utilisation répétée. Cependant, l'avantage de la potion de soin comparé au gland est qu'il permet de se soigner intégralement, pouvant sauver de situations critiques. De même, en le rendant plus commun que le gland, le joueur hésite moins à l'utiliser que le gland.

Toujours en terme d'équilibrage, il m'en a fallu un pour le taux d'apparition d'objet, et j'ai décidé de rendre le jeu aussi généreux que cruel. En effet, entre 0 et 4 coffres peuvent apparaître à chaque étage. Cela signifie que le joueur peu possiblement recevoir autant d'objet que son inventaire peu en contenir comme aucun, le laissant se débrouiller par ses propres moyens.

Evidemment, je parle ici avec le pronom "je", mais l'entièreté du travail à été réalisé en collaboration avec Meven.

Level implementation :

Les tâches qui m'incombaient dans le "Level implementation" étaient les suivantes :

- Création des assets : par asset, j'entend des blocs de pixels de taille 32x32 nous permettant d'afficher les salles. Il s'agit du décor dans lequel évolue le joueur.
- Créations des différentes room possibles : Il s'agit de la création des asset de tous les types de room que le joueur pourra rencontrer, ainsi que de la bossroom (permettant d'accéder à l'étage suivant) et des chest room (là où se trouve les coffres).
- Création du premier étage type : Assemblage type du premier étage visant à montrer la disposition générale des salles.
- Création du système de caméra : Déplacement de la caméra avec le joueur de salle en salle.

L'entièreté de ces tâches ont pu être réalisées dans le temps imparti. A l'avenir, les tâches qui m'incomberont seront :

- Création de la génération aléatoire d'étages.
- Création des asset et des room de l'étage 2.
- Amélioration du design du premier étage.

Ici, contrairement aux items, les tâches ont été réparties en fonction de l'aisance et des capacités de chacun. Erwan s'est occupé de la création des assets ainsi que des différentes room possibles tandis que je me chargeais de l'étage type ainsi que du système de caméra. La raison à cela est tout simplement l'expertise dans le domaine du pixel art que possède Erwan, et l'intérêt que je porte aux caméras.

En effet, j'ai choisi ici de m'imposer une difficulté : ne pas utiliser Cinemachine, l'extension pourtant populaire de Unity. Outre les limites qu'il m'imposait, je trouvais ça plus intéressant de m'en charger afin de reproduire le système des caméras d'anciens jeux tels que les anciens Zelda. A coup de documentation Unity, j'ai donc pu en créer un satisfaisant. Ce système permet, lorsqu'une porte d'une salle est franchie, de faire avancer la caméra à la salle suivante, avec en prime un défilement continu pour y parvenir.

Une fois cela fait, j'ai créé une map type comme nous souhaitons en réaliser avec le générateur aléatoire à l'avenir. Evidemment, cette map n'a pas été créée totalement aléatoirement. Bien que le système de génération ne soit pas implémenté, une version papier de la logique utilisée existe déjà. C'est en la suivant et en utilisant un dé pour l'aléatoire que cette carte a été créée. Les seuls éléments placés volontairement ont été les salles de coffre et la salle de boss.

Génération de la carte :

Dès le départ, nous savions que la génération de carte risquait d'être la partie la plus complexe. En effet, nous souhaitons une génération possédant les caractéristiques suivantes :

- Aléatoire et changeant à chaque niveau
- Capable d'implémenter les salles d'elle même au bon endroit
- Capable de générer des ennemis dans les salles attendues
- Capable de créer des salles possédant des caractéristiques particulières ou non (coffre et boss)

Malgré leur simplicité au premier abord, les implémenter représente un défi en soi. Tout d'abord par le nombre d'implémentations possibles. J'ai tout d'abord pensé à créer un arbre qui, de lui même, serait capable de décider de si la salle possède ou non un certain nombre d'entrées. De même, j'aurais pu ajouter un type d'ennemi propre à chaque nœud de l'arbre, ainsi que le nombre d'ennemis pouvant apparaître et leur position.

Contrôler le nombre de salles pouvant apparaître et faire en sorte que seules les feuilles puissent être des salles de coffre ou de boss aurait été bien plus simple. Cependant, je ne maîtrise que peu les arbres, et nous n'avons pas réellement étudié les arbres autres que les arbres binaires. C'est pourquoi, tant par manque de connaissance que de maîtrise, j'ai décidé d'écarter cette option.

