

# RAPPORT DE SOUTENANCE 2

Avril 2022

Alexandre Agostinho, Sébastien Combe, Nathan Fontaine, Dimitri Brancourt

## Avant-propos

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



FIGURE 1 – Logo (temporaire)

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Rappel du jeu . . . . .	3
1.2	Rappel général de ce qui a été accompli à la 1ère soutenance	3
1.3	La team frelu . . . . .	3
1.4	Avancement à la Soutenance 1 . . . . .	4
1.5	Avancement pour la Soutenance 2 . . . . .	4
1.6	Tableau de l'estimation de l'avancement du Projet . . .	5
<b>2</b>	<b>Avancement du projet</b>	<b>6</b>
2.1	Menus d'affichages - Nathan . . . . .	6
2.1.1	Menu Options . . . . .	6
2.1.2	Menu d'achat . . . . .	7
2.1.3	Inventaire . . . . .	7
2.2	Graphismes - Nathan/Sébastien . . . . .	8
2.2.1	Map - Nathan . . . . .	8
2.2.2	Cycle jour/nuit - Sébastien . . . . .	8
2.2.3	Gestion de lumières - Sébastien . . . . .	9
2.2.4	Sprites - Sébastien . . . . .	10
2.2.5	Game Design - Sébastien . . . . .	10
2.3	Interactions - Dimitri . . . . .	11
2.3.1	Rappel . . . . .	11
2.3.2	Déplacements avancés . . . . .	11
2.3.3	Inventaire . . . . .	11
2.3.4	Sauvegarde . . . . .	12
2.3.5	Magie . . . . .	12
2.3.6	Boss - The Handspinner . . . . .	13
2.3.7	Pièges . . . . .	13
2.3.8	futur . . . . .	14
2.4	Le Multijoueur - Alexandre . . . . .	15
2.4.1	Description . . . . .	15
2.4.2	Implémentation . . . . .	15
2.4.3	Difficultés . . . . .	17
2.5	Sound Design - Nathan . . . . .	17
2.5.1	SFX . . . . .	17
2.5.2	Musique . . . . .	17
<b>3</b>	<b>Conclusion</b>	<b>18</b>
<b>4</b>	<b>Bibliographies</b>	<b>18</b>

# 1 Introduction

## 1.1 Rappel du jeu

Notre projet est un jeu plateforme 2D de style pixelart développé à l'aide de Unity. L'objectif du jeu est de parcourir la map, de s'améliorer en achetant des items et de nouvelles armes, afin de tuer certains boss qui lui barrerons le chemin pour à terme permettre au joueur d'atteindre le royaume céleste.

## 1.2 Rappel général de ce qui a été accompli à la 1ère soutenance

Lors du premier rendez-vous avec les ACDC, il nous avait conseillé de ne pas nous focaliser sur le multijoueur, c'est ce que nous avons fait et suivie mais malheureusement c'est également ce qui nous a été reproché lors de la première soutenance, en effet Alexandre développera se point qui à été une négligence de notre part.

Nous avons beaucoup avancé sur les mouvements et interfaces graphiques du jeu lorsque nous avons passé notre premier oral, Le système de dégâts, de mort, de victoire et d'ennemies étaient dès lors déjà implémentées et il ne manquait plus qu'un boss pour avoir un jeu simple et fonctionnel qui peut sauvegarder, commencer un monde et le finir, ce qui le rendrait complet et quasi finie. Cependant, il reste encore des optimisations, des ajouts à mettre en place ainsi qu'un gros travail de graphisme pour pourvoir finir notre jeu dans les temps.

