

Regnorm - Rapport de Deuxieme Soutenance

Groupe PizzaBièreCafé:

CHEYNET Samuel
DUJARDIN Tristan
MICHALET Alix
PINGLOT Thomas

February 2021



Table des matières

1	Introduction : Récits individuels	2
1.1	Samuel Cheynet	2
1.2	Tristan Dujardin	3
1.3	Thomas Pinglot	4
1.4	Alix Michalet	5
2	Avancements du Projet	6
2.1	Git	6
2.2	Gameplay	7
2.2.1	Tour Par Tour	7
2.3	Physique	8
2.3.1	Dash	8
2.3.2	Super Saut	8
2.3.3	Amélioration	8
2.3.4	Destruction des Tiles	9
2.4	Armes	10
2.4.1	Armes	10
2.4.2	Projectiles	10
2.4.3	Bombes	11
2.5	Terrains	12
2.6	Animations	13
2.7	Reseau local	13
2.8	Intelligence Artificielle	15
2.8.1	Pourquoi une IA ?	15
2.8.2	Recherche d'une IA adaptée	15
2.8.3	Un début d'IA	16
2.9	Ambiance Graphique et sonore	17
2.9.1	Ambiance Graphique : ajout d'ennemis, animations et décors	17
2.9.2	Ambiance Sonore : amélioration de la musique initiale	18
3	Récit de groupe	20
3.1	Bonnes découvertes et points positifs	20
3.2	Peines et points négatifs	20
4	Conclusion	21

1 Introduction : Récits individuels

1.1 Samuel Cheynet

Cette période était de loin la plus éprouvante de l'année pour jusqu'ici, entre les examens, des TP d'IP de plus en plus chronophages, les exposées et la partie technique du projet. Ce deuxième tiers du projet a été signe pour ma part de beaucoup plus de programmations et de beaucoup moins d'art. Ma plus grande difficulté est venue du grand manque de ressources pour la version actuelle de Unity, le peu de tutoriels trouvables en ligne est souvent sur une ancienne version avec une interface différente. J'ai eu l'occasion de comprendre à quel point nos TP d'Informatique Pratique sont "beginner friendly" en nous donnant toutes les fonctions et des explications. Ce que ne propose malheureusement pas Unity. Cela a rendu mon apprentissage très compliqué et semé d'embûches. J'ai fait de mon mieux pour créer une mécanique d'armes à distance avec un système de projectile, qui peut faire des dégâts. Et pour avoir des cibles sur lesquelles tirer, je me suis remis au pixel art pour faire un sprite d'un ennemi, j'en ai aussi profité pour faire les animations et les intégrer. Dans le peu de temps qu'il me restais j'ai essayé de rendre la première version de notre musique de fond plus agréable. Cette partie du projet a été de loin la plus éprouvante pour moi, car contrairement à la réalisation des graphismes et de la musique, où il y avait beaucoup de ressources en ligne. La réalisation des armes, a été très compliquée et chronophage a cause notamment, du manque de ressources utilisables.

1.2 Tristan Dujardin

Après avoir fait beaucoup de recherches et de documentation sur le projet, commencé sa creation et defini nos objectifs finaux avant la première soutenance, je pense que chacun dans le groupe a réalisé ce qu’il voulait vraiment faire dans ce projet. Cela nous a mené a changer en profondeur la répartition des taches. Au final, j’ai donc pris la responsabilité de m’occuper de la gestion du terrain et du gameplay.

Lors de cette seconde partie, je me suis donc concentré sur la gestion d’une partie :

La premiere étape a ete de régler les problèmes et bugs qui persistaient après la premiere soutenance, en particulier sur le git, les points de vie du joueur, la physique du terrain et le système de tour par tour.

Après cela, j’ai pu commencer a travailler sur mon objectif de cette seconde période : le système de destruction des tiles a l’aide des armes. Cette implémentation fut assez difficile, pour des raisons que je developperai dans les Avancements du Projet, mais c’est quelque chose qui m’a permis d’identifier beaucoup de défauts dans ma façon d’utiliser Unity, et je trouve le système de Prefab très intéressant à manipuler.

Finalement, entre les deux soutenances, j’ai eu un emploi du temps assez chargé dans toutes les matières, en plus des mid-terms et du test TOEFL, mais malgré cela, je suis satisfait d’avoir atteint les objectif que je m’était fixé pour cette période.

