

Rapport de soutenance 2
Réalisé lors du S2 de la promo-2021

Jérémie "Cactus" BOURDET
Victor "Xarang" COATALEM
Florian "Dope" RAKOTOARIVONY
Hadrien "Hawk" NAVARRO

14 juin 2017

Les Faltoquets & cornifleurs

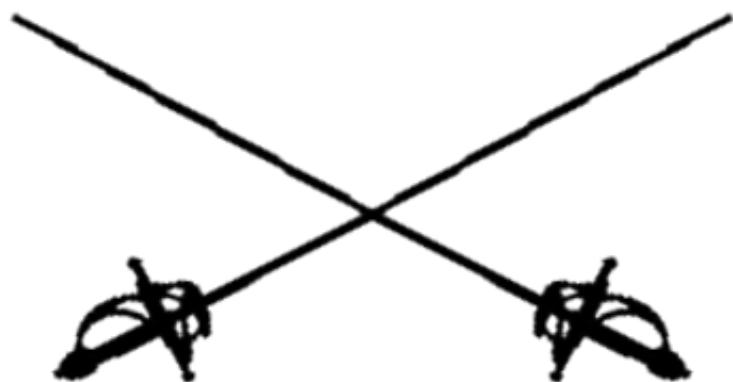


Table des matières

1	Introduction	3
2	Rapide retour sur le cahier des charges	4
2.1	Le principe de Mulligan	4
2.2	Synopsis	4
2.3	Notre répartition des tâches/planning	5
3	La réalisation du projet	6
3.1	Le multijoueur	6
3.2	Le système de combat	8
3.3	Les graphismes	9
3.4	Les animations	9
3.5	Le site web	10
3.6	L'interface	11
3.7	Le plateau	12
3.8	Les sons	13
4	Prochaines étapes	14
5	Annexe : aperçus du jeu	16
6	Conclusion	17

1 Introduction

Bienvenue dans le deuxième rapport de soutenance du projet Mulligan réalisé par des sup de la promo 2021.

Nous allons aujourd’hui vous présenter nos avancées sur le projet, c’est à dire tout ce que nous avons réalisé à ce jour, ainsi que nos prochains objectifs. Nous en sommes à plus que la moitié de l’année, et donc de l’avancement du projet, il est donc temps de faire un bilan de ce qui a été réalisé sur le projet Mulligan depuis la fin de la première soutenance.

Mais pour bien voir l’étendu du travail que nous avons effectué, il est important de récapituler ce qui avait été fait pour la première soutenance.

2 Rapide retour sur le cahier des charges

2.1 Le principe de Mulligan

Mulligan est en fait un rogue-like, dans lequel nous incarnons un héros qui doit réussir à sortir d'un donjon en résolvant des énigmes et en vainquant des monstres de plus en plus forts.

Mulligan est un RPG teinté d'éléments de jeu de cartes dans lequel le joueur progresse à travers les étages d'un vaste donjon, dont la structure est représentée par des cartes. Il doit alors sortir victorieux des péripéties qui se dévoileront devant lui au fur et à mesure que les cartes se révéleront.

L'aventurier, représenté par un pion, se déplacera alors de carte en carte à la recherche de l'escalier menant à l'étage suivant. Ce dernier sera cependant bien défendu par une bête monstrueuse que l'on combattrra alors en vue à la première personne, équipé des cartes 'équipement' que l'on aurait obtenu le long de notre trajet et que l'on aurait alors dans notre main. Les péripéties proposeront au joueur différentes résolutions, ce dernier devra alors user de sa perspicacité pour comprendre de lui même quelle est la décision la plus avantageuse pour lui, et décider en fonction, à la manière d'un jeu de rôle papier. Au cours de ces péripéties, le joueur dépensera des ressources (ex : nourriture, or, moral..), en acquerra d'autres, dénichera de précieux trésors l'aident dans ses combats et fera de surprenantes rencontres..

De plus, au fur et à mesure des parties, il apprendra à mieux connaître les différentes cartes du donjon et évitera les pièges dans lesquels il était tombé auparavant. La mort n'est pas un échec, mais une progression. A chaque partie les connaissances du joueur s'enrichissent et lui permettront d'aller plus loin à la partie suivante. De plus, tous les étages du donjon sont générés aléatoirement : le jeu est fait pour être joué à de nombreuses reprises.

