

Rapport de Projet

Réalisé lors du S2 de la promo-2021

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

14 juin 2017

Table des matières

I	Introduction	4
1	Remerciements	4
2	Avant-propos	4
II	Retour sur le Cahier des Charges	4
3	Principe de Mulligan	5
3.1	Les membres du groupe	5
3.1.1	Jérémie "Cactus" BOURDET	5
3.1.2	Victor "Xarang" COATALEM	6
3.1.3	Florian "Dope" RAKOTOARIVONY	6
3.1.4	Hadrien "Hawk" NAVARRO	7

3.2	Introduction au genre Rogue-Like	7
3.3	Introduction au jeu Mulligan	8
4	Organisation du Projet	9
4.1	Objectifs	9
4.2	Répartition des taches	10
4.3	Logiciels utilisés	11
4.4	Le Site Web	12
5	Technique	15
5.1	Comportement des monstres	15
5.2	Le multijoueur	17
III	Développement du Projet	19
6	Historique	19
6.1	Etat du projet à la soutenance 1 :	22
6.2	Etat du projet à la soutenance 2 :	24
6.3	Etat du projet à la soutenance finale :	25
7	Difficultés	26
7.1	Gestion du temps	26
7.2	Le Game Design du Multijoueur	27
8	Ressources	28
8.1	Illustrations	29
8.2	Modèles 3d	30

8.3	Musiques	31
9	Récit de la réalisation :	32
9.1	Jérémie "Cactus" BOURDET	32
9.2	Victor "Xarang" COATALEM	32
9.3	Florian "Dope" RAKOTOARIVONY	33
9.4	Hadrien "Hawk" NAVARRO	34
10	Conclusion	35
11	Annexes	37

Première partie

Introduction

1 Remerciements

Nous souhaitons avant toute chose remercier tous les artistes ayant pris part à l'aventure Mulligan, sans lesquels le projet n'aurait pas été réalisable.

2 Avant-propos

Cet ultime rapport se construit autour de plusieurs objectifs : le premier, de vous introduire au jeu au cas où vous n'auriez pas eu accès aux rapports précédents. Le second, de vous partager au mieux notre expérience du projet : nos joies mais également nos déceptions. Le dernier mais pas le moindre, dresser l'état du projet au dernier jour : constatation des objectifs remplis, et de ceux qui à l'heure où nous écrivons ce rapport sont encore à pourvoir.

Plus que dans les autres rapports, nous voulons ici vous faire partager l'expérience que fut pour nous tous ce projet de S2, car nous sommes bien conscient que, au delà de la programmation, il s'agissait surtout pour ce projet de nous amener à nous responsabiliser et de nous familiariser avec les méthodes de communication du monde professionnel.

Sans plus nous étendre en généralités, toutes l'équipe des Paltoquets Ecornifleurs vous souhaite une très bonne lecture.

Deuxième partie

Retour sur le Cahier des Charges

3 Principe de Mulligan

3.1 Les membres du groupe

Lors de la réalisation du cahier des charges de notre projet Mulligan, nous nous étions présentés et nous avions donné nos ressentis à propos du projet à cette époque. Il est bien de faire un rapide retour dessus.

3.1.1 Jérémie "Cactus" BOURDET

Je m'appelle Jérémie, j'ai 17 ans et je viens d'une terminale S-SVT option ISN où j'ai découvert l'informatique et la programmation. J'y ai fait un site web pour un commerçant de ma ville et comme projet de fin d'année un Othello en python. Cette année a été décisive pour moi car c'est elle qui m'a donné goût à la programmation et qui m'a donc dirigé vers l'Epita.

Bon, je suis conscient qu'il y a une grosse différence entre ce que j'ai déjà fait et mon projet de S2 mais j'espère que mes expériences passées me permettront de combler certaines lacunes que j'ai ainsi que d'assumer mon rôle de chef de groupe. Je pense que ce jeu est l'occasion de confirmer mon goût pour la programmation en plus de m'apprendre à travailler vite et bien en groupe. Faire un jeu vidéo n'est pas vraiment un rêve de gosse pour moi mais je suis impatient de commencer et je suis sur qu'on va réaliser un projet solide dont je serai fier.

3.1.2 Victor "Xarang" COATALEM

Sortant de Terminale S spécialité Maths, il été tout naturel pour moi d'orienter mes études vers le domaine qui déjà alors occupait la plus grande partie de mon temps libre. Mordu du jeu vidéo honoraire, j'ai depuis longtemps appris l'algorithmie dans le but d'en faire un moi même un jour. Très tôt déjà, que ce soit en pianotant sur le clavier de ma calculatrice durant les cours d'histoire géographie un peu longs, et plus tard par l'apprentissage du C ainsi que du PHP. Je suis donc évidemment décidé à m'impliquer largement dans le projet, que ce soit en m'occupant de la programmation ou bien en apprenant à me servir de nouveaux logiciels au besoin.

3.1.3 Florian "Dope" RAKOTOARIVONY

Je n'ai pas toujours été intéressé par l'informatique et je me suis mis aux ordinateurs assez tardivement. Cependant, j'ai toujours été très appliqué et travailleur dans ce qui m'intéressait. L'informatique en faisant partie, je suis extrêmement motivé à l'idée de ce projet. Durant ma scolarité, j'ai souvent travaillé en groupe que ce soit pour des révisions, des exposés ou des projets. De ce fait, travailler en groupe pour ce projet ne me pose aucun problème et me rassure même. En effet, mes connaissances en informatique étant plutôt lacunaires, je compte sur les membres de mon groupe pour m'accompagner à travers cette aventure. De plus dans le futur, je devrai très probablement travailler en groupe, que ce soit à l'EPITA ou en entreprise, cette expérience d'un effort non-négligeable à fournir pendant une période plutôt conséquente me paraît être une bonne expérience dans ce sens. Comme je l'ai dit plus tôt, je n'ai pas beaucoup d'expérience en informatique et j'espère bien que la réalisation de ce jeu me permettra d'en apprendre plus sur la confection des jeux et sur moi-même (mes capacités, mes limites).