1.3 Thomas Pinglot

Après la première soutenance où les bases du jeu étaient posées, on a commencé à implémenter les plus grosses parties de notre jeu à savoir la conception des armes, le réseau, la physique de colision avec le terrain et une intelligence artificielle. On a dû réorganiser la répartition des tâches car après la première soutenance certains se sont retrouvés sans partie attitrée, de ce fait de mon côté je me suis occupé de faire l'intelligence artificielle des ennemies. Il y a eu dans un premier temps une longue partie de recherche d'idées sur comment faire notre IA, à quoi veut-on qu'elle ressemble tout en restant accessible de réalisation. Une fois l'idée trouvée, j'ai d'abord essayé dans un premier temps des fonctions simple permettant d'interagir avec un personnage en jeu de manière autonome, comme par exemple le faire venir vers nous si on entre dans son périmètre. Cela était surtout pour faire des premiers test avec une IA basique, car dans un jeu comme le notre cela n'a pas trop de sens de faire ce genre de déplacement, il fallait donc trouver quelque chose de plus poussé, basé sur le fait qu'un personnage ne peut jouer que pendant son tour.

De ce fait, entre la première et la deuxième soutenance, mon travail sur le projet fut principalement d'avancé sur l'IA afin d'obtenir au moins un début d'IA fonctionnel qui pourra évoluer avec les nouveaux décors.

1.4 Alix Michalet

Après la première période de réalisation du projet, nous avons pris de l'avance sur certaines parties mais du retard sur d'autres. Compte tenue des recherches et des renseignements que j'avais trouvé pour avancer dans mes parties respectives je n'étais pas si mal parti mais avec le temps je me suis rapidement rendu compte que comprendre et apprendre ce n'était pas du tout la même chose que réussir à réaliser les codes et à faire fonctionner le tout. Depuis le début du semestre, mon moral a été rudement mis à l'épreuve et ma motivation pour les cours ainsi que pour le projet en a pâtit. Nous avons accumulé les retards, les complications et les problèmes liés au projet après la première soutenance. Ces événements se sont rajoutés à mes problèmes personnels avec les cours et ma scolarité ce qui a rendu l'avancement du projet encore plus complexe qu'il ne l'était déjà. Nous avons quand même réussi, malgré les complications et les problèmes auxquels nous avons du faire face avec Unity et Git, à avancer de manière raisonnable dans le projet. Chaque partie du projet étant relié, nous avons besoin « d'avancer groupés » de manière simultanée afin d'éviter au maximum les bug Git. La campagne a été mise à l'épreuve car nous devons développer l'IA, le tour par tour ainsi que les niveaux afin de produire un game-play satisfaisant. Néanmoins, malgré mon retard en IP et mon manque de pratique j'ai réussi à faire quelque chose de buvable qui mérite d'être amélioré mais qui nous permet d'avancer dans la réalisation globale du projet. En résumé, mon « décrochage » a pas mal handicapé le projet mais rien n'es irrécupérable et nous avons une grande marge de progrès.

2 Avancements du Projet

2.1 Git

Tout d'abord, reprenons là où nous en étions restés. Nous avions trois branches Git, chacune avec une version différente du projet. Notre problème était qu'il nous était impossible de les rassembler sur le Git. Cependant, peu après la première soutenance, alors que Tristan essayait de régler les conflits entre les versions, il a trouvé la cause : le `.gitignore`.

En effet, pour ne pas polluer notre Git de tous les fichiers de cache de Unity, nous utilisons le fichier `.gitignore` fourni par GitHub fait spécialement pour Unity. Ce fichier précise qu'il doit être placé à la racine du projet Unity, mais par une erreur de compréhension, Tristan l'avait placé dans le dossier contenant la racine à la création du git. La première chose qui a été faite lors de cette période a donc été la création d'un nouveau git, avec cette fois le `.gitignore` placé dans la racine Unity. Cela était nécessaire car le premier était complètement pollué de fichiers de cache et autres, générés par Unity et qui bloquaient le rassemblement des versions. Ainsi, nous avons pu réellement commencer à travailler en équipe.

Le merge a ensuite été fait par Samuel et Thomas, qui ont rassemblé tout ce qui avait été fait par le groupe en une scène de menu et une scène principale.