Éloignant de cette manière l'option la plus aisée à envisager, je me suis penché un peu plus sur une méthode récursive. Générer la première salle et, à partir de celle-ci créer une boucle qui fait apparaître de nouvelles salles en se basant sur la position initiale. Problème avec cette option est de générer des ennemis différents en fonction des salles sans les attribuer directement. Le code n'aurait absolument pas été optimisé et se serait plus approché d'un brut force.

De même l'emplacement, l'orientation et les collisions entre les salles n'était pas optimale, loin de là même. La moitié des caractéristiques que nous attendions n'étaient même pas atteintes, ce qui ne rendait pas cette option viable, voire même la pire d'entre toutes. Mais travailler avec cette récursion m'a permis d'envisager d'autres options et d'explorer des idées qui m'ont été utiles pour la solution finale. Par ailleurs, m'exercer à utiliser les salles de la sorte m'a permis d'être bien plus à l'aise avec celles-ci et à gérer les cas de génération infinie, chose bien compliquée avec Unity puisqu'il ne comprend aucun outil pour s'en échapper dans l'application en elle-même.

C'est ainsi que j'en arrive à la méthode pour laquelle j'ai opté : une génération aléatoire itérative sans utiliser d'arbres. Ma manière de procéder était d'abord ineffective : générer d'abord les cul de sac (salle à une entrée possédant soit un boss, soit un coffre, soit aucun des deux) et de créer un algorithme capable de les regrouper. Outre la difficulté du code, il générerait des cartes inintéressantes dans lesquelles l'exploration était limitée et remplie de bug, d'erreur de génération de cartes et j'en passe.

Décidant alors de procéder à l'inverse, je m'inspirais de la méthode récursive et me basait sur une unique salle d'apparition. A partir de celle-ci, cinq objets attachés : un en son centre et un pour chacun des centres des salles à apparaître autour. Les autres salles possèdent toutes deux ou une porte.

Dans le cas de deux portes, le point au centre reste afin d'éviter les doubles générations et un second point apparaît là où la salle suivante est censée apparaître. Dans le cas d'une porte, seul le point en son centre est conservé. De cette manière, chacun des points n'étant pas au centre se voit attribuer un script de génération qui va faire apparaître la salle suivante jusqu'à arriver à un cul de sac. Une fois la génération entièrement aléatoire achevée, un second script vient parcourir à nouveau l'entièreté des salles exceptée la salle d'apparition et faire apparaître des ennemis. Dans la salle la plus éloignée du joueur, la créature qui apparaît est un boss. Les coffres apparaissent eux de manière entièrement aléatoire dans les dernières salles au bout des chemins. Cependant, travailler avec cette option m'aura apporté un avantage et un défaut :

- Avantage : utiliser une liste permet de travailler avec des probabilités, permettant de prévoir une taille moyenne pour la carte et évitant des générations trop grandes ou trop petites tout en conservant des aspects de salles pouvant différer.
- Défaut : utiliser des salles à trois entrées génère des donjons bien trop grands et devient incontrôlable, forçant l'abandon de l'utilisation de salles à trois entrées.

Malgré cela et la diminution en complexité du donjon (le joueur fait face à quatre chemins différents seulement), certains aspects la rendent réellement intéressante, notamment lorsque deux salles s'entrecroisent, créant une intersection et ne figeant pas le donjon dans un éternel cycle à quatre chemins.

Réseau :

Les tâches qui m'incombaient dans le réseau étaient les suivantes :

- Possibilité que deux joueurs puissent jouer en même temps.
- Séparation des caméras entre les deux joueurs.



- Séparation des fonctionnalités telles que l'inventaire entre les deux joueurs.
- Indépendance des dégâts.
- Gestion des deux joueurs par les IA.
- Changement de room.

L'entièreté de ces tâches ont pu être réalisées dans le temps imparti, malgré des difficultés majeures pour les réaliser. A l'avenir, les tâches qui m'incomberont seront :

- Implémenter un mot de passe pour accéder à la room.
- Adapter les fonctionnalités à venir.

Cette partie à été pour moi la plus complexe. La raison majeure à cela était la nécessité de tout merge correctement à chaque push et de faire en sorte que les fonctions réalisées fonctionnaient tout de même. C'était une des raisons justifiant la réalisation quasi complète nécessaire, et nous avons bien fait d'en demander tant. Outre la complexité d'apprentissage et de compréhension de photon, c'est cette problématique de devoir vérifier la validité de chaque script qui nous a été particulièrement chronophage. En ajoutant à ça la génération de map, la génération d'ennemi et les bug rencontrés durant la première soutenance, la partie réseau à encore une fois su bien nous occuper.