3.1.4 Hadrien "Hawk" NAVARRO

Pour ma part, le projet de S2 est quelque chose dont j'attends beaucoup. En effet, après mes années en première et terminale S, je n'avais toujours pas d'idée à propos des études que je comptais effectuer. Ayant un certain goût pour les mathématiques et la logique, j'ai alors décidé de faire une école d'ingénieur, en prépa intégrée si possible. Passant la majeure partie de mon temps sur mon ordinateur, j'ai alors voulu savoir comment il fonctionnait, et si j'étais capable de créer des choses moi-même. Je n'ai personnellement jamais vraiment programmé et c'est pour cela que j'attends beaucoup de ce projet, car j'espère qu'il me donnera des réponses à propos de mon goût pour la programmation et l'informatique en général. Ce projet est intéressant car il aborde de nombreux aspects ce que l'on peut nommer informatique. Il utilise en effet des logiciels de modélisation comme Unity ou Blender ainsi que des langages de programmation comme le C#. Il aborde également d'autres aspects, pour notre part la conception d'un jeu vidéo, autant du point de vue narratif que conceptuel. Cela me permettra j'espère, d'améliorer mes compétences en tant que possible futur ingénieur informatique ainsi que de me guider dans mes futurs choix et réalisations.

3.2 Introduction au genre Rogue-Like

Le genre « Rogue Like » désigne toute une série de titres sortis à la suite du jeu Rogue en 1980, et reprenant les bases de ce dernier comme concept fondateur : une longue descente aux enfers semée d'embûches dont le joueur ne pourra, de manière réaliste, pas triompher dès le premier essai, mais seulement au travers de multiples échecs ainsi que d'un apprentissage progressif des pièges et ennemis qui arpencent les lieux. Il est également commun aux Rogue Likes de générer les différents niveaux de manière procédurale et que ces derniers soient d'autant plus difficile que l'on approche le cœur du donjon.. Ces jeux ont tendance à avoir une difficulté relevée : la mort est monnaie courante, et lorsqu'elle survient, le personnage est immédiatement effacé. En découle un sentiment d'anxiété d'autant plus fort que les niveaux s'enchaînent, l'adrénaline monte alors que vous prenez part à un combat permanent pour votre survie, plus que pour votre victoire. De plus, les

différents étages du lieu que l'on explore renferment souvent de nombreux secrets, ce qui ne rend leur exploration que plus passionnante.. Le genre Rogue Like n'est prévu que pour être joué en solo, à l'exception de certains titres récents (ex : Risk of Rain). Mais cela n'empêche en rien les compétiteurs de s'acharner sur le tableau des scores et les rôlistes de raconter leurs dernières péripéties sur quelques forums de passionnés..

La popularité du genre remonte à l'époque des jeux à jouer depuis la console . Ils proposaient en effet un gameplay assez riche pour ce que le support pouvait permettre. Depuis, les nouvelles techniques ont permis de nombreuses déclinaisons du genre : jeux de plateforme, FPS, RPG, jeux de gestion.. Tous sont passés sous les bannières de Rogue. La scène indépendante s'est de plus largement emparée du concept, très populaire auprès des amateurs.

3.3 Introduction au jeu Mulligan

Mulligan est un RPG rogue-like 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.

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...

L'aventurier, représenté par un pion, se déplacera alors de cartes en cartes 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 pérégrinations, le joueur dépensera des ressources (ex : nourriture, or, moral..), en acquerra d'autres, dénichera de précieux trésors l'aidant 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.

4 Organisation du Projet

Très vite, nous avons pris à coeur la bonne organisation du travail ainsi que la répartition entre les différents membres du groupe.

4.1 Objectifs

Nous avons établi un emploi du temps strict mais raisonnable afin de pouvoir juger en toute circonstance de notre avance ou retard sur ce qui était prévu. Néanmoins certains des objectifs que nous nous étions fixés se sont révélés peu adaptés à la manière dont le jeu se développait. Les animations, par exemple, n'avaient de sens qu'une fois le système de combat et les monstres bien implémentés, à la deuxième soutenance donc. S'avancer en début de projet en disant que nous les aurions commencé pour la première soutenance, était une simple erreur de jugement dûe à notre faible (nulle) expérience en création de jeux vidéos. Ainsi ce planning s'est-il naturellement assoupli, mais en compensant en toute circonstance un retard sur un point par une avance au moins plus importante sur un autre, dans la mesure du raisonnable évidemment.

Tache/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	**	***	***

4.2 Répartition des taches

Afin de faciliter le travail de groupe et d'optimiser notre temps, nous avons logiquement réparti notre travail en différentes taches que nous nous sommes partagées. Considérant nos préférences et nos facilités sur certains domaines (ou non facilités), nous avons choisi qui devait faire quoi afin que le travail soit plus efficace. Tout cela bien sur en répartissant le plus équitablement possible le travail. Bien sur certaines taches sont plus complexes que d'autres et demandent plus de travail, mais nous avons fait de notre mieux pour que cela soit le plus juste.

Vous trouverez, ci-dessous, un tableau de répartition de ces tâches :

Tache/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

4.3 Logiciels utilisés

Git : Afin de faciliter la tache pour tout le monde, nous avons créé dès le début du projet un repository sur github afin que tous les membres du groupe puissent travailler simultanément et que leurs modifications soient bien prises en compte.

Nous avons néanmoins eu de nombreux problèmes avec ce logiciel, notamment lorsque les fichiers .unity (les scènes) étaient modifiées.. A de nombreuses reprises, ces derniers ont été corrompus et dans l'impossibilité complète d'être réouverts, nous faisant perdre beaucoup de temps à remettre le jeu en état en passant par

des versions plus anciennes.

