

Rapport de soutenance 1

EPITA - *Projet S2*



HERMITA

Produit par Epistars™

ANOUAR BELMDEJENNEH

ANTOINE BLUMENROEDER, BAPTISTE DURRINGER,
DYLAN DE ARAUJO

2022-2023

Table des matières

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Origine | 1 |
| 1.2 | Objectif | 1 |
| 1.3 | Rappel du scénario | 1 |
| 1.4 | Attentes | 2 |
| 2 | Baptiste | 4 |
| 2.1 | Game Design | 4 |
| 2.1.1 | Items et Crafts | 4 |
| 2.1.2 | Ennemis | 4 |
| 2.1.3 | Boss | 5 |
| 2.1.4 | Personnage | 5 |
| 2.2 | Level Design | 6 |
| 2.3 | Implémentation des items | 7 |
| 2.4 | Intégration du craft | 7 |
| 2.5 | Prochaine soutenance | 8 |
| 3 | Dylan | 8 |
| 3.1 | Procreate | 8 |
| 3.2 | Logo | 9 |
| 3.3 | Naissance d'un mages | 9 |
| 3.4 | Le gobelin : un premier ennemi ! | 10 |
| 3.5 | Retour au mage | 11 |
| 3.6 | Prochaine soutenance | 11 |
| 4 | Antoine | 11 |
| 4.1 | Réalisation du site web | 11 |
| 4.1.1 | Choix des librairies | 12 |
| 4.1.2 | Le style | 13 |
| 4.1.3 | La page de blog | 14 |
| 4.2 | Architecture du projet | 15 |
| 4.2.1 | Pour l'inventaire | 15 |
| 4.2.2 | Les statistiques | 15 |
| 4.2.3 | Bases de données | 15 |
| 4.2.4 | Le joueur | 15 |
| 4.3 | L'inventaire | 16 |
| 4.4 | Multijoueur | 17 |
| 4.5 | Console | 17 |
| 4.6 | Prochaine soutenance | 17 |
| 5 | Anouar | 17 |
| 5.1 | Gestion des déplacements | 18 |
| 5.1.1 | Première difficulté | 18 |
| 5.1.2 | Le renouveau | 18 |
| 5.2 | Déplacement au clavier | 19 |

| | | |
|----------|--|-----------|
| 5.3 | Implémentation des animation | 19 |
| 5.4 | Gestion de l'interface | 20 |
| 5.5 | Prochaine soutenance | 21 |
| 6 | Conclusion | 21 |

1 Introduction

1.1 Origine

L'histoire de notre groupe commence lors d'une belle et chaude journée d'automne, les futurs membres se rencontrent dans un parc d'accrobranche à l'occasion de la journée d'intégration organisée par leur BDE. Le futur chef du groupe s'approche des différents membres et leur parle d'un certain projet qui occupera leur deuxième semestre de première année. Les discussions vont bon train alors que nos compères s'amuse à grimper aux arbres. Les premières idées ne tarderont pas à émerger et l'objectif est de réaliser un jeu dans lequel on incarnerait un sorcier en reprenant un grand nombre de codes du style RPG.

Durant cette journée, le groupe participera à une chasse au trésor et c'est au cours de cette dernière que le nom de l'équipe fut adopté. En effet, l'un de nos membres portait sur lui un T-shirt Star Wars et comme la tradition de faire un jeu de mot avec le nom de l'école est de rigueur, le nom « Epistars » fut proposé. Cette journée fut également l'occasion de découvrir les points forts de chacun, tous les membres sont pourvus de compétences complémentaires qui leur seront utiles pour réaliser leur projet. Ainsi nous avons : Dylan un grand artiste, Baptiste un grand joueur, Anouar un grand chef et Antoine un grand matheux. Le groupe se rend compte que leur idée est ambitieuse et ils décident de se mettre au travail plus tôt que prévu par l'école. Dotée d'une forte motivation, nous avons la grande ambition de faire de la société Epistars un acteur majeur du monde vidéoludique dans les années à venir.

1.2 Objectif

Le but que l'on s'est fixé pour cette première soutenance fut d'une part de mettre parfaitement au clair la direction que prendrait le projet. Durant le temps que nous avons passé sur le développement du jeu et notamment au fur et à mesure que la première soutenance approchait des divergences d'opinions ont commencé à apparaître et la nécessité de mettre à fixer les idées fut une de nos priorités. Ce document évoquera le ressenti de chacun sur ce projet en plus de présenter de la manière la plus précise possible l'avancée du projet du point de vue de chacun des membres de l'équipe.

1.3 Rappel du scénario

Ce projet consiste en un jeu que nous réaliserons dans un style pixel art, il se présente sous la forme d'un monde de fantasy fait d'îlots flottants (avec de légères touches steampunk) en deux dimensions avec une vue isométrique. Notre jeu aura une composante housing et appartiendra au genre du RPG. Lorsqu'on lance le jeu, le joueur apparaît dans une zone qui lui servira de base et lui donnera accès aux autres îlots qui constituent son monde. Son objectif sera d'explorer son univers afin de réunir les ressources qui lui seront nécessaires pour faire évoluer sa base tout en ayant à combattre les créatures hostiles qui se dresseront sur sa route.