Malgré les difficultés rencontrées, moi-même et Meven avons pu tout de même atteindre les objectifs fixés initialement, non sans difficulté. Nous les avons même dépassés ! En effet, la partie réseau ne manque quasi de rien, si ce n'est les fonctionnalités à venir et un possible mot de passe pour accéder aux room.

IA :

Les tâches qui m'incombaient concernant l'IA étaient les suivantes :

- Creation d'un modele d'ennemi animé.
- Suivi du joueur par l'ennemi.
- Dommages infligés au joueur.
- Apparition aléatoire dans les salles
- Santé, mort et nombre de créatures
- Confinement dans la salle

L'entièreté de ces tâches ont pu être réalisées dans le temps imparti, malgré la complexité. La seule exception est le confinement dans une seule salle. La raison à cela est simplement une décision commune déterminant que les IA devait être capable de suivre le joueur entre les salles, ne lui laissant aucun répit. A l'avenir, nous tacherons d'améliorer les points suivants :

- Création de différentes IA (en fonction de la classe d'ennemi)
- Obligation d'élimination du boss pour changer d'étage

La création de l'IA est aussi intéressante que complexe. Tout d'abord en raison du peu de temps que nous avons à cause du réseau, mais aussi car créer une IA n'est pas une mince affaire. C'est pour ça que nous nous sommes contentés de créer un algorithme qui permet aux ennemis de suivre le joueur, ainsi que la possibilité des dégâts. L'ennemi va suivre le joueur, plus ou moins rapidement en fonction de sa classe, et tenter d'entrer en contact avec lui pour lui infliger des dégâts. Le joueur, de son côté, va devoir les frapper avec son arme afin de leur infliger des dégâts et de les éliminer. Pour le moment, le boss n'est qu'un ennemi classique mais à l'avenir il s'agira d'une créature à part avec ses propres attaques, une résistance plus élevée et une meilleure IA que la moyenne.

## 6.2 Erwan Doussin

Bonjour, c'est Erwan Doussin qui vous écrit à présent.

Avant d'écrire à propos de mes tâches, je me dois de vous expliquer mon rapport avec le projet. Tout d'abord, il s'agit d'une expérience agréable : la création d'un jeu vidéo est une envie de longue date pour moi, mais je dois admettre préférer un jeu sur Unity que sur Pygame, la bibliothèque python de création de jeux vidéo.

Je digresse ; toujours est-il que je peux écarter les difficultés auxquelles j'ai fait face pour dire que j'apprécie ce travail de groupe.

L'ordre dans lequel j'ai abordé mes travaux depuis la dernière fois est le suivant :

- Graphismes
- Items & Equilibrage (avec Jean)
- Menus (avec Meven)
- Level Design (avec Jean)

Je suis responsable des Menus, du Level-Design et des Graphismes, ainsi qu'un assistant dans la partie Core Gameplay.

Cela signifie que mon travail est essentiel à beaucoup de rôles : par exemple, menus et salles doivent avoir été créés et implémentés avant leur utilisation dans le multijoueur.

Aussi, je m'efforce de faire toute partie graphique le plus rapidement possible, car travailler est bien agréable pour mes camarades avec un rendu visuel plutôt qu'un placeholder (un substitut à l'aspect simpliste, en somme).

Et voici les pourcentages d'avancement qui étaient attendus dans lesdites catégories pour cette soutenance:

- Menus (Responsable) : 90 %
- Items & Equilibrage : 60 %
- Graphismes (Responsable) : 80 %
- Level Design (Responsable) : 65 %

Ces objectifs que nous nous sommes fixés devaient nous donner une idée d'une progression idéale, c'est-à-dire de ce que nous espérions avoir fait d'ici cette seconde soutenance.

Toutefois, je ne m'y suis pas tant collé que ça cette fois-ci : je constate un retard dans la partie Graphismes, qui est pourtant celle qui m'a le plus occupé. Cela peut toutefois être dû à mon envie de remplir toutes ces salles au mieux : ce que nous avons est amplement suffisant.

De plus, celui nous a permis de prendre de l'avance concernant le Level Design.

À présent, voici en détail le travail que j'ai effectué pour cette seconde soutenance:

#### Menu Interfaces

Cette fois-ci, c'est avec Meven que j'ai travaillé les menus : en particulier ceux du multi-joueur.

Car si ce que nous avions réalisé pour la première soutenance était fonctionnel, c'était loin d'être ergonomique !

Nous nous sommes orientés vers un menu plus sobre, mais dont le fond ferait rêver. Il s'agit là de notre premier asset qui n'est pas de nous (c'est une image évoquant le jeu Donjons et Dragons).