Discord : Car la communication est indispensable, nous avons également créé un serveur Discord pour le groupe où nous pouvions facilement échanger sur le projet, les échéances, la répartition des tâches et ainsi de suite.

Overleaf : Pour écrire ce merveilleux rapport (ainsi que les précédents) nous avons utilisé Overleaf, éditeur LaTeX en ligne, toujours dans l'objectif que tous les membres du groupe puissent facilement travailler de leur côté en ayant des mises à jours en temps réel du travail des autres membres du groupe.

Visual Studio : Integrated Development Environment (IDE) pour le développement en C#. Il nous a permis d'écrire les scripts pour Unity.

Inno-Setup : Installeur gratuit pour programme windows qui nous a permis de créer l'installateur de notre jeu.

4.4 Le Site Web

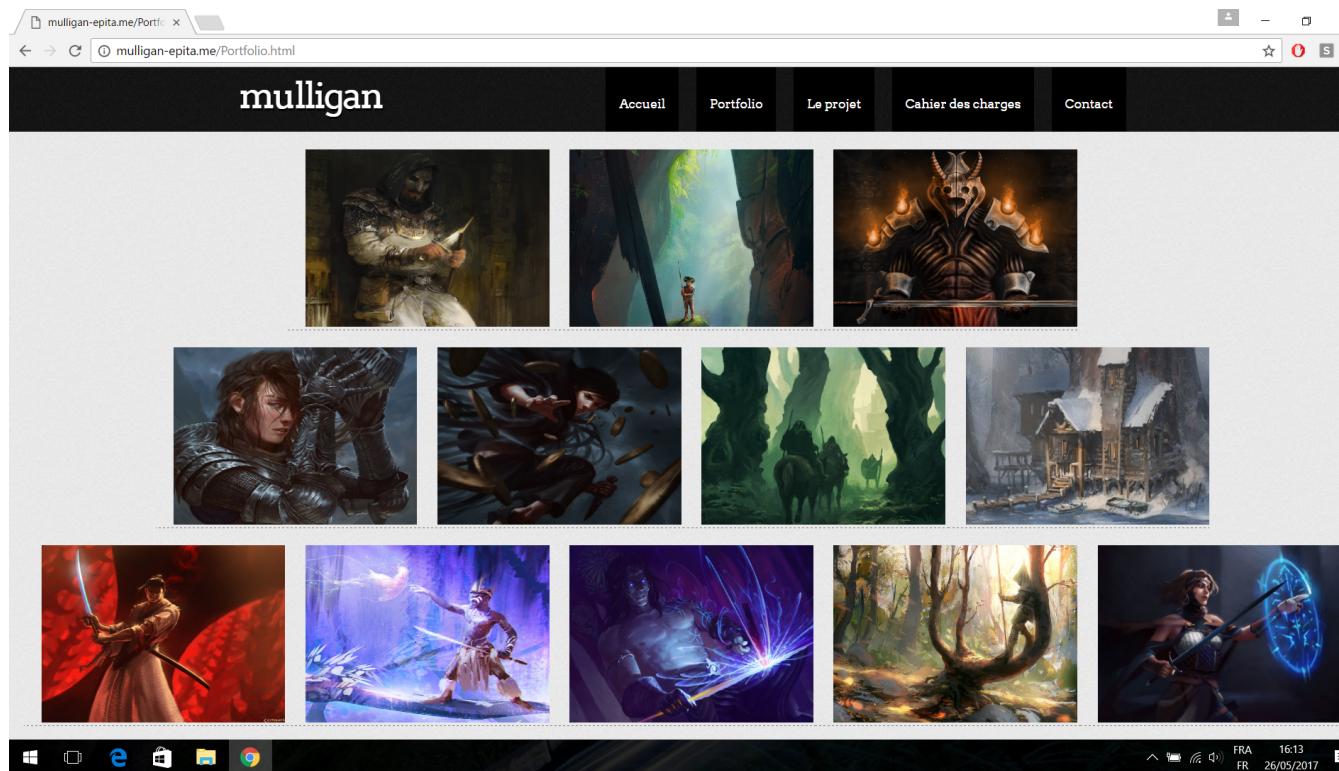
Dès le début du projet, nous avons créé un site web sur lequel nous publions nos dernières avancées. Nous avons pour cela hébergé le site sur un serveur git. Le site a été réalisé avec du HTML et CSS basique. Nous avons également un portfolio en ligne sur lequel nous postons les images pour lesquelles nous disposons des droits. Pour notre site web, nous sommes partis d'une template très austère afin de développer le site web à plus forte raison de nos propres moyens. L'architecture du site est la suivante :