2.2 Synopsis

Au début de l'aventure, le joueur, un aventurier, entre dans une taverne pour se remettre de sa dernière escapade. Il y rencontre un voyant, qui lui propose alors de lire son avenir dans les cartes de tarot. Vous acceptez alors, en lui tendant une bourse d'or dérobée plus tôt dans la journée. Ce dernier disposera alors les cartes sur la table, qui formeront le premier étage du donjon, la partie peut alors commencer...

2.3 Notre répartition des tâches/planning

Tâche/personne	Jérémie	Hadrien	Victor	Florian
Menus		X		0
Variété des cartes			0	X
Gestion du donjon			X	0
Animations	X		0	
Modèles 3D	X			0
Sons		0		X
Système de combat	0	X		
IA		0	X	
Multijoueur	0		X	
Leaderboard		X		0
Site web	X		0	

X=Titulaire 0=Suppléant

Tâche/Soutenance	1 ^{ère} soutenance	2 ^{ème} soutenance	3 ^{ème} soutenance
Menus	**	***	***
Variété des cartes	*	**	***
Gestion du donjon	**	**	***
Animations	*	**	***
Modèles 3D		*	***
Sons		*	***
Système de combat		**	***
IA		**	***
Multijoueur	*	*	***
Leaderboard	*	**	***
Site web	**	***	***

3 La réalisation du projet

Un projet tel que le notre requiert non seulement une masse de travail importante mais surtout une bonne organisation et une bonne répartition du travail entre nous 4. C'est pourquoi nous avons délimité différents axes de travail et différentes étapes.

3.1 Le multijoueur

Conformément à ce qui avait été prévu dans notre premier rapport de soutenance, nous avons réalisé le multijoueur avec l'asset gratuit PUN(Photon Unity Networking). Actuellement, plusieurs joueurs peuvent se connecter à une même partie dans laquelle ils incarnent le personnage du mode solo et peuvent se frapper les un les autres. Nous allons maintenant vous expliquer comment nous avons procédé.

- La première étape était de connecter simplement plusieurs ordinateurs. Pour cela nous utilisons dans un script des fonctions de base du PUN telle que "OnPhotonRandomJoinFailed" ou "OnJoinedLobby". Le principe est simple, au lancement du programme, le client va se connecter au serveur de Photon puis tenter de rejoindre une salle aléatoire. Si ce n'est pas possible, c'est à dire qu'il n'existe aucune salle, alors il va en créer une puis la rejoindre.

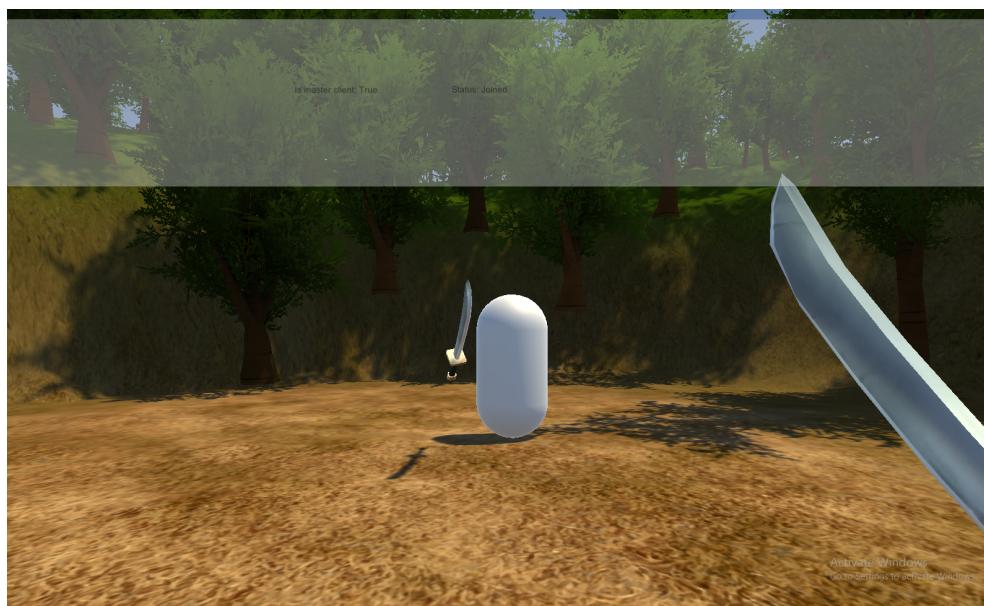
La différence majeure entre le multijoueur et le solo est qu'un certain nombre d'éléments doivent dépendre du nombre de joueurs connectés (un personnage par joueur par exemple), il n'est donc pas possible de disposer de tout dans la scène de départ. Au contraire, la scène est plutôt vide au début et à chaque connexion, le prefab d'un personnage est instancié dans la scène.