La partie la plus difficile, ironiquement, était de tester : démarrer deux instances du jeu, créer une salle et la rejoindre furent fastidieux, surtout pour des changements mineur.

C'est là un problème qui, à ce qu'on dit, concerne tout le multijoueur. Toutefois, j'ai eu la chance de seulement tester les menus en multijoueur : il n'est même pas question de synchronisation entre deux clients.

Nous pensions lors de notre dernière soutenance qu'il serait important de recréer l'écran de chargement. Cependant, ayant réglé des problèmes de multijoueur, la durée de celui-ci a été fortement réduit. De ce fait, nous passons cette tâche en second plan, afin de se concentrer sur les principales.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Les paramètres devront être fonctionnels.

#### Items & Equilibrage

En sortant de notre première soutenance, nous n'avions qu'un seul item non-fonctionnel. Depuis, nous avons pu en créer d'autres.

Bien qu'ils ne soient pas tous fonctionnels, nous cherchons à trouver des utilités diversifiées pour ceux-ci. En effet, les buffs et armes sont utiles, mais peu réfléchies. Une idée que nous aimerions bien implémenter serait le fait de poser un décor : une soie posée équivaldrait à une toile d'araignée.

Cela nous mène à un autre virage dans notre cursus : les objets peuvent être trouvés dans les coffres, mais et s'ils pouvaient être lâchés par les ennemis à leur mort ? Je les ai conceptualisés en conséquent, et attend de pouvoir les tester lorsque les ennemis pourront mourir. Ainsi, les araignées pourraient lâcher à leur mort de la soie, tandis que les arbres pourraient lâcher des glands, qui, quant à eux, guériraient le joueur à leur utilisation. Ce serait alors une bonne façon de gérer une possible difficulté via leurs taux d'apparition.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Si possible, implémenter des drops à la mort des ennemis.

- Rendre totalement aléatoire leur apparition dans les coffres.
- Les rendre fonctionnels
- Diversifier leurs effets

### Level-Design

Le Level-Design consiste en l'imagination d'un environnement cohérent, sa conceptualisation, mais aussi en son implémentation.

Dans une continuation de ce qui avait été fait pour la première soutenance, j'ai tout d'abord réfléchi à ce que l'environnement devait évoquer.

Toutefois, la première soutenance derrière moi, j'avais déjà l'expérience du premier étage.

Avec Jean, nous avons commencé la génération aléatoire. Le modèle que nous avons suivi divergeait toutefois drastiquement de ce que nous avions prévu : j'avais créé des salles orientées dans toutes les directions et avec entre une et trois sorties, mais ces dernières nous posaient problème.

Suivant les conseils de Jean, je n'ai donc que créé des salles avec au plus deux sorties pour le second étage.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Un placement plus diversifié des décors dans les salles.
- Donner plus d'interactions avec lesdits décors (hitbox, effets sur le joueur...)

### Graphismes

Je parle ici de toutes les images utilisées pour l'instant dans le jeu : nous nous devons d'éviter l'utilisation d'assets gratuits si nous voulions créer un jeu de toutes pièces.

J'ai commencé par implémenter les décors que j'avais créé pour la dernière fois : j'avais réalisé de nombreux décors pour l'étage dit "naturel", mais je ne savais pas encore comment j'allais les utiliser.

Notre projet étant un projet de jeu 2D, il m'a suffi de les poser sur un calque placé au-dessus du sol et des murs. Ainsi, les torches que j'avais faites la dernière fois sont à présent séparées de leur support. On peut donc les placer où on le souhaite, sur le sol comme à un étage différent.

En parlant des étages, j'ai fini bien plus vite les graphismes du nouvel étage en ce qui concernent l'environnement. En effet, je connaissais cette fois-ci les dimensions et palettes de couleurs que j'allais utiliser.

J'ai même pu consacrer plus de temps sur certains aspects : j'ai réalisé une différente source de lumière pour l'étage futuriste (c'est ainsi que nous l'appelons), plus détaillée que n'importe quel autre sprite.

J'ai donc pu plus vite m'atteler à la création des items : j'ai commencé par en créer des classiques (potion, couronne), puis seulement ensuite des moins conventionnels.

Un grand avancement pour moi a été l'animation : non seulement j'ai trouvé de nouvelles façons d'animer des mouvements, et commencé à animer un peu tout (y compris les items),

mais j'ai surtout enfin réussi à les implémenter de manière dynamique pour les monstres. Je détaillerai dans la partie "Autres", car cela concerne l'IA.