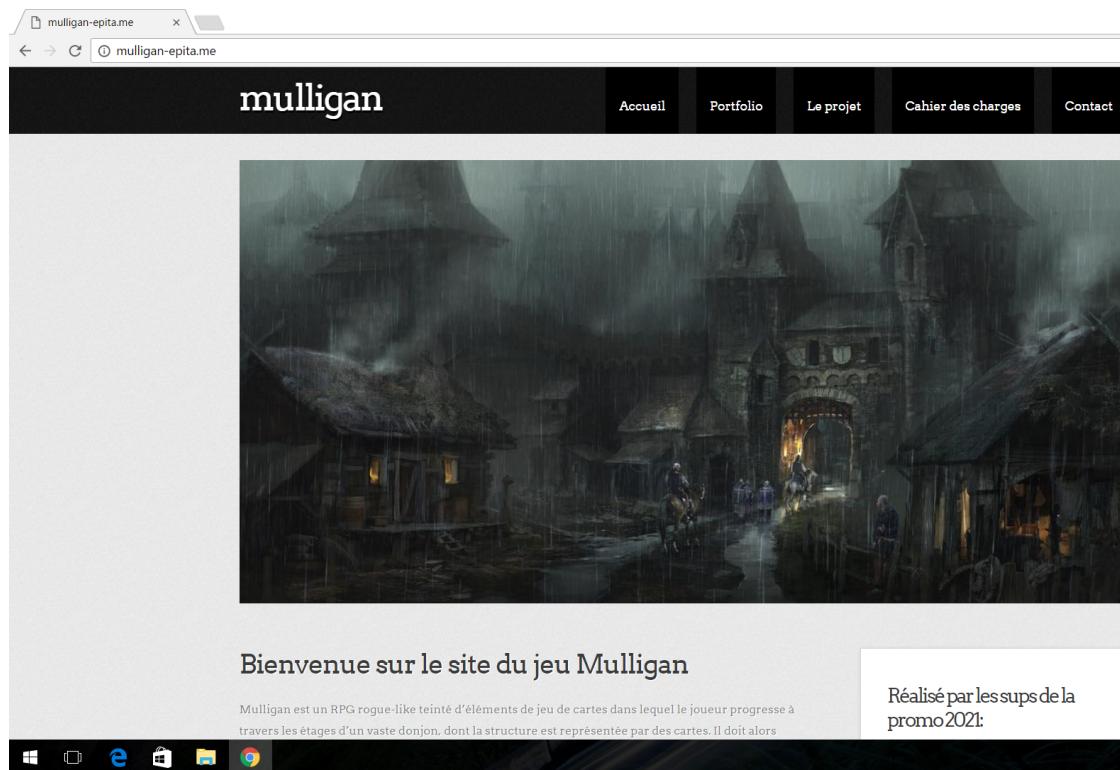
- Une page d'accueil qui explique rapidement le jeu.
- Un portfolio présentant les principales images utilisées pour faire nos cartes, ainsi que des screens shots du jeu à l'avenir. Au survol d'une image, elle apparaît en plus

grand pour plus de visibilité. Nous avons vraiment eu du mal à implémenter cette fonctionnalité de manière propre. Dans un premier temps, l'image apparaissait juste en plus grand, du coup les autres images se décalaien et c'était visuellement perturbant. Maintenant nous faisons apparaître une autre image, plus grande et surtout en absolu, c'est à dire qu'elle survole toutes les autres.

- Une page qui présente le jeu, avec le principe, le synopsis et l'histoire de l'état des Rogue Like.
- Un onglet contact permettant de nous envoyer des mails. Pour ce qui est de l'hébergement web, nous avons utilisé nos comptes Epita pour obtenir un nom de domaine gratuit. Le site est actuellement en ligne grâce à Github pages, un service permettant de host gratuitement des sites basiques.

Vous pouvez le consulter à cette adresse : <http://mulligan-epita.me>





5 Technique

Dans cette section seront évoqués certains détails techniques jugés suffisamment importants / intéressants pour figurer sur ce rapport. Lecteurs ne connaissant rien du code, n'ayez crainte, nous parlons ici de fonctionnements simples et imagés.

5.1 Comportement des monstres

Le comportement des monstres, vaste sujet ! Pathfinding, Aggressivité, Cohérence, Tactique.. Intelligence le terme est fort, aussi paraît-il primordial de méanger la chèvre et le chou afin que le comportement fasse sens d'une part, et puisse challenger le joueur d'une autre sans être injuste pour autant.

Tous les monstres possèdent un script "mobAnimation", que nous appellerons script chef d'orchestre. Tous les autres scripts attachés à ce monstre détermineront les actions que ce dernier pourra prendre à tout instant. Par exemple, un monstre doté d'un "mobAnimation", d'un "approachPlayer" ainsi que d'un "attackPlayer" pourra s'approcher du joueur et, s'il se trouve à une distance suffisante, l'attaquer.

A un intervalle de temps régulier (0.1 secondes) le script mobAnimation vérifie l'état du monstre. S'il est occupé (à attaquer par exemple) alors le script n'a aucun effet, et attend donc que le monstre finisse son action. Au contraire, si le monstre est inactif, alors le script mobAnimation parcourra tous les autres scripts attachés au monstre et vérifiera si leur condition est respectée : distance suffisante, temps de recharge écoulé par exemple. Dans le cas où cette condition est respectée, on ajoute l'action dont il était question à la liste des actions possibles.

Une fois ce parcours fini et la liste effectuée, un algorithme de quantification décide de la meilleure action à prendre en fonction de divers paramètres : vie du joueur, vie du monstre, "le joueur pare-t-il?", etc..

Par défaut, voici par ordre croissant les actions les plus intéressantes pour le monstre :

- s'approcher du joueur
- faire des pas de côté autour du joueur
- bloquer
- attaquer
- si le joueur attaque, bloquer

Evidemment, toute prise de décision aura également une part d'aléatoire afin de surprendre le joueur. La décision prise, l'action est effectuée et l'algorithme recommence.

A l'origine, nous n'utilisions pas de script "chef d'orchestre", ce qui nous a causé énormément de tort lorsqu'il s'agissait d'attendre qu'une action se termine avant de lancer les autres. Un monstre possédant "attackPlayer" et "dashOnPlayer", par exemple, passait son temps à commencer une attaque et à l'interrompre pour courir sur le joueur. Peu efficace, donc. Nous possédons plusieurs types de monstres, de l'araignée géante en passant par une horde de squelettes pour arriver aux gobelins, le type d'ennemi le plus fréquent du jeu.

5.2 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 actionss 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 régulierement récupérer les déplacements des autres joueurs, corriger leurs po-

sitions 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.