2.2 Gameplay

2.2.1 Tour Par Tour

Nous avons déjà un système de tour par tour fonctionnel lors de la première soutenance, il fonctionnait en activant et en désactivant les updates de 4 Joueurs déjà présents sur la scène. Cependant, pour implementer l'IA et le Reseau par exemple, nous avons besoin d'un autre système. En effet, pour pouvoir avoir un nombre de joueur variable et la possibilité de jouer en Reseau, notre scène doit "créer" les joueurs au début de la partie. Il nous fallait alors implémenter un système de tour par tour sur ces joueurs créés.

Thomas et Tristan avaient déjà commencé à y travailler avant la première soutenance, mais il n'était alors pas fonctionnel et les joueurs étaient créés en dehors du Terrain. Cela était dû à notre manque d'expérience de la fonction "Instantiate", qui clone un objet appelé "Prefab" à une position et une rotation donnée. Après avoir appris à se servir de cette fonctionnalité d'Unity et l'avoir testée, Tristan a créé un Prefab pour le joueur et un système de tour par tour basé sur une Liste de joueurs créés par la fonction "Instantiate".

Chaque tour dure au maximum 20 secondes, au cours desquelles le joueur est libre de se déplacer où il le souhaite sur le Terrain, et peut effectuer une attaque. Attention toutefois, lorsqu'un joueur déclenche une attaque, il ne peut plus se déplacer et son tour se termine.

2.3 Physique

Samuel s'est occupé de la partie physique.

2.3.1 Dash

Pour pouvoir améliorer la mobilité horizontale des personnages. J'ai implémenté un dash, qui permet au personnage le possédant de faire un rapide mouvement dans une direction tout en évitant la gravité en appuyant sur une touche. Cette ajout va permettre aux personnages d'aller à des endroits ou ils n'auraient normalement pas eu accès.

2.3.2 Super Saut

Et pour améliorer la mobilité verticale, j'ai implémenté un "Super Saut" qui permet au personnage qui le possède de faire un saut avec une hauteur augment en appuyant sur une touche. Ce qui peut permettre au joueur de sortir son personnage de situations périlleuses.

2.3.3 Amélioration

Nous projetons d'implémenter un système d'inventaire pour tous les personnages, les différentes armes et les améliorations de déplacement en feraient partie. Ainsi pour pouvoir faire une dash ou un super saut il faudra remplir une condition, que ce soit ramasser une boîte ou survivre un certain temps. Les dash et super saut de seront donc pas accessibles à tous les personnages sans conditions.

2.3.4 Destruction des Tiles

La Destruction des Tiles était la partie principale de Tristan lors de cette période. Notre objectif était d'avoir un terrain qui évoluerait au cours d'une partie, il pourrait être affecté par différentes actions du joueur. En particulier, nous voulions des armes capables de déclencher des explosions qui détruirait le terrain dans leur rayon d'action.

Pour implémenter cela, la première chose a été de créer un Prefab pour l'explosion que l'on pourrait ensuite cloner quand on en a besoin. Ce prefab est tout d'abord composé d'un détecteur de collision circulaire qui va nous permettre de savoir quelle partie du terrain est en contact avec l'explosion.

Ensuite, on donne un script à ce prefab, qui va faire 2 choses :

- Lorsque le détecteur de collision de l'explosion est en contact avec un joueur, elle lui inflige 70 points de dégâts.
- Lorsque ce détecteur est en contact avec un tileset destructible, pour chaque tile en contact avec l'explosion, on convertit la coordonnée du point de contact en coordonnée dans le tileset, et on enlève la tile à cette position.

Ainsi, nous avons un prefab qui nous permet de créer une explosion circulaire qui détruit le terrain autour d'elle.

2.4 Armes

2.4.1 Armes

Samuel à donc cherché à implémenter une arme à distance. Pour l'implémentation d'une arme à distance, deux options se proposaient à moi. Utiliser des "prefabs" qui utilisent des projectiles (méthode plus graphique et plus couteuse), ou des "raycasts" qui utilise des lasers (méthode moins graphique mais moins couteuse). J'ai choisi la première méthode car je compte faire à terme plusieurs armes qui aurons différents projectiles qui se comporterons de différents manières. j'ai donc crée un "Prefab" pour un projectile qui possède différentes propriétés tel que la vitesse ou l'importance de la gravité.