Contrairement à la dernière fois, j'ai donné plus d'importance à la répétition des textures. Cela concerne aussi les animations, dont on ne peut normalement pas remarquer le début. À présent, je pose bien toutes les images qui ne sont pas des animations sur les spritesheets. J'ai réglé plusieurs bugs graphiques qui nous posaient problème avant :

- Traitement des images : aucune image ne devrait à présent être floue. En effet, l'antialiasing du projet ne doit pas traiter les sprites dit "pixelisés".
- Gestion des fichiers : les images et spritesheets ne sont plus compressés, améliorant le rendu des couleurs.
- Animations : les animations sont à présent instanciées séparément, ce qui permet de gérer leur vitesses et départ différemment. Ainsi, les torches ne sont pas toutes synchronisées, tandis que les machines le sont.

Les tâches que je me suis imposé pour la suite, en pensant les réaliser, étaient :

- Finir l'étage mécanique.
- Créer de nouveaux ennemis, certains pouvant être utilisés en tant que boss.
- Une refonte du coffre en apparence.

#### Autres

- IA : Afin d'implémenter les animations dynamiques pour les ennemis, j'ai apporté quelques changements aux scripts concernant leurs déplacement, permettant de mieux contrôler leurs déplacements, connaissant ainsi leur direction.
- Core Gameplay : L'attaque, si le joueur est immobile, est maintenant dirigée vers la dernière direction que le joueur a regardé.
- Réseau : J'ai aidé Meven à synchroniser les animations, les déplacements de mobs et la génération de map entre plusieurs joueurs.

Je sais que mes parties, à l'exception des graphismes, sont assez avancés, suffisant sur ce que nous avions prévu. J'espère toutefois tout améliorer : ce que nous avons est fonctionnel mais pourrait être pauffiné.

Je continue d'être à disposition de qui-le-veut, et à attendre l'accord de mes camarades avant de commencer un nouveau travail.

### 6.3 Thibaud Guignon

Ce projet a donné lieu à ma découverte du logiciel Unity. Malgré la difficulté que j'ai eu à le prendre en main en premier lieu, ce qui a pu causer des pertes de temps conséquentes à plusieurs reprises, mon utilisation en est devenue de plus en plus fluide. Appréciant travailler sur ce projet, j'ai donc pu participer à son développement à un rythme raisonnable.

Concernant ma contribution au projet, j'étais chargé des sections "Core Gameplay", "Menu Interfaces", "Site Web", "Graphismes" et "Sound Design". Ces sections m'intéressaient particulièrement pour diverses raisons. Tout d'abord, je suis guitariste et j'aime beaucoup écouter de la musique, donc je tenais particulièrement à m'occuper de la musique du jeu. Ensuite, je souhaitais participer à la création du site web car j'avais apprécié pratiquer de l'HTML et du CSS au lycée. Les sections "Core Gameplay" et "Menu Interfaces" m'intéressaient car elles sont fondamentales à n'importe quel jeu vidéo. Enfin, j'avais envie de m'essayer aux graphismes.

Core Gameplay :

Voici ce que j'ai effectué au cours du semestre dans cette section :

- Création du personnage et de ses statistiques : le personnage possède des statistiques telles que des points de vie (maximums et actuels), ou une vitesse de déplacement.
- Mouvements du personnage : Récupération des directions données par le joueur pour faire en sorte que le personnage se déplace dans ces dites directions.
- Attaque avec l'épée : Lorsque le joueur appuie sur la touche d'attaque quand il est en possession de l'épée, le personnage donne un coup d'épée, tant qu'un certain temps dépendant de la vitesse d'attaque s'est écoulé depuis la dernière attaque. L'attaque inflige des dégâts aux ennemis que l'épée touche.
- Ajout du dash : Le joueur peut effectuer un dash, c'est-à-dire brièvement se déplacer très vite dans une direction choisie. Une fois un dash effectué, il faut attendre un peu avant de pouvoir en refaire un.
- Knockback : Attaquer les ennemis leur donne un mouvement de recul. Cela permet au joueur de mieux ressentir l'impact du coup.
- Attaque avec la hache : Lorsque le joueur appuie sur la touche d'attaque quand il est en possession de la hache, le personnage donne un coup de hache, tant qu'un certain temps dépendant de la vitesse d'attaque s'est écoulé depuis la dernière attaque. L'attaque inflige des dégâts aux ennemis que la hache touche.
- Attaque secondaire avec la hache : Lorsque le joueur maintient la touche d'attaque secondaire quand il est en possession de la hache, le personnage charge son attaque spéciale. Cette dernière existe en 3 versions, dépendant de la durée du maintien de la touche. Lorsque la touche est relâchée, la hache effectue de une à trois rotations complètes autour du personnage en fonction de la durée de chargement.