## 1.3 La team frelu

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

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

R : responsable, S : suppléant

## 1.4 Avancement à la Soutenance 1

Pour rappel, voici l'état d'avancement du projet à la première soutenance

Tâches	Alex.A	Dimi.B	Séba.C	Nath.F	Total
Objects Design	0%	5%	0%	0%	5%
Mobs/Characters Design	0%	0%	0%	5%	5%
Game Design	0%	0%	2%	0%	2%
GUI Design	0%	0%	0%	2%	2%
GUI Programing	0%	10%	0%	50%	60%
Gameplay Programing	5%	40%	0%	5%	50%
Multiplayer	0%	0%	0%	0%	0%
IA Programing	20%	0%	0%	0%	20%
Website	25%	0%	0%	0%	25%
Sound Design	0%	0%	0%	5%	5%

## 1.5 Avancement pour la Soutenance 2

Voici l'état d'avancement actuel du projet

Tâches	Alex.A	Dimi.B	Séba.C	Nath.F	Total
Objects Design	0%	30%	0%	10%	40%
Mobs/Characters Design	0%	40%	0%	20%	60%
Game Design	0%	0%	60%	5%	65%
GUI Design	0%	0%	0%	50%	50%
GUI Programing	0%	10%	0%	70%	80%
Gameplay Programing	5%	60%	0%	5%	70%
Multiplayer	60%	0%	0%	0%	60%
IA Programing	50%	0%	0%	0%	50%
Website	25%	0%	0%	0%	25%
Sound Design	0%	0%	0%	20%	20%

## 1.6 Tableau de l'estimation de l'avancement du Projet

Soutenance	1 estimation	1 réel	2 estimation	2 réel	3 estimation
Objects Design	20%	5%	80%	40%	100%
Mobs/Characters Design	10%	5%	40%	60%	100%
Game Design	0%	2%	50%	65%	100%
GUI Design	0%	2%	50%	50%	100%
GUI Programing	50%	60%	80%	80%	100%
Gameplay Programing	40%	50%	75%	70%	100%
Multiplayer	0%	0%	50%	60%	100%
IA Programing	10%	60%	60%	50%	100%
Website	20%	25%	60%	25%	100%
Sound Design	0%	5%	10%	20%	100%

Nous avons dans l'ensemble plutôt bien avancer sur nos objectifs initiaux mais nous avons un peu sur-estimé notre capacité à avancer dans l'ensemble du projet par rapport aux estimations réalisées lors du cahier des charges et de la première soutenance car la création design prend énormément de temps s'il nous voulons tout faire pas nous même.

Il n'y a que le site web pour lequel Alexandre n'a pas beaucoup avancé car il avait beaucoup de problème avec l'implémentation du multijoueur

## 2 Avancement du projet

Nous allons maintenant vous présenter plus en détail ce que nous avons fait et comment nous prévoyons la suite du projet.

### 2.1 Menus d'affichages - Nathan

Lors de la première soutenance, la plupart des menus d'affichages ont déjà été mis en place, cependant il restait néanmoins l'implémentation en terme de script du menu options à faire et cela à été plus difficile que prévu, en effet dans ce menu, il faut permettre à l'utilisateur d'interagir avec les boutons pour changer les touches du jeu, les options graphiques ou encore certains paramètres audio.

#### 2.1.1 Menu Options

Initialement j'avais prévu de décomposer le menu options 4 parties : Les boutons "video", "audio", "help" et "commands". Finalement, nous avons décider de le décomposer seulement en 3 parties afin de mettre la documentation d'aide sur le site web.

Le menu options comme dit précédemment à été difficile à implémenter surtout la manière dont on change les touches mais finalement après avoir longuement recherché sur la documentation et des forums j'ai finalement réussi à mettre en place ce système.