2.4.2 Projectiles

Pour l'instant le projectile n'est qu'une boule blanche, qui n'est pas affectée par la gravité et qui est plus rapide que le personnage. Au contact d'un mur ou d'un ennemi le projectile disاپrait et inflige des dégâts s'il touche un ennemi. Je ferai par la suite un visuel en pixel art pour le projectile, mais aussi pour son explosion lors qu'il touche un ennemi ou un mur. Par la suite les différents armes seront différenciées par différents projectiles possédant différentes caractéristiques.

2.4.3 Bombes

La dernière arme que nous avons réalisé lors de cette période est une bombe, créée par Tristan, et qui a évidemment pour principal but d'exploser.



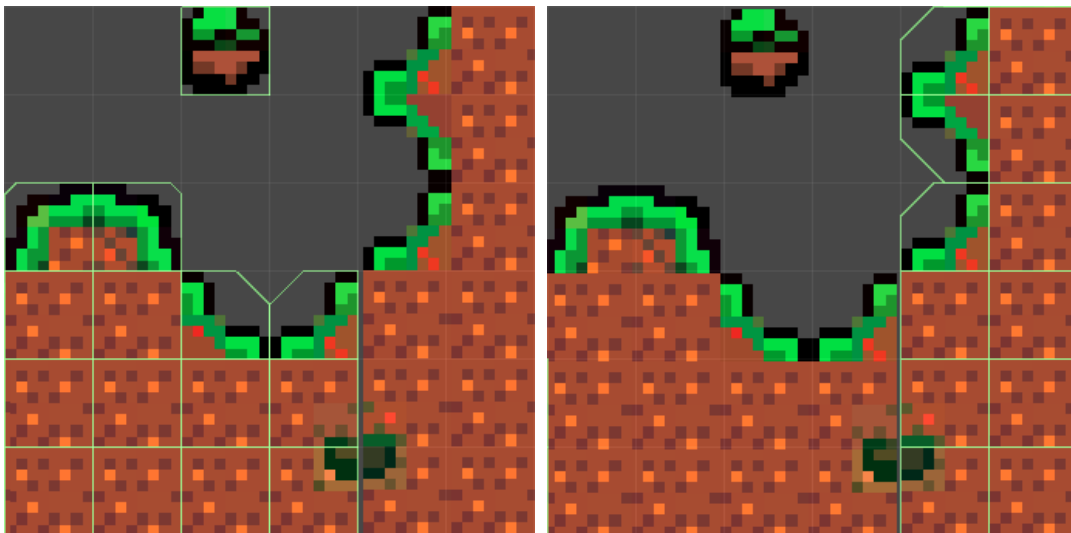
Cette bombe est légèrement différente des autres armes et ne fonctionne pas de la même manière. Lorsqu'un joueur pose une bombe, son tour n'est pas immédiatement terminé, il a encore une seconde ou il peut bouger pour éviter l'explosion. Pour pouvoir gérer ce temps en plus, la bombe n'est non pas créée dans le script du joueur mais dans celui de la partie.

Cette bombe est un Prefab qui est cloné avec la commande Instantiate a la position du joueur. Elle dispose d'un script qui la fait exploser au bout d'une seconde. Cette explosion est en réalité un appel a la fonction Instantiate qui va cloner un Prefab Explosion de Rayon 3.5.

Ainsi, on a une arme assez stratégique, la bombe est une arme risquée mais qui permet de modifier le terrain a sa guise et d'infliger de gros dégâts a ses adversaires.

2.5 Terrains

Pour les terrains, la première chose qui était à faire était de corriger un problème qui venait d'un fix de Tristan : le changement de physique pour que les personnages ne restent pas accrochés aux murs a également remplacé la physique du sol. La friction du sol avait donc été effacée, et les déplacements des personnages étaient devenus "glissants". Pour cela, Tristan a séparé l'ancien Tileset en 2 : un pour les murs, qui garde la physique sans friction, et un pour le sol, qui utilise la physique classique d'Unity.



Tileset du Sol et Tileset des Murs

2.6 Animations

Samuel s'est chargé de la création des animations. Dans mes recherches antérieures j'avais découvert que l'animation est un monde à part entière où des personnes se filment pendant plusieurs heures pour faire un mouvement de quelques secondes. Pour pouvoir dormir et réussir mon année j'ai donc très rapidement fais le choix de faire des personnages de 8 pixels sur 16, pour ainsi faciliter les animations. J'ai donc rélisé l'animation de la courses du saut et du coup de poing. Grâce au "faible" nombre de pixel (8 par 16) à animé, les animations des personnages n'a pas était trop chronophage. Ces ajouts on un important effet dans l'art du jeu et permettent de le rendre plus vivant et naturel. La parite animations et presque finit car il ne reste maintanat plus que les animations de mort, de super saut et de dash.