Malgré les nombreux problèmes que j'y ai rencontrés, j'ai beaucoup apprécié travailler sur cette section. J'ai eu du mal avec le knockback à cause de la manière dont on faisait se déplacer les ennemis, mais j'ai pris du plaisir à implémenter le dash car c'est une mécanique que j'affectionne particulièrement dans les jeux qui la proposent, et je suis particulièrement satisfait de l'attaque secondaire de la hache.

Site Web :

Voici ce que j'ai effectué au cours du semestre dans cette section :

- Création de plusieurs pages HTML permettant de présenter notre projet.
- Ajout de liens permettant une bonne connectivité entre les pages du site.
- Ajout de style en CSS pour ne pas avoir tout en noir et blanc et aligné à droite.
- Ajout du téléchargement du projet et de ce rapport.

Le style CSS n'est pas très beau mais je trouve que c'est mieux que rien.

Sound Design :

Voici ce que j'ai effectué au cours du semestre dans cette section :

- Musique du menu : Une musique se joue dans le menu. L'ambiance est menaçante.
- Musique du premier niveau : Une musique se joue lorsque le joueur parcourt le premier niveau. Les sonorités ont été pensées pour faire penser aux aspects plaisants de se promener dans une forêt (car l'étage a un thème végétal), mais la musique est dissonante, car les végétaux ont évolué de sorte à devenir hostiles et menaçants.
- Musique du deuxième niveau : Une musique se joue lorsque le joueur parcourt le deuxième niveau. Il y a des rythmes un peu robotiques.
- Bruits de pas : Lorsque le joueur se déplace, on entend les bruits de pas du personnage.

J'ai adoré travailler sur cette section. Tous les sons ont été faits à la guitare électrique. La qualité du son n'est pas très bonne, car je n'avais à ma disposition qu'un téléphone portable pour enregistrer (je referrais peut-être les enregistrements si j'acquiers le matériel nécessaire et que j'en ai le temps). Les clips audio ont été agencés entre eux pour créer la musique sur le site BandLab.

Graphismes :

Voici ce que j'ai effectué au cours du semestre dans cette section :

- Animation de l'attaque avec l'épée et attribution de l'animation au personnage : Lorsque le joueur appuie sur le bouton d'attaque, l'épée devient active et tourne dans un quart de cercle autour du personnage dans la direction du déplacement. Si le personnage n'est pas en train de se déplacer, il attaque vers la droite.
- Attribution des animations de déplacement au personnage : Lorsque le joueur ne fait rien, le personnage est dans l'animation par défaut. Dès que le joueur fait bouger le personnage, les animations de déplacement (correspondant à la direction choisie) de celui-ci se déclenchent, avec une animation de transition si le déplacement est vers la droite ou la gauche.
- Trainée derrière le personnage lorsqu'il dash.
- Animation de l'attaque avec la hache et attribution de l'animation au personnage : Similaire à l'épée.

- Animation de l'attaque spéciale avec la hache et attribution de l'animation au personnage : Similaire à l'attaque normale, mais la hache fait un, deux ou trois tour complets autour du personnage et l'animation ne dépend pas de la direction du déplacement.

Nous avons choisi de créer nous même les visuels de notre jeu, même si nous aurions pu prendre parmi les nombreux visuels gratuits trouvables sur Internet. Ce choix me convient, car il nous permet de faire ce que l'on veut. Je me suis principalement occupé des animations correspondant aux actions que j'ai implémentées dans la partie Core Gameplay. C'était plutôt amusant.

Menu Interfaces :

Voici ce que j'ai effectué au cours du semestre dans cette section :

- Création d'un support pour le HUD (Head Up Display, affichage tête haute en français) qui se situe en dehors de l'endroit où le joueur évolue et utilise une caméra autre que la caméra principale.
- Création d'une barre de vie qui se vide ou se remplit (et change de couleur) lorsque que le joueur perd ou regagne des points de vie. Elle se trouve dans le HUD.
- Création de l'écran de game over qui apparaît lorsqu'un joueur meurt. Il propose de revenir au menu principal ou de quitter le jeu.

Je n'ai pas beaucoup participé à cette section, mais j'ai trouvé mes contributions intéressantes, car elles m'ont permis de mieux comprendre comment fonctionnent l'affichage et les boutons dans Unity.



## 6.4 Meven Cassez

Voici ma partie, je suis Meven CASSEZ. Je suis dans ce projet le responsable Réseau et site web ainsi que l'assistant IA, item et équilibrage et pour finir sound design.