Dans un monde de magie parsemé d'îles volantes, jouez un mage dont le vaisseau s'est écrasé à l'écart de toute vie intelligente. Accompagnez-le dans une aventure d'action et d'exploration pour l'aider à regagner sa vie normale en l'aidant à réparer son vaisseau au fur et à mesure qu'il progresse en faisant face à la solitude. . .

1.4 Attentes

| | Anouar | Antoine | Baptiste | Dylan |
|---|--------|---------|----------|-------|
| Programmation | | | | |
| Multijoueur | | R | S | |
| Héberger une partie | | X | | |
| Rejoindre une partie | | X | | |
| Combat en multijoueur | | | X | |
| Implémentation des combats | R | | S | |
| Gestion des déplacements | X | | | |
| Lancer des sorts | X | | | |
| Interface | X | | | |
| Intelligence artificielle et gestion des ennemis | | R | S | |
| Apparition des ennemis | | X | | |
| Mécaniques des boss | | | X | |
| Intelligence artificielle | | X | | |
| Fonctionnement du craft et de l'inventaire | | | S | R |
| Intégration de l'établi | | | X | |
| Création des sorts | | | | X |
| Fabrication de l'équipement | | | | X |
| Élaborations des potions | | | | X |
| Inventaire fonctionnel | | X | | |
| Système de sauvegarde | R | | S | |
| Stockage de l'inventaire | X | | | |
| Enregistrement des stats | X | | | |
| Retenir l'état de l'île de départ | X | | | |
| Game design | | S | R | |
| Agencement du monde | | | X | |
| Établissement du housing | | | X | |
| Système de combat | | | X | |
| Visuels et animations | | | S | R |
| Personnage | | | | X |
| Ennemis | | | | X |
| Environnement | | | | X |
| Site web | S | R | | |
| Page d'accueil | | X | | |
| Mise en page | | X | | |
| Système de blog | | X | | |

TABLE 1 – Répartition des tâches sur le projet

R : Responsable || S : Suppléant || X : Se charge de la tâche

| <i>Soutenance</i> | <i>Première</i> | <i>Deuxième</i> | <i>Finale</i> |
|---|-----------------|-----------------|---------------|
| Programmation | | | |
| Multijoueur | | | |
| Héberger une partie | ** | *** | *** |
| Rejoindre une partie | ** | *** | *** |
| Combat en multijoueur | * | ** | *** |
| Implémentation des combats | | | |
| Gestion des déplacements | ** | *** | *** |
| Lancer des sorts | * | ** | *** |
| Interface | ** | *** | *** |
| Intelligence artificielle et gestion des ennemis | | | |
| Apparition des ennemis | * | ** | *** |
| Mécaniques des boss | * | ** | *** |
| Intelligence Artificielle | * | ** | *** |
| Fonctionnement du craft et de l'inventaire | | | |
| Intégration de l'établi | ** | *** | *** |
| Création des sorts | * | ** | *** |
| Fabrication de l'équipement | * | ** | *** |
| Élaborations des potions | * | ** | *** |
| Inventaire fonctionnel | * | ** | *** |
| Système de sauvegarde | | | |
| Stockage de l'inventaire | * | ** | *** |
| Enregistrement des stats | * | ** | *** |
| Retenir l'état de l'île de départ | * | ** | *** |
| Game design | | | |
| Agencement du monde | ** | *** | *** |
| Établissement du housing | ** | *** | *** |
| Système de combat | ** | *** | *** |
| Visuels et animations | | | |
| Personnage | ** | *** | *** |
| Ennemis | * | ** | *** |
| Environnement | * | ** | *** |
| Site web | | | |
| Page d'accueil | ** | *** | *** |
| Mise en page | * | ** | *** |
| Système de blog | *** | *** | *** |

TABLE 2 – Évolution de l'avancement du projet par soutenance

Tâche commencée : * || Tâche avancée : ** || Tâche terminée : ***

2 Baptiste

2.1 Game Design

La première étape de la réalisation de notre jeu après celle du cahier des charges a été de mettre au clair le moindre détail à propos du jeu pour que tout le monde ait la même vision du projet. Pour cela j'ai donc pris en charge la réalisation de documents de game design détaillant chaque aspect du jeu.

Comme nous avons pris la décision de ne faire qu'un seul niveau pour notre rendu du jeu, j'ai pu élaborer facilement des listes exhaustives des items, monstres etc... qui seront disponibles étant donné qu'il n'y aurait pas cette peur d'avoir au milieu du projet l'ajout de contenu qui viendrait chambouler le tout.

2.1.1 Items et Crafts

J'ai donc tout d'abord rédigé une liste de tous les items qui seront obtenables dans le jeu : les bâtons, les robes, les colliers, les potions, les loots élémentaires, les gemmes et les composants d'artisanat. Chaque item y est détaillé avec son utilité, sa rareté, son obtention ainsi qu'une illustration de ma vision graphique de cet item.