2.7 Reseau local

Lors de la dernière soutenance de notre projet, l'aspect réseau n'étais pas évolué à cause du « désordre » dans lequel nous avons avancé le projet et à cause de l'avancement ou du retard de certaines parties. Il pouvait plus être qualifié de multijoueur en tour par tour sur une seule console, ce qui ressemblais, dans le fonctionnement, à un jeux flash en ligne. Pour cette soutenance nous avons essayé de faire en sorte que le tour par tour fonctionne sur une base chronométrée afin qu'il soit défini de manière fixe pour faciliter le passage d'un joueur à l'autre dans la partie réseau. Pour le réseau nous avons essayé de faire un réseau local afin que l'on puisse se connecter à une même partie à partir de plusieurs moniteurs. Dans le

développement du réseau, j'ai rencontré beaucoup de difficultés dans la manière d'héberger les parties : fallait-il choisir un serveur, pouvions nous le faire depuis notre site, ou bien un des pc des joueurs pouvait-il suffire ? J'ai donc opté pour la dernière option puisque le jeu n'est de toutes façons pas destiné au grand public et ne servira qu'en parties « locales ».

Les principales difficultés auront été de faire en sorte que le jeu reste agréable à jouer même lorsque l'on est pas actif (lorsque c'est le tour d'un autre joueur), faire en sorte que le jeu reste fluide même à plusieurs ou encore s'assurer une connexion fiable à la partie. Pour ce faire, nous avons dû adapter les mouvements de la caméra afin qu'ils suivent le joueur actif et que tous les joueurs aient la même P.O.V (point of vue). Nous avons également dû faire des modifications sur les scripts que nous avions écrits pour le tour par tour afin que ce ne soit pas qu'un seul PC qui ai le contrôle de tous les personnages/joueurs tout au long de la partie.

2.8 Intelligence Artificielle

2.8.1 Pourquoi une IA ?

Regnorm est un jeu jouable en multijouer contre d'autres joueurs, mais on veut aussi qu'il y ai la possibilité de jouer en solo. Pour cela nous comptons réaliser une campagne qui regroupera plusieurs niveaux avec une difficulté croissante. Or, le principe de Regnorm reste quand meme de jouer contre quelque chose, cela peut etre un humain : dans ce cas la c'est du multijoueurs, ou alors un bot permettant de jouer tout seul.

2.8.2 Recherche d'une IA adaptée

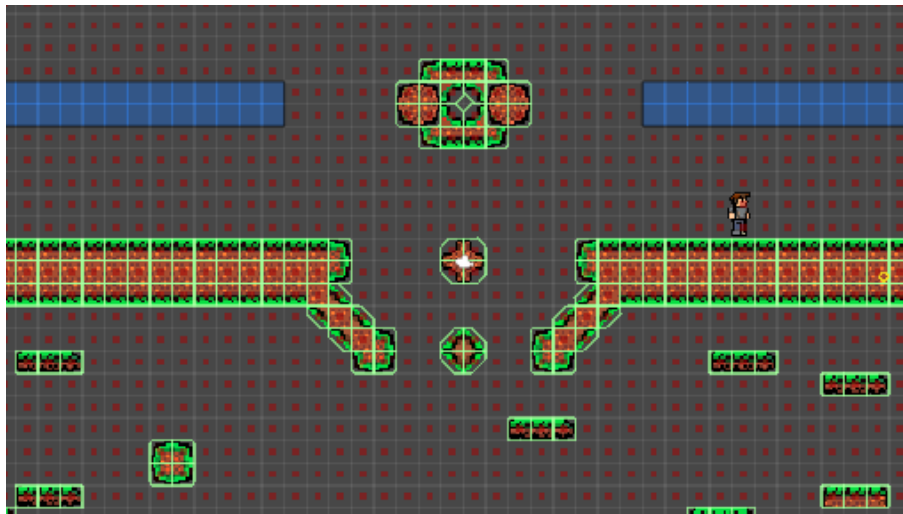
Pour que ces bots puissent jouer, il faut qu'ils soient doté d'une IA adaptée à notre jeu, Thomas s'est donc occupé d'implémenter une intelligence artificielle. La première chose à faire est de déterminer les actions possibles et obligatoires à faire pendant un tour de jeu. Il faut que le bot soit capable de :