Voici donc un ordre de l'importance accordée à chaque partie et par conséquent du temps consacré:

- Réseau
- item équilibrage
- IA
- Site web
- sound design

Tout d'abord, après avoir mis en place les bases du gameplay avec mes collègues, c'est-à-dire, les déplacements ainsi que l'animation, les attaques et les collisions. Je me suis penché sur la mise en place du en ligne. Une fonctionnalité qui est une pièce maîtresse mais aussi une des plus complexes à mettre en place.

Elle s'est décomposée en cette liste d'étapes:

- Mise en place de l'asset photon
- liaison au serveur en ligne
- mise en place du lobby
- connection au lobby
- possibilité de créer des rooms et de les rejoindre
- mise en place de room privées
- synchronisation des joueurs et de leurs apparitions
- synchronisation des déplacements et des animations de mouvements
- mise en place de hud individuel
- synchronisation des caméras
- mise en place de l'apparition des ennemis côté serveur
- mise en place de l'affichage côté client

De ce fait, après avoir chercher de nombreux moyens d'héberger notre multijoueur, nous avons décidé d'utiliser Photon.

Nous avons ensuite mis en place le système de connexion au "lobby" du serveur ainsi que la création de "room" privées permettant au joueur de rejoindre la partie de leurs amis en communiquant le nom de celle-ci.

Après cette étape, qui fut relativement longue mais dans un certain sens peu complexe lorsque comparée à la suivante, je suis passé à la synchronisation de chaque joueur sur le serveur. Une étape quelque peu longue et fastidieuse, qui fut accompagnée d'un problème sur le serveur photon qui a mis plusieurs jours à être résolu.

La solution ?

La Whitelist de la géolocalisation du serveur qui avait des problèmes, et qui créait une partie sur un serveur en Europe mais tentait lors d'une connexion à cette partie de la rejoindre sur un serveur US. Une fonctionnalité de Photon fort utile dans certains cas mais qui fut un réel problème dans le notre.

Ce problème enfin résolu, nous sommes passés à la synchronisation des déplacements, des caméras ainsi que des inventaires.

Néanmoins, afin de mettre en place des bases solides sur le multijoueur et éviter les bugs, nous avons dû réduire le temps passé sur les IA, puisque dans la logique, si le serveur ne fonctionne pas même si notre IA, elle, fonctionne, nous ne pourrions même pas en profiter. Voici donc la répartition d'une grande partie de notre temps.

Pour la partie du développement sur les items ainsi que leur équilibrage, nous avons commencé par la recherche d'un algorithme de type "randomizer" afin de pouvoir mettre en place des pourcentages d'obtention des items.

Après cette partie majeure, nous avons implémenté des objets afin d'aider notre joueur dans son aventure.

Ainsi, le joueur pourra trouver durant sa quête des potions de soins et d'autres objets spéciaux. Ceux-ci sont pour le moment obtenables via des coffres. Pour plus de détails, je vous laisse le soin de lire la partie de Jean Béchet dans laquelle il fournit de plus amples détails.

Pour la partie sur l'IA nous avons pour le moment de nombreuses recherches et algorithmes en préparation mais nous avons déjà implémenté l'algorithme de "following" des ennemis.

Et pour finir, la dernière partie sur laquelle nous avons travaillé sur cette période : Le site web. Celui-ci est pour le moment du même niveau que celui d'un site web moyen dans les années 1990 mais il est voué à évoluer rapidement.

Nous avons pour le moment mis en place les différentes pages ainsi que le header présent sur toutes celles-ci et permettant de naviguer au travers des différentes pages. Il va dans un premier temps regrouper nos rapports de soutenances, dont celui-ci et cela dès la seconde soutenance.

Passons maintenant au progrès de la deuxième soutenance.

Ce qui a été réalisé:

- Mise en place système de pseudonymes
- Changement dans le lobby
- Liste de salles en temps réel
- Mise en place des rooms avant lancement
- Lancement de partie uniquement pour le leader
- Fix bugs :
  - Superpositions hud
  - Health bar fonctionnel
  - Fix drop d'objet
  - Mob inflige dégâts
  - Fix animation d'attaques simultanées
  - (à fix) animation d'attaque au moment du drop d'items
  - Amélioration du site web

Pour commencer, j'ai commencé par re-travailler la totalité des écrans de chargements et du multijoueur. En commençant par la mise en place d'un système de pseudonymes ainsi que l'écran qui se joint à ce système.