Troisième partie

Développement du Projet

6 Historique

Mulligan, c'est un projet mené depuis le mois d'Octobre par deux membres du groupe, et portant depuis son origine de hautes ambitions. Le jeu se veut en effet à la croisée des genres, avec des éléments de gameplay provenant des univers des RPGs, Rogue-Likes mais également jeux de combats en vue à la première personne et jeux de cartes..

Dans le lexique des joueurs de cartes, d'ailleurs, "Mulligan" renvoie à une phase de jeu, souvent située temporellement au début de la partie, où le joueur est libre de se défausser de certaines cartes de sa main et d'en repiocher dans une certaine mesure. Un contre à la malchance, en somme. Ce nom pour le projet fut choisi pour diverses raisons ; le fait que nous utilisions des cartes, évidemment, mais également la consonnance qui nous était agréable (il n'en faut pas plus..). Un regret que l'on pourrait avoir est le fait que, au fur et à mesure du développement, l'aspect "jeu de cartes" se soit estompé aux profits d'éléments plus propres aux jeux de rôles traditionnels. En ce sens, Mulligan fait surtout allusion au synopsis derrière le jeu : la partie de cartes entre l'aventurier solitaire et le mysterieux narrateur..

Une fois le groupe au complet et déterminé à mener le projet à bout, un vaste spectre d'objectifs s'est dévoilé devant nous. En effet, si le game design avait déjà été en grande partie déterminé et théorisé, le défi de son implémentation demeurait complet.

Dès le début du projet, nous avons établi une liste précise et ordonnée des features que nous voulions ajouter au jeu. Nous avons ainsi démarqué l'indispensable du superflux, et ce non sans sacrifice..

Avant de se consacrer pleinement au développement du projet, nous avons voulu nous assurer que des artistes accepteraient de nous laisser utiliser leurs illustrations pour nos cartes. En effet, sans ces dernières, le jeu perd énormément de son âme - et de son intérêt par la même occasion - Nous avons donc cherché sur le site ArtStation des artistes en Freelance, maîtres de leurs travaux et disposés à nous laisser les utiliser. Les premières réponses positives ont créé une véritable euphorie ; Des artistes ayant travaillé sur des jeux AAA tels que The Witcher 3 ou Call of Juarez : Gunslingers nous répondaient avec enthousiasme, nous demandant même de leur envoyer le projet quand il serait achevé.. Il s'agissait dès lors que l'on soit à la hauteur !

Cela nous a ammené à la question délicate de la langue. Au cours du projet, nous avons souvent hésité entre anglais et français. L'anglais était le choix de l'universel, nous permettant de partager le jeu en ligne au tout venant. Le français était le choix plus littéraire, Mulligan étant en effet un jeu à forte raison textuel, la qualité de l'écriture n'est pas à négliger. Si écrire en anglais ne nous pose pas de problème concret, c'est tout de même une tâche plus longue et fastidieuse que d'écrire dans notre langue maternelle. De plus, nous sommes plus à même d'écrire des textes bien construits et au lexique riche en français. Cela nous a amené à choisir la seconde option, au grand dam des non-francophones.. Une traduction intégrale du jeu est envisageable, mais semble plutôt pénible à mettre en place et peu intéressante pour ce qui est de la programmation.

Pour l'heure, cette proposition est donc écartée.

La première phase de développement aboutit à la création de l'architecture du jeu ; Toute partie de Mulligan se présente comme une alternance entre deux phases : une phase "stratégique", s'apparentant à un jeu de plateau dans lequel le joueur est un pion, se déplaçant de cartes en cartes, les révélant sur son passage. Les cartes, révélées, narrent alors des événements aux joueurs, et demandent à ce dernier de prendre une décision parmi 4, influant la fin du récit. Certaines cartes amèneront le joueur à affronter des ennemis, passant alors dans la deuxième phase de jeu : la phase "de combat". Cette alternance entre les deux phases, à la base même du jeu, fut naturellement implémentée dès le début du projet. Malheureusement, la conséquence fut que lors de notre première soutenance, notre travail faisait plus office de grand menu que de jeu à proprement parler. Nous avions en effet la structure mais pas le contenu ; Les cartes étaient mises en place mais les actions du joueur sur ces dernières n'avaient aucunes conséquences ; le joueur pouvait se déplacer dans une arène mais en aucun cas n'avait d'armes ni de monstres à affronter. Cette première soutenance fut donc plutôt démoralisante, car ne mettant pas bien en avant tout le travail que nous avions accompli..

La deuxième phase de développement vit la naissance du gameplay du jeu tel que nous l'avions imaginé. En implémentant les fonctionnalités élémentaires, telles que le système de combat, la gestion d'inventaire ainsi que la fiche de personnage, Mulligan a commencé à prendre son sens.

6.1 Etat du projet à la soutenance 1 :

Le donjon etait crée procéduralement, de sorte que chaque carte soit adjacente à une autre. La carte contenant le boss du niveau etait placée en dernier. Toutes les cartes etaient placées face cachée sur la table ; ce n'est que lorsque le joueur arrive dessus qu'elle se révèlaiennt. Néanmoins, à chaque début d'étage, une méthode remplaçait aléatoirement certaines cartes face cachée par des cartes révelées, qui indiquaient la nature de la carte qu'elles remplacaient (péripetie, trésor, rencontre) sans en donner l'identité exacte. Le nombre de cartes de chaque catégorie etait fixée lors de la création du niveau, de sorte que chaque niveau contienne au moins un trésor, une rencontre, et une grande quantité de péripéties.