FIGURE 2 – Aperçu du Menu Options

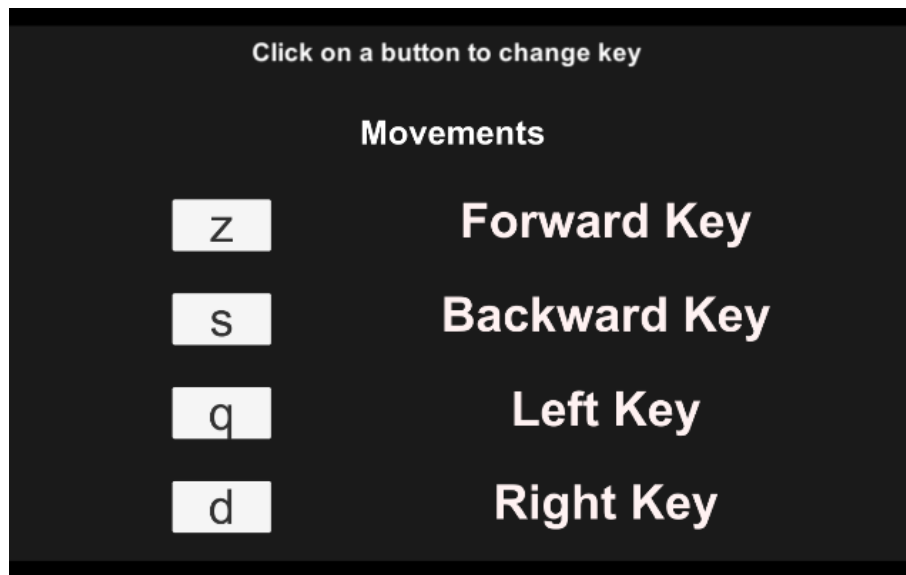


FIGURE 3 – Aperçu des commandes

Pour utiliser ces menus, il suffira de suivre les indications en haut de l'écran et lorsque l'on clique sur un bouton. Par exemple pour le menu commande, le joueur doit cliquer sur la touche visuelle en gris puis cliquer sur une touche de son clavier pour pouvoir faire le changement effectif de la touche en jeu. Le joueur pourra également utiliser le scroll de la souris pour pouvoir découvrir de nouveaux paramètres situés plus bas.

### 2.1.2 Menu d'achat

Nous voulons mettre en place une boutique pour pouvoir s'acheter de l'équipement avec des pièces et c'est sur quoi nous travaillerons pour la soutenance finale.

### 2.1.3 Inventaire

Nous avons déjà mis en place un système d'inventaire mais celui-ci n'était pas un menu (c'est un HUD de l'écran de jeu), c'est pourquoi nous ne nous focaliserons pas là-dessus et nous ne l'implémenterons qu'en bonus.

## 2.2 Graphismes - Nathan/Sébastien

En ce qui concerne les graphismes nous n'avons plus qu'à implémenter les animations d'attaques des différents personnages et des monstres ce qui s'avèrera plutôt long pour le personnage principal étant donné que l'on doit créer une animation pour chacun des items que l'on a créé. De plus il faudra par la suite remplacer les graphismes des textes et des menus pour les mettre un peu plus dans le style du jeu.

C'est pourquoi, il faudra également remettre à jour le logo car nous l'avons fait sans idée très claire et précis de ce que nous allions faire.

### 2.2.1 Map - Nathan

Avec l'aide de Sébastien, nous avons créer et finie le tuto du jeu qui permettra au joueur de comprendre les touches du jeux ainsi que de pouvoir de les changer selon sa volonté.

De plus, nous avons également finie l'implémentation du premier niveau avec un boss quasiment opérationnel.

Pour l'instant, notre tutoriel et niveau principal ont été créer à la main à l'aide de Gimp et nous les avons utiliser comme calque pour pouvoir créer ces maps avec tous les éléments.

Cependant, je suis entrain de créer un système de génération procédurale (comme en NTS (mauvais souvenir)) pour pouvoir générer une cave ou une map normal selon les paramètres choisies.

Cela nous permettra d'une part de pourvoir créer les plusieurs niveaux que nous désirions à l'origine et nous permettra d'avoir un jeu plus complet, ce sera également un gain de temps non négligeable pour la suite de la création de la map.