Puis j'ai complètement modifier le lobby. La création de salle utilise toujours la même procédure. Néanmoins, pour rejoindre les salles, tout a changé. Il y a maintenant une liste des salles actuellement ouvertes. Il suffit de cliquer sur une salle affichée dans la liste afin de la rejoindre. Avant le lancement de la partie, tous les joueurs se retrouvent dans un lobby et uniquement le leader de la salle (personne qui l'a créée peut lancer la partie). Une fois celle-ci lancée, les joueurs rejoignent directement la partie.

Au lancement, les joueurs apparaissent sur la carte pendant que la carte se génère aléatoirement.

Le reste de mon temps fût consacré à la résolution de bugs.

J'ai commencé par résoudre les problèmes de hud (superposition non voulue) ainsi que la barre de vie qui peut enfin diminuer.

Le ramassage d'objet a également été fixé évitant que l'objet se place dans l'inventaire des deux joueurs alors qu'un seul fût ramassé. Nous avons également résolu les problèmes liés aux mobs qui ne pouvaient pas attaquer.

Un autre fix concernant les animations cette fois-ci, les animations d'attaques ne se lancent plus sur les deux joueurs en même temps ainsi que lorsqu'on clique pour dropper un item. En plus de cela, nous avons ajouté les animations de déplacement sur les mobs.

Pour finir, un travail sur le site web s'est imposé afin de continuer à le développer et suivre l'avancée du projet.

Maintenant, qu'est ce qu'il s'est passer depuis la seconde soutenance.

- ajout selection apparence du personnage
- rework du déplacement des ennemis qui posait problème

- ajout (fix) des sons de pas
- ajout de quelques items
- Amélioration du site web
- Fix bugs :
  - potion de heal qui ne fonctionnait pas
  - fix ennemis qui passe à travers les murs
  - fix des effets d'objets qui créaient des bugs et ne fonctionnait en fait pas
  - Fix les morceaux de code non fonctionnel de Jean et dépit de repos

En somme, en dehors de quelques implémentations d'objets ou fonctionnalités, j'ai passé la majeure partie du temps à fixer les bugs déjà existant ainsi qu'à merge proprement les morceaux de code de mes camarades au projet (Chose plus facile à dire qu'à faire).

## 7 Ce qu'on en a pensé:

### 7.1 Erwan Doussin

Le développement d'un petit jeu pareil m'a profondément intéressé en tant qu'étudiant en ingénierie informatique pour plusieurs raisons. Tout d'abord, cela me permet de mettre en pratique mes connaissances en programmation et de consolider mes compétences. Créer un jeu complet implique de maîtriser divers concepts tels que la gestion des entrées, la logique du jeu, les collisions, l'animation, l'intelligence artificielle et bien plus encore. En travaillant sur un projet concret comme un jeu, je peux appliquer ces concepts de manière cohérente et les voir prendre vie.

### 7.2 Jean Béchet

Développer RogueLite présente un certain niveau de complexité qui stimule mon esprit de résolution de problèmes. La génération procédurale des niveaux, la gestion des objets et des ressources, la conception des mécanismes de progression et la prise en compte de la difficulté sont autant de défis intéressants qui nécessitent une réflexion approfondie et créative. Cela m'aide à développer mes compétences d'analyse et de conception de systèmes complexes.

### 7.3 Meven Cassez

En travaillant sur un projet de jeu, j'ai également eu l'occasion d'explorer de nouveaux domaines de programmation, tels que les moteurs de jeu, les bibliothèques graphiques et les outils de développement. Cela m'a poussé à étendre mes connaissances et à m'adapter à de nouvelles technologies, ce qui est essentiel dans le domaine de l'ingénierie informatique en constante évolution.

Développer un tel jeu m'a également permis d'améliorer mes compétences en collaboration et en gestion de projet. La création d'un jeu nécessite souvent le travail en équipe, que ce soit avec des artistes pour les graphismes, des concepteurs pour les mécaniques de jeu ou des testeurs pour les retours utilisateurs. Apprendre à communiquer efficacement, à coordonner les efforts et à respecter les délais est une expérience précieuse qui m'aidra dans ma future carrière en tant qu'ingénieur informatique.

## 7.4 Thibaud Guignon

Le développement d'un RogueLite en 2D m'a passionné et a été une expérience enrichissante. Il m'a permis d'appliquer mes connaissances en programmation, de relever des défis stimulants, d'explorer de nouveaux domaines technologiques et d'améliorer mes compétences en collaboration et en gestion de projet. Je suis convaincu que cette expérience m'aidera à progresser en tant qu'étudiant en ingénierie informatique et à me préparer pour les défis futurs de ma carrière.