Le joueur disposait d'un inventaire, dans lequel il pouvait stocker les trouvailles qu'il fait au cours de son aventure. Ce dernier etait représenté par un tableau d'`InventaireSlots`, une classe créée par nos soins pour contenir entre autres l'id de l'objet présent dans cette case de l'inventaire, ainsi que sa nature : Le joueur disposait d'un `InventaireSlot` dédié à chaque partie d'armure qu'il pouvait porter : Casque, Armure, Bottes, mais également Arme main principale et Arme main secondaire. Nous ne voulions pas que nos preux chevaliers puissent équiper leurs épées dans leurs bottes ou inversement. Lorsque le joueur cliquait sur un `InventaireSlot` et que son id n'était pas nul (`InventaireSlot` non-vide), il gardait en mémoire l'id de l'`InventaireSlot` sur lequel il viennait de cliquer et associait 0 à l'id de cet `InventaireSlot`. La prochaine fois qu'il cliquait sur un `InventaireSlot` vide, il associait à cet `InventaireSlot` l'id qu'il avait en mémoire avec la condition que la nature corresponde : épée dans la main, casque sur la tête, pas l'inverse. Il pouvait alors de nouveau cliquer sur un `InventaireSlot` plein pour en prendre le contenu, et ainsi

de suite.

-Le système de combat : A ce moment, nous avions implémenté par nous-même un module de déplacement en vue à la première personne dans un monde en 3 dimensions. Ce module était 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 pouvait effectuer un dash, c'est à dire se déplacer rapidement dans une direction, ce qui améliorerait les combats. Nous avions également 3 barres, respectivement la vie, la nourriture, et le moral. En fait ce sont des rectangle colorés dont la scale en x dépend d'une variable. Quand la nourriture diminuait, le rectangle devenait plus petit et se décalait vers la gauche pour rester toujours à la même origine. Nous avions également créé une arène sur Unity dans laquelle nous téléportions le joueur lorsqu'il devait combattre. Nous l'avions créée en creusant et texturant un terrain basique.

-Les graphismes : Pour notre arène, nous avions utilisé des assets gratuits de l'Assets Store Unity(nature starter kit 2), du plus bel effet. Pour ce qui est des illustrations des cartes, nous avions désormais l'accord de nombreux artistes pour utiliser leurs travaux. De l'artiste sous contrat au freelancer en passant par l'amateur talentueux et le professeur, tous ont montré un grand enthousiasme pour le projet.

Nous possédions un menu principal avec deux boutons, un bouton pour lancer le jeu et un pour le quitter.

Nous possédions également un menu pause à l'intérieur du jeu qui met le jeu en pause lorsque l'on appuie sur la touche echap.

Ce menu de pause possède un bouton pour enlever la pause et un pour relancer

le jeu, un bouton pour revenir au menu principal et un bouton pour quitter le jeu. Nous aurons plus tard également un bouton d'options afin de régler différentes choses, telles que le son principal ou la résolution de l'écran.

Pour ce qui est du code des menus, nous avons créé un canvas en 2D avec plusieurs boutons, ainsi qu'un game object affilié aux boutons qui possède un script permettant de lier des fonctions à nos boutons.

Exemple :

Pour le bouton start the game, nous avons utilisé Application.LoadLevel(). Pour le bouton Quit, Application.Quit() tout simplement.

6.2 Etat du projet à la soutenance 2 :

Lors de notre deuxième soutenance, nous avons commencés à finaliser tout l'aspect fonctionnel du jeu, c'est à dire tout les scripts et tout ce qui est de l'ordre d'unity et non pas du jeu lui même.

Tout ce que nous avons fait pour cette soutenance est résumé dans les parties comportement des monstres et multijoueur car c'est ce dont nous nous sommes principalement occupés.

Le jeu était alors presque fini, ne restait plus qu'à rajouter un maximum de contenu et quelques nouvelles fonctionnalités sympathiques.

6.3 Etat du projet à la soutenance finale :

Pour la dernières soutenance comme une grosse partie du fonctionnel était déjà fait, par exemple les scipts de combat et d'animations était déjà fait, nous avons pu nous concentrer beaucoup sur l'ajout de contenus c'est à dire plus de cartes pour améliorer le jeu.

-L'installateur :

Mulligan est un projet et non un simple TP, il nous fallait un installateur pour donner de la crédibilité à notre jeu.

Afin de construire notre installateur, nous avons fait le choix d'utiliser un logiciel nommé "Inno-Setup". Ce logiciel, est un logiciel gratuit mais qui permet une certaine efficacité. Inno Setup permet de créer un logiciel d'installation pour les programmes Windows, ce qui est donc parfait pour notre projet. Ce logiciel permet une installation rapide, car il regroupe les dossiers en rapport avec notre jeu.

La génération de scripts modifiables nous a fait profiter pleinement de ce logiciel en utilisant nos connaissances personnelles afin de personnaliser notre installateur. Cette personalisation démarque notre jeu en offrant à l'utilisateur une autre approche de l'installation.

Enfin, lors de l'installation, le programme place les fichiers du moteur de mise à jour et du jeu sous forme de paquets dans un dossier dédié. Il modifie aussi les entrées registres du système d'exploitation Windows, afin d'informer le système d'une nouvelle installation. Ainsi, de nombreux raccourcis ont pu être créé automatiquement.

7 Difficultés

7.1 Gestion du temps

En janvier déjà, une bonne partie du travail avait déjà été ébauchée depuis le mois d'octobre. Nous avions donc un peu d'avance sur le projet, mais avons tout de même décidé de capitaliser cette avance en ne relâchant pas le rythme, afin que le travail soit bien réparti sur le semestre. Les deux premières soutenances, néanmoins, ont été difficiles à assumer. Passant le premier jour ces deux fois, nous n'avions que peu de temps -un week end- pour se consacrer pleinement au projet lors des fameuses journées (nuits) de rush. Et, bien qu'au lendemain de la soutenance, nous avions encore une semaine entière sans cours, il était plus difficile de trouver la motivation pour continuer à exploiter cette semaine de soutenance. Nous avons donc perdu un peu de notre avance. En réponse à ses délais cours, nous ne nous sommes toutefois pas laissés abattre et avons rentabilisé chaque minute afin d'être prêt pour la soutenance. Néanmoins, il demeurait que nous finissions le développement le matin même de la soutenance, et n'avions en conséquent que peu de temps pour préparer notre présentation en soi, que ce soit par l'élaboration d'un Prezi ou simplement en révisant nos textes. Heureusement, notre dernière soutenance nous laisse beaucoup de temps à la suite des partiels -une semaine et demi- afin que notre projet puisse atteindre la hauteur de nos espérances.