### 2.2.2 Cycle jour/nuit - Sébastien

Cette partie sera centrée sur le cycle jour/nuit, de la gestion des lumières et d'un peu de pixel art sur le fin

Bonne nouvelle, le cycle jour/nuit est maintenant opérationnel, cependant ce petit cycle jour/nuit a posé pas mal de problèmes

En effet, le principal problème du cycle jour nuit était causé par la sauvegarde. En fait, lorsque l'on sauvegardait en pleine nuit et que l'on relançait le jeu, l'heure était bien sauvegardé mais pas la luminosité, on se retrouvait donc en plein jour alors qu'il était 3h du matin



Finalement ce problème n'était pas si compliqué à résoudre, il suffisait de sauvegarder cette fameuse luminosité dans un variable et le tour était joué, ou presque. Il fallait maintenant toucher au scripts de sauvegarde, cependant ce n'est pas moi qui ai codé la sauvegarde, il a donc fallu que je fouille un peu dans ces fameux scripts, avec Dimitri qui lui a codé la sauvegarde, pour trouver le problème et pour enfin sauvegarder cette luminosité. Le problème est maintenant résolu et le cycle jour/nuit marche parfaitement bien, et ce, avec la sauvegarde. La prochaine étape seras de faire apparaitre plus de monstres la nuit ou des les rendre plus agressif, mais ça c'est une autre histoire.

### 2.2.3 Gestion de lumières - Sébastien

Parlons maintenant de la gestion des lumières, étant donné que je m'occupe du cycle/nuit il est normal que je m'occupe des lumières. La gestion des lumières était, est, et sera sans doute toujours un grand problème dans les jeux-vidéo, même si notre jeu est en 2d, la gestion des lumières reste pénible à gérer.

Tout d'abord plaçons correctement nos lumières et faisons en sorte qu'elles éclairent correctement les objets aux alentours, unity gère cela et pourtant cela reste une plaie et gérer les lumières pose souvent des bugs (tout les sprites au alentours deviennent noirs par exemple).

Bon nos lumières sont correctement placés et que l'éclairage est bien réglé, il faut maintenant réglé le rayon d'éclairage et voilà, nos lumières sont bien jolies mais elles ne servent à rien sans habillage, attachons nos lumières à des sprites.

Nos lumières sont maintenant bien placés et opérationnelles mais elles sont tout le temps actives, et oui nous avons un cycle jour/nuit il faut donc que nos lumières soient éteintes la jour mais allumés la nuit.

Et pour cela il "suffit" de retourner dans notre script du cycle jour/nuit, faisons en sorte de mettre nos lumières dans une liste et de les activer à une certaine heure. Et ça fonctionne bien, la transition est encore un peu brutal mais nos lumières ne sont plus actives le jour. Et encore une fois c'est la sauvegarde qui pose problème, comme pour le cycle jour/nuit lorsque que nous sauvegardons alors que les lumières sont actives et que nous relançons le jeu, les lumières ne sont plus actives alors qu'il fait nuit. Ce problème a déjà été rencontré il suffit donc de le résoudre

de la même façon.

Au final, la gestion des lumières est assez pénible et est le plus souvent source de problèmes.

#### **2.2.4 Sprites - Sébastien**

Comme dit plus tôt, il nous faut des sprites pour nos lumières et pour cela il nous faut encore une fois faire appel à notre fibre artistique.

Outre le feu pour la sauvegarde (créer par Nathan), il nous faut des sprites de lampes ou de torches, je me suis donc attelé à la tâche et ai créer des sprites de lampes, allumée et éteinte et d'une torche.

Cette partie ne demande pas plus de développement et ne m'a pas posé plus de difficultés que ça, dans le futur ces sprites seront utilisés pour le cycle jour/nuit. et permettront de donner une plus value à la map

#### **2.2.5 Game Design - Sébastien**

J'ai mis en place les éléments du jeu pour créer la map, comme les sprites avaient déjà été créer lors de la 1ère soutenance, je n'avais plus qu'à les organiser sur la map pour pouvoir créer un jeu intéressant qui donnerait du challenge au joueur.