- Déterminer quel joueur est le plus proche de lui
- Se déplacer vers le joueur le plus proche de lui pendant le temps de son tour
- Une fois son temps de déplacement terminé, utiliser son arme vers l'ennemi le plus proche

Je pensais bien naïvement que cela n'allait pas être très compliqué de gérer le déplacement du bot vers le joueur le plus proche et que la difficulté principale allait être la partie utilisation d'armes, gérer la précision de tir et le chronomètre du tour. Mais rien qu'essayer de rendre fonctionnel sur une map plate, sans obstacle, le déplacement du bot, cela fut plus compliqué que prévue.

2.8.3 Un début d'IA

Pour commencer j'ai essayé de faire une IA basic, pour voir comment cela fonctionne avec Unity. En mettant une zone de détection à un personnage, si un autre personnage entre dans cette zone il va pouvoir se déplacer vers ce dernier. Mais cela était principalement pour faire des tests et voir les comportements des colliders sur unity. Il a fallu donc une fonction capable d'avoir les coordonnées des joueurs sur la map et de retourner les coordonnées les plus proches au bot qui est en train de faire son tour de jeu. Le plus gros problème est de trouver le chemin vers le joueur le plus proche sachant que le terrain comporte des trous, des plateformes. Grâce à Unity on peut identifier les différents obstacles permettant ainsi de simplifier la recherche de chemin et éviter des collisions avec les murs et les plateformes

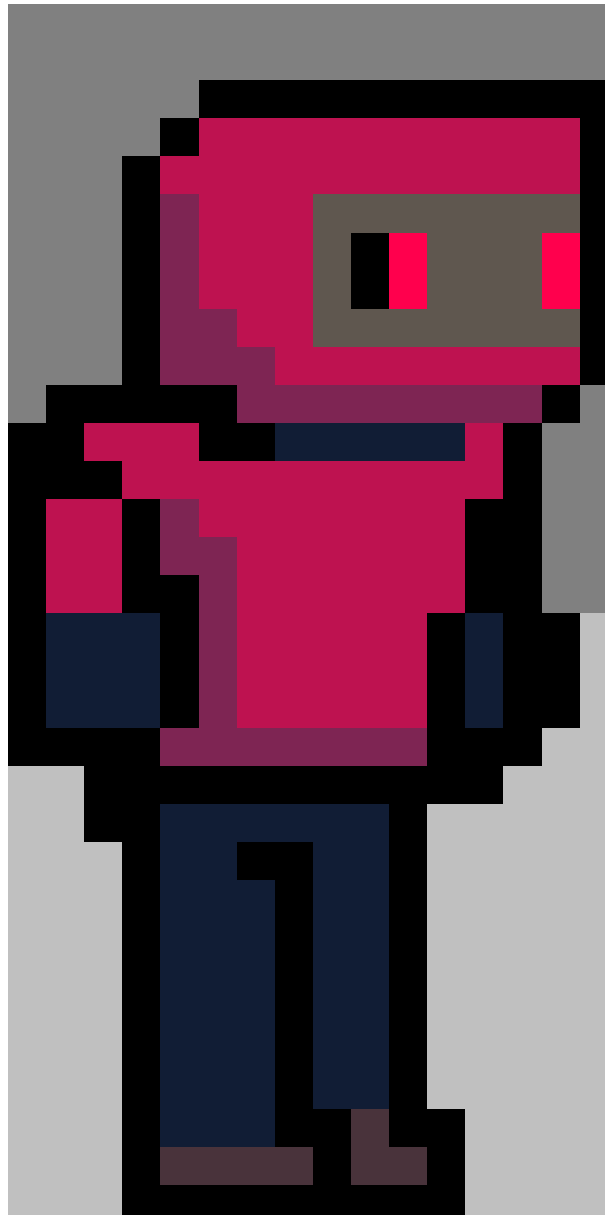


2.9 Ambiance Graphique et sonore

Samuel s'est chargé de faire les Graphique et Sonore.