7.2 Le Game Design du Multijoueur

Historiquement, Rogue-Like et multijoueur ne font pas bon ménage. Si certains tentent tout de même d'introduire la possibilité à plusieurs joueur de jouer en coopération -souvent sur la même machine- l'expérience est entièrement différente et généralement moins poignante. L'implémentation d'un multijoueur pour ce projet n'étant pas une option, nous avons tout de même réfléchi à différents moyens de rendre un tel mode de jeu intéressant.

Dans nos brainstormings, deux courants de pensée majeurs se sont démarqués : le premier, qui prône le fait de rester attaché à l'idée du Rogue Like et de simplement le porter en multijoueur, coopération ou versus. Le second, au contraire, consistait à mettre de côté l'aspect Rogue-Like du jeu et de faire du multijoueur un jeu de combat au corps à corps, plus dynamique.

Parmis les idées du premier courant : - un mode coopératif où tous les joueurs se déplacent ensemble et chaque décision est soumise à un vote ; simple, mais inintéressante - un mode versus au tour par tour où les joueurs explorent tous le même donjon et se battent pour en obtenir les trésors ; Necéssite de réécrire une quantité importante de script, pour un intérêt discutable.

Enfin, une dernière idée que nous avions vite écartée en début d'année consistait en un mode versus dans lequel un des deux joueurs jouerait le maître du jeu, disposant les cartes selon ses envies et tendant des pièges sournois au joueur. Si c'est de loin l'idée la plus intéressante et offrant le plus vaste horizon en terme de game design, c'est également la plus difficile à transcrire dans le code. Nous l'avions tout de même évoqué dans le cahier des charges, comme un lointain objectif qui pourrait éventuellement voir le jour mais pour lequel nous ne nous avancerions

pas.

A la fin de la deuxième soutenance, notre examinateur nous a implicitement dis que, pour obtenir la note maximale, il nous faudrait réaliser cette idée que nous avions émise sans trop y croire au mois de Janvier. Un défi de taille, car nécessitant la réécriture de quasiment tous les scripts écrits jusqu'alors. Un challenge qui nécessite la réorganisation de tout notre emploi du temps et objectifs pour la troisième soutenance.

8 Ressources

Le développement d'un jeu vidéo n'étant pas qu'une histoire de programmation, il nous semble sage, pour cette fin de projet, de citer tous les éléments sans lesquels Mulligan ne serait guère plus que l'aventure d'un tictac dans un silencieux monde de cubes.

8.1 Illustrations

Indispensables à tout jeu de cartes, les illustrations se devaient d'être soignées et cohérentes les unes avec les autres afin d'éviter que l'on se retrouve avec un mélange de styles graphiques peu propice et brisant de ce fait l'immersion ainsi que l'univers du jeu. Nous avons donc fait des recherches approfondies d'artistes dont le style correspondait à notre vision du jeu sur ArtStation. Dans un soucis de professionalisme, nous n'avons pas volé leur travail comme de fieffés charognards mais leur avons bien demandé l'autorisation par mail en précisant que le projet n'avait aucun but lucratif. Si certains artistes n'avaient aucun problème à nous laisser utiliser les illustrations dont ils détenaient les droits, d'autres poussaient la démarche un peu trop loin à notre goût, nous poussant à poliment leur répondre que nous nous contenterions des artistes plus disposés à coopérer.

Ci-dessous la liste des artistes ayant participé au projet :

- Andrezj Dybowsky
- Manuel Castanon
- Andreas Rocha
- Dominik Mayer
- Josh Bentley

8.2 Modèles 3d

Les modèles 3d utilisés dans le jeu ont été tous sans exception obtenus gratuitement depuis l'Asset Store Unity. Ces derniers étaient également munis d'animations que nous avons de même utilisées.

Ci-dessous la liste des assets utilisés :

- FreeWeapons
- RPG Swords
- Fantasy RPG Icon Set - Free
- Fantasy Spider
- Goblin
- DungeonCharacters Skeleton - Free
- NatureStarterKit2
- PyroParticles
- Wispy Sky

8.3 Musiques

Les musiques du jeu se veulent à la croisée de la musique d'ambiance de taverne médiévale et de thèmes plus lourds, pesants, faisant monter la tension du jeu au fur et à mesure des choix qui se présenteront au joueur. Nous avons pour cela utilisés des compositions du compositeur Kevin McLeod, réputé sur Internet pour ses musiques libres de droit.

Ci-dessous la liste des musiques utilisées :

Kevin McLeod :

- Volatile Reaction
- Mystery Bazaar
- Black Vortex

9 Récit de la réalisation :

9.1 Jérémie "Cactus" BOURDET

Faire un grand projet, en groupe , en parallèle de TP de programmation qu'il fallait terminer en une semaine m'a permis de diversifier mes expériences. Il y a une nette différence entre le travail hebdomadaire et ce genre de projet. Même le projet que j'ai fait en ISN en terminale l'année dernière n'avait rien de comparable avec la réalisation de Mulligan. J'ai appris énormément cette année grâce a Mulligan et je pense que cela me sera très utiles pour plus tard. Par exemple, en tant que chef de groupe,j'ai appris à gérer un groupe, organiser une répartition des charges et un planning.