## 2.3 Interactions - Dimitri

La partie Interactions sera centrée sur le Gameplay programing, cette partie englobe donc les actions du personnages et ses interactions avec d'autres objets ...

### 2.3.1 Rappel

Je vais rapidement reparler de ce qui avait été réalisé à la dernière soutenance. Nous avons parlé de la prise en mai de Unity, d'à quel point cet outil est puissant mais aussi complexe. Nous avons aussi parlé de la prise en mai de la programmation dans Unity, donc notamment de la gestion de la bibliothèque de Unity.

Les déplacements du personnage avaient déjà été pris en charge. Il y a aussi eu la gestion de la barre de vie et des dégâts, de la deathzone, des checkpoints, des portails et enfin des coffres.

### 2.3.2 Déplacements avancés

Il y a maintenant deux nouvelles options de déplacements. La première est le dash. Cette option connue dans les jeux vidéos permet au personnage de faire un déplacement rapide sur un court laps de temps. La principale difficulté de ce déplacement étant de le rendre fluide et de lui ajouter un délai.

Enfin le deuxième et le plus complexe est le double saut. Il y avait deux principales difficultés : n'avoir qu'un double saut, et non pas des sauts à l'infini, et de faire un vrai saut, car il y avait des problèmes de gravité qui rendait le deuxième saut négligeable par rapport à la vitesse de chute.

### 2.3.3 Inventaire

Une partie de l'inventaire à aussi été implémentée. Il y a maintenant 4 items "consommables" : une potion qui fait regagner un coeur, une potion qui fait gagner 2 de l'armure (qui prend les dégâts à la place des points de vie du joueur), une potion d'invincibilité, qui permet d'annuler les dégâts du prochain coup reçu par le joueur, et enfin le dernier objet permet de ressusciter (une fois évidemment).

pour le coup la difficulté était de gérer le nombre d'objets et leur affichage sans perturber le reste du jeu (notamment la sauvegarde). Bien sûr il y avait aussi de petites difficultés au niveau de la gestion de l'invincibilité (pour qu'elle soit unique) et le fait de ressusciter une fois donc sans et sans devenir immortel, cela va de soi.

Une autre difficulté était d'implémenter les touches permettant de changer d'items et de mettre un effet de rotation (possible avec des modulo).

#### **2.3.4 Sauvegarde**

Un gros point important qui à été mis en place est la sauvegarde. Son système est très simple, chaque fois que le joueur appuis sur E près d'un feu de camp, une sauvegarde est effectuée. La sauvegarde est effectuée sous format de fichiers json.

Cela signifie bien entendu que les sauvegardes sont chargés si elles existent au lancement même de la partie. Cela signifie que les informations de santé du joueur sont chargés, les informations sur la position aussi, le joueur est donc replacé au feu de camp correspondant.

Il y a aussi les informations de l'inventaire, donc l'argent et les items que le joueur a en sa possession. Et enfin les informations de l'heure de la journée avec la luminosité et les dégâts, qui changeront avec le futur équipement.

Les difficultés ici étaient les problèmes de conflit entre nouvelle information et ancienne information, par exemple lorsqu'on met à jour une information à haute fréquence et donc qu'elle écrit par dessus le nouvelle qui à été chargée par le biais de la sauvegarde.

l'autre difficulté s'apparente aussi à des conflits dans le programme, cette fois ci c'est au niveau du remplacement du joueur (qui ne se remplaçait pas)

#### **2.3.5 Magie**

Un système de magie a été mis en place. Il y a donc du mana, qui se gagne petit à petit et permet de lancer des sorts. Il n'y a pour l'instant que deux sorts : un bouclier et une boule de feu.

Pour l'instant les dégâts du joueur se résument à la boule de feu. La boule de feu se lance au niveau du personnage et se déplace en ligne droite devant lui. La difficulté était de la faire disparaître au contact de certains objets ou murs et de faire des dégâts au boss et aux ennemis. Bien sûr il fallait aussi vérifier si le mana est suffisant et le mettre à jour

Le bouclier était bien plus dur, il fonctionne en continu jusqu'à ce que le joueur appuie sur le bouton d'activation ou n'ait plus le mana nécessaire.