Lorsqu'on parle de multijoueur, on devrait plutôt dire multijoueur simulé car ce n'est pas comme si les différents joueurs étaient présent dans la même scène, pour l'instant il y a bien autant de personnages que de joueur mais les actions ne sont pas synchronisées, c'est à dire que si un joueur se déplace dans son client, les autres ne le verront pas. Pour rendre la synchronisation possible, chaque client va régulièrement récupérer les déplacements des autres joueurs, corriger leurs positions et envoyer par le réseau sa position et sa rotation. Pour faire la différence entre le vrai joueur qui se déplace avec la souris et le clavier et les autres qui ne sont que des prefabs vident simulant un vrai joueur, le personnage possède un Photon view ce qui me permet d'utiliser la fonction IsMine.

Nous avons rencontré beaucoup de difficultés avec cette synchronisation. Au début on ne transmettait que le transform du personnage aux autres clients, ce qui faisait des saccades, on a donc utilisé la fonction Lerp de la classe Vector 3 pour rendre les déplacements plus fluides. Comme nous utilisons le personnage du mode solo, nous avons été obligé de l'adapter un peu pour résoudre des bugs quand plusieurs joueurs étaient présents. Par exemple les caméras ne suivaient pas le bon personnage ou encore les animations d'attaque n'étaient pas synchronisées...

Bref nous avons fait face à énormément de problèmes inattendus. Pour finir, nous allons expliquer comment les dégâts sont appliqués en multijoueur. D'abord la détection se fait exactement de la même manière qu'en solo, les joueurs possèdent un script Target avec un attribut Life qui diminue quand une collision avec un objet avec le tag "weapon" se produit. Cependant, pour que toute les instances du jeu mettent bien à jour les points de vie, nous faisons un appel RPC (Remote Procedure Call) à toute les autres instances pour faire baisser les points de vie de ce joueur(L'identification se fait grâce à l'id du Photon view) Un appel RPC est un appel d'une certaine méthode sur tout les clients connectés dans la même room.

Pour éviter qu'un joueur puisse se faire "one shot", après avoir pris des dégâts, un joueur devient invisible pendant un petit laps de temps et il devient rouge pendant ce temps.



3.2 Le système de combat

Nous avons désormais un module de déplacement en vue à la première personne dans un monde en 3 dimensions.

Ce module est constitué de deux scripts : un gérant les déplacements du joueur, et l'autre s'assurant que la caméra suive le curseur de la souris. Le joueur peut effectuer un dash, c'est à dire se déplacer rapidement dans une direction, ce qui améliorera le dynamisme des combats. Nous avons également créé une arène sur Unity dans laquelle nous téléporterons le joueur lorsqu'il devra combattre. Nous l'avons créée en creusant et texturant un terrain basique. Le combat à la première personne repose sur un vaste système d'animation et de collider.

Lorsque le joueur rentre l'input correspondant à l'attaque, l'épée qu'il tient s'anime d'un mouvement et son collider s'active. Lorsque l'épée rentre en contact avec un quelconque objet, son collider se désactive et on détermine si l'objet touché est un monstre. Si c'est le cas, on lui applique les dégats de l'épée.

Les dégats, comme dans tout RPG qui se respecte, sont le fruit d'une équation complexe. Selon si le joueur utilise une épée ou une masse, par exemple, les dégats appliqués sur le monstre seront soit des dégats dits tranchants, plus élevés par nature mais moins efficace contre les armures, ou des dégats d'écrasement à forte pénétration. A noter que les monstres appliquent également au joueur ces dégats spécifiques. Ainsi chaque monstre devra être appréhendé différemment pour peu que le joueur décide de passer plus d'un niveau de jeu..

Nous avons ajouté un objet permettant au joueur de tirer des boules de feu qui explosent à l'impact et font des dégats élémentaires de zone. En fait une boule de feu possèdent deux collider, un petit, pour la boule , actif dès le début et un autre plus gros pour l'explosion qui ne s'active brièvement que lorsque le premier touche quelque chose. Pour l'heure, les boules de feu n'appliquent aucun knockback aux ennemis mais nous comptons bien rajouter cela au plus vite.