Bien évidemment j'ai découvert et appris à maîtriser des outils de travail comme unity, blender et overleaf. Personnellement, j'ai beaucoup aimé le travail en groupe, surtout lors dans rush quand nous étions tous les 4 dans le même salle en train de tout donner pour finir dans les temps. Même si nous n'avons pas pu rendre Mulligan tels que nous l'avions prévu, toute expérience est bonne à prendre et le bilan est positif.

9.2 Victor "Xarang" COATALEM

Dès les prémisses du projet en octobre dernier je me suis beaucoup investi dans le projet, notamment pour ce qui est de prendre Unity en main. C'est alors que j'ai pris un peu d'avance en commençant le projet "tout seul", ce que je regrette un peu aujourd'hui car je pense qu'il aurait été plus intéressant pour tout le monde que l'on commence ensemble. La conséquence fut que très vite, les autres membres

du groupe ne s'y retrouvaient pas dans mes scripts -parfois un peu tirés par les cheveux-, m'obligeant à m'occuper d'une grande partie du développement. Je pense néanmoins avoir beaucoup appris, et aimerais beaucoup prendre le temps à l'avenir, pourquoi pas l'été prochain, de développer un autre jeu. Je me suis rendu compte de l'importance de ne pas se jeter dans les lignes de code sans savoir où l'on va, et prendrai en conséquent le temps de bien théoriser le code de bout en bout. Pour ce qui est de l'aspect "professionalisant" de ce projet, je pense que nous avons fait du bon travail étant donné des échéances qui nous étaient imposées. Je suis notamment fier d'avoir obtenu l'accord des artistes qui nous intéressait, plutôt que d'avoir volé leur travail sans vergogne -impensable dans le monde du travail-. L'un dans l'autre, ce projet fut vraiment une excellente expérience.

9.3 Florian "Dope" RAKOTOARIVONY

Personnellement ce projet m'a beaucoup apporté en terme de qualité de travail. J'ai par exemple appris à rendre les tâches qui m'étaient assignés en temps et en heure. De plus j'ai apprécié travailler en groupe. En effet cette méthode de travail est bien plus motivante et ludique que le travail personnel. Du fait de mon année au lycée et de mon parcours la partie de "coding" à été un peu difficile à mes yeux. Vu que je viens d'une terminale S et que j'avais très peu d'expérience dans le domaine informatique certaines notions et techniques me paraissaient compliquées à appréhender. Le fait d'être dans un groupe m'a permis de mieux comprendre grâce aux explications de mes camarades. En ce qui me concerne j'ai eu beaucoup d'idée en terme de contenu, de création de carte et d'ajout de gameplay. J'ai notamment créer une grande quantité de carte pour pouvoir avoir le plus de

diversité possible.

9.4 Hadrien "Hawk" NAVARRO

Ce projet est, sans aucun doute, le plus grand, le plus long, et le plus dur que j'ai jamais réalisé, du à son écoulement sur tout un semestre et à la masse de travail nécessaire à son aboutissement. Par rapport à nos attentes vis-à-vis de notre jeu, je dirais que nous avons au final un jeu qui ressemble à ce que nous voulions, sans pour autant l'être exactement, peut être avions nous visé trop haut. Malgré cela, je suis très fier du travail accompli et du résultat qui est plus que satisfaisant. J'ai pu acquérir tout au long de ce semestre de nombreuses connaissances, autant en matière de programmation qu'en travail d'équipe. Grâce à mes partenaires j'ai pu apprendre beaucoup plus vite et j'ai pu construire moi même des choses que je ne me pensais pas capable de faire. Si nous devions avoir à refaire un tel projet, je pense que celui ci pourrait être bien meilleur, car au lieu d'avancer bout à bout comme nous l'avons fait et de se débrouiller pour rajouter des fonctionnalités de manière un peu "sale", nous pourrions implémenter tout le code nécessaire directement très proprement, et nous aurions alors beaucoup moins de temps à passer sur des choses qui peuvent paraître futiles. Pour cela je considère que le projet Mulligan m'a été bénéfique et que grâce à lui j'ai pu grandement progresser sur plusieurs plans.

10 Conclusion

Ce projet a pour nous tous été une grande et laborieuse aventure, qui nous a tous été bénéfique. Bénéfique sur plusieurs plans, au niveau programmation car nous avons découvert plein de nouveaux outils, qui se sont avérés indispensables, comme Unity, Git ou encore Visual Studio qui nous ont permis de réaliser notre jeu. Nous avons évidemment, comme tout débutant, passé plus de temps à corriger des bugs sur unity, ou des algorithmes sur nos scipts qu'autre chose mais cela nous a permis de progresser et faire en sorte que cela n'arrive plus désormais. Faire des erreurs est uné étape nécessaire de l'apprentissage, nous pouvons donc dire que nous avons beaucoup appris.

Ce projet à également été bénéfique au niveau du travail de groupe, nous avons appris à s'organiser, à travailler en équipe sans interférences. De ce coté ci, nous n'avons pas eu de réels problèmes de groupe ou de désentente, la coordination et l'entente ayant été bonnes du début jusqu'à la fin.

Nous sommes fiers de notre jeu, de ce que nous avons accompli en un semestre, mais nous avons le sentiment que si nous avions à refaire un projet de cette envergure, qui pourrait être également un jeu, celui-ci serait bien meilleur car nous n'aurions plus de difficultés et de pertes de temps sur les bases du jeu.

Au final, notre point de vue n'a pas réellement changé par rapport au début de notre projet. Nous étions toujours aussi excité qu'à nos débuts lors de ces derniers mois. Dès que nous arrivions a debugger une partie de notre code nous étions extrêmement satisfaits. En outre lorsque nous pouvions jouer à notre jeu sans que ce denier ne bug, nous étions vraiment heureux. Même si notre jeu n'est pas exactement celui que nous avions imaginé, nous restons fiers de ce que nous avons

fait et espérons qu'il vous plaira.

11 Annexes

Voici quelques aperçus du jeu :