Ici la plus grosse difficulté était de faire arrêter le bouclier correctement.

### **2.3.6 Boss - The Handspinner**

Le boss à été implémenté de la même manière que les mobs basiques, il cherche le joueur et se déplace pour le chercher, ici il reste quand même à une distance prédéfinie. Il va ensuite attaquer à distance via les même calculs que pour le déplacement. rentrer en collision avec lui fait aussi prendre des dégâts au joueur

Ici la difficulté était de le faire déplacer correctement et de le faire attaquer précisément. Une autre difficulté était de gérer ses différents collider.

### **2.3.7 Pièges**

Enfin la dernière partie est la partie sur les pièges. Il y a une pléthore de pièges maintenant. Le premier piège et le plus basique et le pic. Ce piège est en fait une zone dans laquelle le joueur prend des dégâts en rentrant dedans et en y restant.

Ici le problème était de faire en sorte que le joueur se prenne des dégâts que en étant dans le piège et pas trois secondes après en être sorti.

Un autre piège est le champignon, ce piège ignore l'armure mais fonctionne de la même manière que les pics, le joueur prend des dégâts en restant et en entrant dans la zone du piège.

Il y a aussi un interrupteur, quand le joueur marche dessus des lances apparaissent et se déplacent vers le joueur.

Ici la difficulté était de faire activer le piège lorsque le joueur passe dessus.

Un autre piège est le canon, il envoie le boulet jusqu'à un point prédéfini et déplaçable. La boule de feu annule le boulet.

La difficulté ici était de diriger le boulet.

Le dernier piège est une boule avec des piques qui peut se déplacer en fonctions d'un nombre variable de points différents. Ici la difficulté était de gérer correctement les listes de points.

### **2.3.8 futur**

Pour l'avancement de cette partie il ne manque plus que la création d'une boutique et d'items d'équipements, qui ont déjà été prévus, il ne manque plus que le design et le choix des statistiques. Il manquera aussi des items consommables des coffres différents et des sorts différents, mais tout ceci se basera sur ce qui à déjà été fait.

## 2.4 Le Multijoueur - Alexandre

### 2.4.1 Description

Bien que de base les jeux présentant une dimension narrative ne nécessitent pas forcément une partie multijoueur, il peut tout de même être intéressant de pouvoir taper sur des ennemis avec des amis ou encore faire une course à celui qui aura tué le plus d'ennemis en un temps imparti. C'est en partie la raison qui explique que notre partie multijoueur se focalisera sur une map spécialement conçue pour, où des vagues de monstres arriveront et où le but sera de survivre le plus longtemps possible. Si possible, nous essayerons d'implémenter d'autres mods de jeux en multi, mais puisqu'il s'agit d'une partie mineure de notre jeu, cela passera en second plan.

### 2.4.2 Implémentation

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

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

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

Une fois arrivé dans la room, les joueurs et les ennemis sont instanciés et les informations concernant la position et l'état des entités présentes sur la scène de jeu sont émises et reçues par tous les membres connectés.

à la room. Par exemple, quand le joueur se déplace, les valeurs de sa position sont envoyés sur le serveur et redistribués aux autres membres du serveur. De cette manière chacun peut savoir graphiquement où se situe les autres personnes présentes avec lui et l'état de leur santé par exemple.