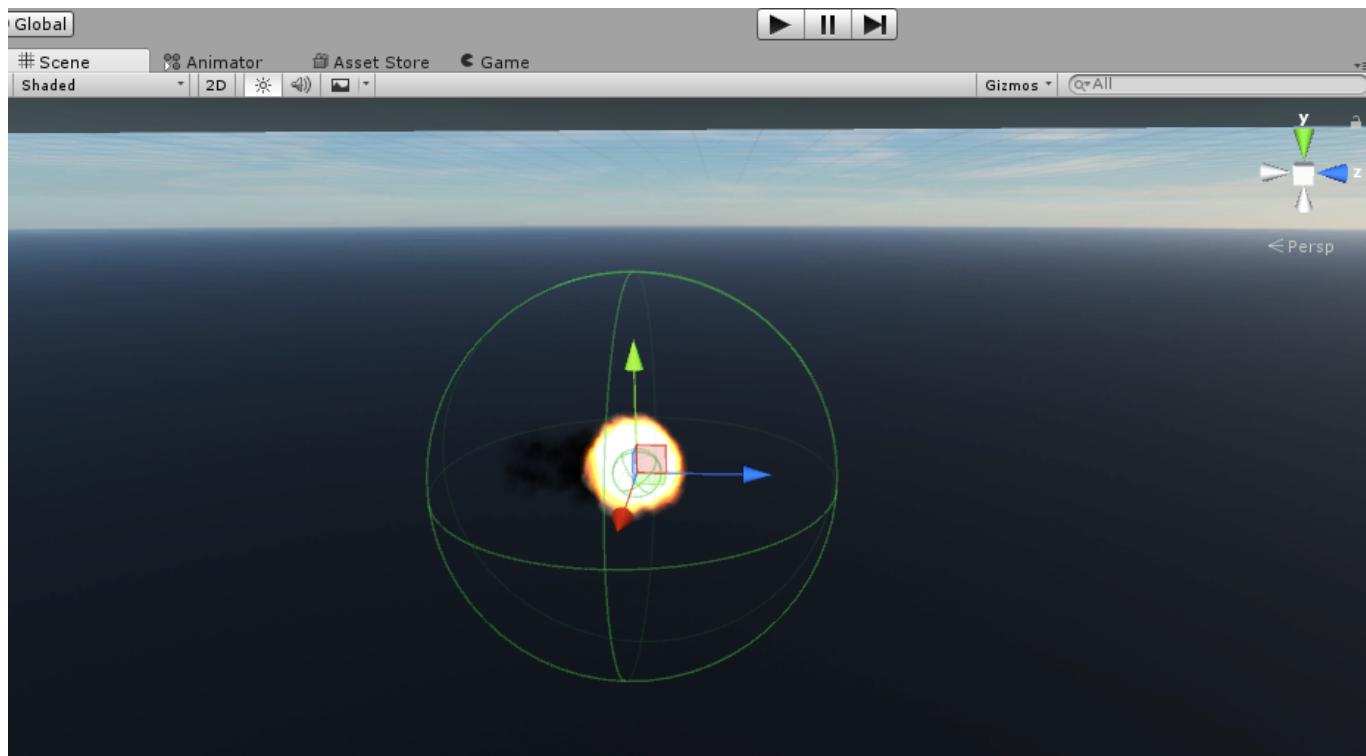
3.3 Les graphismes

Au cours de ces dernières semaines de travail nos efforts ont été entre autre portés sur les modèles 3D. Les monstres que nous utilisons ont été gratuitement importés de l'Asset Store Unity.

Pour créer la boule de feu nous avons mis deux systèmes de particules côté à côté, un pour les flammes et le second pour la fumée. Nous avons joué sur des paramètres tels que la vitesse et la rotation des particules pour rendre notre travail réaliste. De plus à l'impact, les deux premiers systèmes se désactivent et un troisième s'active pour faire une explosion de feu. Les matériaux des particules viennent d'un asset gratuit de unity.

3.4 Les animations

Les animations ont été l'un des domaines de travail les plus importants de ces dernières semaines. Elles ont en effet été indispensables à notre système de combat à la première personne, mais également à l'animation des monstres. Si ces derniers nous ont été livrés avec leurs animations déjà créées, il fut nécessaire de les coordonner habilement... Pour ce qui est des animations de combat, soit les animations de l'épée du joueur, elles ont été réalisées par nos soins.



l'IA

Chaque monstre se voit attribué différents scripts comportementaux, par exemple "AttackPlayer", "DashOnPlayer"... Originaiement, ces derniers disposaient d'une simple condition et s'exécutaient dès que celle-ci était complétée. Le problème de ce procédé était que l'on déboucha bien vite sur une cacophonie de mouvements désordonnés et n'ayant pas le temps de s'achever : la hache du gobelin n'avait pas le temps de retomber fendre le crâne de notre héros que déjà il décidait de se lancer dans un dash, interrompant sa frappe, et ainsi de suite..