2.9.1 Ambiance Graphique : ajout d'ennemis, animations et décors

Un des premier ajouts dans la partie à été de faire un ennemi pour nous aider à implémenter les autres parties tel que armes ou IA. Pour ce faire j'ai repris le design de notre personnage, j'ai changé les couleurs pour du rouge et noir étant donné que c'est comme ça que les méchant s'habillent dans les films (et les films de mentent jamais). Et j'ai essayé de rajouter plusieurs petits détails pour rendre le personnage moins humain et ainsi réduire le phénomène d'emphatie du joueur envers les méchants, car il doit avant tout les faire disparaître de notre monde virtuel. Je n'ai pour l'instant réalisé qu'un type de méchant car nous allons faire en sorte que chaque différents méchants possèdent leur propre attrait. J'ai aussi réalisé les animations, que nous avons implémenté, pour rendre notre jeu plus dynamique et améliorer l'art de notre jeu. Finalement j'ai aussi essayé de réaliser d'autres sprites/dessins qui sont destinés à être dans le fond, pour rendre le visuel plus complet et ne pas laisser de trous. J'ai ainsi commencé par l'ajout de différents nuages pour ne pas rendre le ciel répétitif.



2.9.2 Ambiance Sonore : amélioration de la musique initiale

J'ai tout d'abord par recouter la musique que j'avais fait à la première soutenance, et ça devait être le fait d'avoir du recul via le temps que j'avais passé à ne pas l'écouter, mais je l'ai trouvé très désagréable à l'écoute. Il y a avait donc 2 choix qui s'offraient à moi, soit faire de mon mieux pour la rendre plus agréable soit repartire à 0 et recommencer. J'ai choisi la deuxième option car je suis convaicu que c'était la

meilleure option pour arriver à une musique un minimum musicale/agréable. Cependant tout n'était pas perdu car le savoir que j'ai acquis lors de la réalisation de la première musique m'a été très bénéfique et m'a permis de me concentrer sur la réalisation et la création artistique plutôt que les aspects techniques. Les gros changements avec la nouvelle musique sont donc multiples :

- Le rythme est plus lent pour réduire le nombre de notes par minute et rendre la musique plus relaxante et moins stressante
- Réduire le nombre d'instruments, car j'ai fait l'erreur dans la première musique de vouloir mettre trop d'instruments différents ce qui rendait la musique très brouillon, j'ai donc cette fois préféré faire quelque chose de simple et efficace plutôt que brouillon est compliqué.

3 Récit de groupe

3.1 Bonnes découvertes et points positifs

Cette expérience de groupe nous aura déjà énormément apporté alors que nous n'en sommes qu'à la moitié du projet. Nous avons appris à gérer les problèmes les uns des autres, à nous entraider et à trouver des solutions en groupe.

Nous avons découvert qu'avoir une bonne ambiance ainsi qu'une bonne entente au sein du groupe nous permettait d'avancer de manière collaborative et efficace tout en s'amusant sur ce que l'on faisait.

Nous nous sommes également surpris malgré certains désagréments à apprécier l'utilisation de UNITY car, une fois pris en main de manière correcte, son utilisation est satisfaisante et permet un travail productif

3.2 Peines et points négatifs

On a tous été triste d'apprendre que Samuel ne sait pas sauvegarder. Samuel : j'ai découvert l'immense sentiment de vide qu'on ressent quand on se rend compte qu'on vient de se faire perdre plusieurs heures de travail. :(

La prise en main de UNITY nous aura été très utile dans la gestion des physique et le développement des graphismes mais, d'après l'avis général, une utilisation du C dès le départ nous aurait sûrement épargné des souffrances.(exemple : le dash)

Dans le cas particulier de Samuel, il aura également appris, grâce à nous, à régler un four afin de faire réchauffer des pizzas congelées suite à un grave accident : il a fait cramer 6 pizzas. (il ne sais pas différencier le mode GRILL du mode Chaleur tournante)

4 Conclusion

En conclusion, cette période de projet a été relativement productive car nous avons réussi à régler nos problèmes de Git, de pull ainsi qu'à avoir une certaine version stable pour tout le monde. Nous avons également réussi à faire évoluer le projet de manière satisfaisante en créant du gameplay et du contenu. On peut dire que l'on part sur de bonnes bases pour finir le projet d'ici la soutenance finale même si certains aspects du jeu restent améliorables et modifiables. Nous sommes en tout cas fiers d'avoir un jeu qui soit jouable.