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

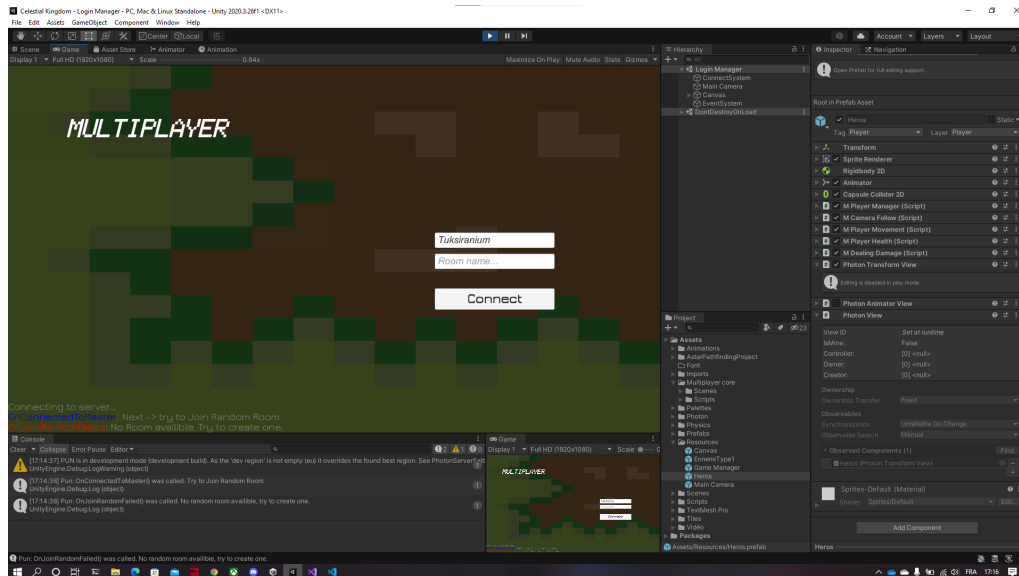


FIGURE 4 – Aperçu de la page de login quand aucune room n'existe et que on essaye de se connecter sans mentionner de room.



FIGURE 5 – Aperçu du jeu multijoueur sur le terrain d'essai.



### 2.4.3 Difficultés

Lors de l'implémentation, nous avons rencontré deux grosse difficultés.

La première était que beaucoup des fonctionnalités multijoueur ne sont pas compatible avec celles du monojoueur (en tout cas, de la manière dont on l'a implémentée). Cela a fait que beaucoup de scripts ont dû être réécrit (en partie ou dans leurs intégralités). Notre manière d'initialiser les objets aussi a dû être modifiée. Dans la partie solo, on initialise les ennemis et le joueur directement dans la composition de la scène, alors que dans la partie multijoueur, on doit créer les entités dans le code avec l'API de Photon pour que les données soit retransmises à tous les joueurs. La modification des scripts avait pour but aussi la synchronisation des données avec tout le monde.

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

## 2.5 Sound Design - Nathan

### 2.5.1 SFX

En termes d'avancement sur les effets sonores, il ne reste plus qu'à rajouter tous les objets du jeu pour que je puisse les implémenter un par un avec un effet stéréo dans le jeu. J'ai mis sur un google drive tous les effets sonores nécessaires que nous utiliserons dès lors que tout sera fini donc a priori cette catégorie sera terminée d'ici la prochaine soutenance.

### 2.5.2 Musique

En ce qui concerne la musique, j'ai déjà fait plus de la moitié des musiques que je comptais faire, nous avons déjà une musique pour les combats de boss, une autre musique que l'on pourra utiliser n'importe où dans le jeu. Il ne restera plus qu'à implémenter le système de lecture de son pour jouer une musique à un temps donné.

### 3 Conclusion

Malgré tout les problèmes que nous avons eu le plus dur a déjà été rajouter dans le jeu Merci d'avoir eu le courage d'être arrivé jusqu'ici et d'avoir lu notre rapport de soutenance, nous espérons que vous appréciez notre projet et notre avancement, et nous vous invitons à nous donner des conseils pour améliorer quoi que ce soit.

Nous vous prions d'agréer nos plus sincères salutations.  
La team FRELU ;)

### 4 Bibliographies

- [Notre organisation Trello](#)
- [Le site web](#)
- [Notre github du site web.](#)
- [Notre gitlab du projet](#)