Il fallut implémenter un script chef d'orchestre, qui pourrait à un intervalle de temps donné détecter les actions pouvant être effectuées, et choisir la plus avantageuse. Celle-ci serait alors effectuée, et le script chef d'orchestre attendra la fin de l'animation avant de chercher de nouvelles actions. Le comportement des monstres tend à être le suivant :

- s'avancer jusqu'à une certaine distance
- à partir de cette distance, faire des pas de cotés autour du joueur
- sauter sur le joueur !
- attaquer le joueur
- si le joueur attaque, tenter de bloquer le coup.

3.5 Le site web

Le site web ayant été plus avancé que prévu à la première soutenance, nous l'avons laissé tel quel et nous sommes focalisés sur d'autres aspects du jeu.

3.6 L'interface

Pour ce qui est de l'interface RPG, nous avons implémenté le gain de niveau au jeu. Classique des classiques des RPG, il permet au joueur d'attribuer un point de compétence entre différentes caractéristiques (force, agilité, intelligence, chance, charisme) ainsi que de choisir une compétence passive aléatoire parmi 2 proposées. Ces dernières auront souvent des effets étonnantes, encourageant le joueur à adapter sa façon de jouer pour en tirer le profit maximal.

Nous avons implémenté des ressources : Vie, Nourriture, Karma et Or. La vie vous sépare évidemment de la mort et est de fait la ressource la plus précieuse. Elle détermine en effet la vie maximum que vous aurez en lançant un combat. La nourriture, ensuite, est utilisée par chaque déplacement sur la carte. Si elle vient à manquer, vous perdrez de la vie à chaque déplacement.

Le Karma, nourriture spirituelle, évoluera en fonction de la morale dont vous faites preuve. Poignarder des marchands dans le dos le fera baisser. Sauver un village des brigands sans demander compensation l'augmentera. Votre alignement pourra vous apporter différents bonus. Karma élevé comme Karma faible ont leurs avantages et leurs inconvénients. L'or, nourriture des marchands, vous permettra d'acheter des priviléges, droits de passage, ou simplement d'aller faire vos courses lorsque vous arriverez en ville.



Coté combat, nous avons implémenté 3 jauge : vie, endurance, mana. L'endurance permet d'effectuer des dashes ainsi que de bloquer des coups. La mana quand à elle permet d'utiliser les armes main gauche au combat, recelant souvent de puissant sortilèges..

Une fiche de personnage à également été créée. Celle-ci comporte de nombreuses indications sur les caractéristiques du joueur, ainsi que son nom, son niveau, l'expérience nécessaire pour passer au niveau suivant, etc.. Sur cette fiche apparaissent en évidence la Force, l'Agilité, l'Intelligence, le Charisme et la Chance.

La Force augmente la vie du joueur, ainsi que ses dégats d'écrasement au combat. Elle peut également avoir un impact sur l'issue d'un dialogue si le joueur tente d'intimider d'autres personnages ou de faire des actes héroïques.

L'Agilité augmente l'endurance du joueur, ainsi que ses dégats tranchants au combat. Elle peut également avoir un impact sur l'issue d'un dialogue si le joueur tente de faire preuve d'adresse ou de furtivité.

L'Intelligence augmente les dégats magiques et élémentaires au combat. Elle peut également avoir un impact sur l'issue du dialogue si le joueur se décide pour les actions les plus sages.

Le Charisme n'a aucun impact sur le système de combat mais augmente la quantité d'expérience gagnée en toute circonstance, ainsi que la réussite dans tous les dialogue.

La Chance, quand à elle, porte bien son nom, et augmentera l'éclat des trésors comme des bourses des gobelins que vous pourfendrez.

Afin que le joueur ne soit perdu dans ces attributs barbares, nous avons implémenté une fenêtre d'aide s'ouvrant lorsque le joueur glisse le curseur de sa souris sur un élément. Ainsi, pour en apprendre plus sur l'agilité il suffira au joueur de glisser le curseur sur son indicateur, dans la fiche de personnage.

3.7 Le plateau

Le plateau de jeu est représenté par la table sur laquelle vous jouez et où sont disposés les cartes sur lesquelles vous vous déplacer. Chaque fois que vous cliquez sur une carte plusieurs choix vont s'offrir à vous. Selon le nombre de choix, les boutons de choix seront plus ou moins gros et ce grâce à un script ce qui permet de rendre le jeu plus agréable visuellement.

3.8 Les sons

Comme dit dans le précédent rapport de soutenance, nous voulions des musiques immersives et dans le ton du jeu, plutôt médiévaux donc. Ainsi avons nous ajouté différentes musiques (libres de droit bien sur) dans le jeu. Par exemple, le menu principal possède sa propre musique et l'arène de combat également. Le plateau de jeu possède plus de musiques, car c'est la scène sur laquelle le joueur passera le plus de temps. Pour que l'expérience sonore ne soit pas répétitive, nous avons créé un script permettant de jouer une playlist de musiques de manière aléatoire.

En d'autres termes, dès qu'une musique sera terminée ou que le joueur retournera sur le plateau depuis l'arène, une musique aléatoire parmi une liste sera lancée. Un autre script se nommant "switchcamera" permet de changer la musique selon la caméra sur laquelle on se positionne. Le script désactive les listener des autres caméras pour qu'une seule musique soit écoutée en même temps. Bien entendu une musique adéquate est utilisée dans l'arène, c'est à dire une musique plus épique, de combat, alors que le plateau de jeu à une ambiance médiévale. Les musiques que nous utilisons sont des compositions de Kevin MacLeod.



4 Prochaines étapes

Notre premier objectif pour la soutenance finale est de présenter un jeu et non un projet. Nous avons donc à cœur le soucis de la consistance et de la cohérence du jeu. Le contenu devra se parcourir d'un bout à l'autre sans que l'usage de cheats ou de passe-droits soit nécessaires.

La richesse de tout Rogue-Like étant le reflet de la richesse de son contenu, nous comptons également multiplier les monstres, objets et cartes afin que chaque partie soit une expérience nouvelle. Cette troisième session de travail ne sera pas tant celle de l'implémentation de nouveaux concepts, que celle de la complétion de ceux déjà implémentés.

Le système de combat : Dans cette démarche de complétion du jeu, l'objectif est de multiplier les objets :

- les armes, que le joueur tient de sa main principale et avec lesquelles il attaque

- les armes de main gauche, artefacts possédant de puissant pouvoir moyen-nant un coût en mana

- les talismans, amulettes et colifichets conférant aux joueur des bonus passifs au combat

De même, nous souhaitons ajouter au moins 2 autres ennemis de base, et 3 autres boss. Ces derniers, contrairement à l'araignée géante que nous avons déjà aujourd'hui et qui office de "boss de base", un monstre plus gros et méchant que les autres, nous voulons que les autres boss et des patterns spécifiques et des pouvoirs afin de relever le défi pour le joueur (et pour nous autres à la programmation)

Dans la même optique, nous essaierons de rajouter au moins 2 nouvelles arènes.

Le donjon : devra être plus élaboré, avec suffisamment de cartes pour ne pas tomber sur les mêmes à chaque étage. Pour l'heure, une dizaine de cartes se trouvent dans le jeu. Une trentaine serait très à propos.

Un leaderboard local est prévu, c'est à dire qu'il faudra trouver une manière pour évaluer une partie et lui attribuer un certain nombre de points. Les meilleurs scores seront conservés. La difficulté sera de ne pas effacer toute les valeurs une fois le programme coupé.

Le multijoueur : n'est pas encore au niveau du mode solo. Certaines fonctionnalités sont encore à implémenter ce qui améliorera l'expérience de combat comme les barres de vie ou d'énergie par exemple. Le mode est pour l'heure fonctionnel. Notre objectif est de le rendre plus intéressant.

L'aspect RPG : Pour la prochaine soutenance, la création du personnage devra être implémentée : à la manière d'un jeu de rôle papier, une part de hasard décidera de la manière dont débutera votre aventure. Le joueur devra pouvoir rentrer son nom, choisir une arme de base parmi 3 ainsi que déterminer ses attributs initiaux par le lancer de dé. Il pourra néanmoins relancer les dés un certain nombre de fois jusqu'à avoir une répartition lui convenant.

Installateur : Conformément aux exigences du projet, nous développerons un installateur de jeu pour la prochaine soutenance.

5 Annexe : aperçus du jeu



6 Conclusion

Au cours des dernières semaines, Mulligan est passé de simulateur de ran-donnée en arène à jeu de rôle teinté de combats. Nous sommes donc très satisfait de l'évolution du jeu et toujours motivés pour donner de notre mieux jusqu'à la dernière