Avec la rédaction de la liste des items est venue l'élaboration de la liste des crafts, cette catégorie se découpe en 3 parties distinctes : les crafts de l'Atelier, de l'Établi et du Laboratoire.

Le premier pourrait être qualifié de table de craft classique présente dans beaucoup d'autres jeux, il permet, avec une interface de sélectionner un craft et de l'élaborer avec les composants requis. Il permet de fabriquer l'équipement et les potions.

Le second a suscité plus de réflexion, car je ne voulais pas seulement créer un second atelier qui ne servirait qu'à crafter des gemmes. J'ai donc pensé à l'idée d'ajouter des crafts à durée, par exemple pour transformer 100 poussières d'eau en 1 perle d'eau, il faudra lancer le craft et revenir 1 heure plus tard pour pouvoir récupérer le résultat (l'établi sera inutilisable le temps du craft). Il permettra également d'ouvrir des géodes pour récupérer des gemmes et des composants élémentaires, ceci sera aussi soumis à un temps de craft.

Et pour finir, le laboratoire, ici l'originalité viendra dans le fait que les crafts possibles ne seront pas affichés, chaque sort sera le résultat d'une certaine combinaison d'éléments. Pour découvrir chaque sort, il faudra donc essayer toutes les combinaisons d'éléments, cela rendra l'expérience du nouveau joueur pleine de découvertes tout en laissant les joueurs expérimentés créer le sort qu'ils souhaitent. De plus, selon le tier du matériaux élémentaire choisit le niveau du sort sera différent, le joueur devra donc reconcocter ses sorts afin d'en extraire leurs pleins potentiels.

2.1.2 Ennemis

Après avoir élaboré tous les objets ainsi que tous les crafts, il nous faut un moyen d'obtention de ces objets, ainsi que des composants pour les crafter. Je me suis donc attelé à la réalisation de la liste des mobs. Celle-ci indique leurs noms, statistiques, types, descriptions, loots et illustrations.

La carte étant séparée en 3 grandes zones, chaque zone aura des monstres différents qui suivront chacun les thèmes de zones, par exemple la première zone accueillera des gobelins juniors, des gobelins chercheurs d'or, des gobelins mécaniciens, des robots rafistolés, des golems

de ruines et des inventions gobelines, ceux-ci seront de types foudre et terre. Tandis que la seconde zone sera composée de gobelins juniors, gobelins guerriers, chamans gobelins d'eau, élémentaires d'eau, canards, sangliers et ents corrompus qui seront principalement de types eau et terre. J'ai choisi les monstres principaux de chaque zones comme étant des gobelins pour simplifier le travail de Dylan au niveau des dessins. De plus, chaque zone aura des mobs permettant d'obtenir des éléments différents, ce qui forcera le joueur à adapter ses sorts à chaque zone ainsi que de parcourir toutes les zones pour obtenir tous les éléments différents.

Pour l'apparition des ennemis, elle se fera de 2 manières : les gobelins ainsi que leurs créations apparaîtront à des emplacements prédéfinis comme dans des camps, tandis que les ennemis comme les sangliers, les golems etc... apparaîtront aléatoirement dans les zones. De plus, les gobelins ne réapparaîtront pas tant que l'instance du niveau ne sera pas réinitialisée, alors que les autres ennemis pourront apparaître à l'infini. Cela permettra au joueur d'avancer dans le niveau sans que tous les mobs des camps réapparaissent constamment, tout en permettant au joueur de tuer certains mobs en boucle pour récupérer leurs butins.

2.1.3 Boss

Pour finir, il y aura 3 mini boss et 1 boss dans le niveau. Le mini boss de la zone de ruine sera un roi gobelin cupide, il sera plutôt simple à battre comme il constitue le premier boss du jeu, ils n'aura donc par conséquent pas énormément d'attaques spéciales, simplement des attaques à esquiver pour que le joueur puisse découvrir cette mécanique avant d'arriver face à un réel boss.

Le second mini boss sera un arbre vivant, ce sera la vraie première difficulté du jeu, la première chose qui m'est venu à l'esprit quand j'ai pensé à un boss arbre a été des attaques à base de racines, j'ai donc pensé à des coups de racines venant du sol à esquiver suivants plusieurs patterns, ainsi que l'invocation de petits arbres à éliminer durant le combat.

Le troisième mini boss sera un golem dompté par les gobelins, je l'ai imaginé comme un boss massif qu'il faudra esquiver, car il chargera, frappera le sol et produira des ondes de choc à esquiver avec le dash.

Enfin le boss du niveau, je souhaite vraiment en faire un combat plaisant, difficile et impressionnant à la fois. Il reprendra une mécanique de chaque boss vu précédemment : l'attaque classique du roi gobelin, l'invocation du boss arbre et la mécanique d'onde de choc du boss golem. Et selon moi un vrai boss ne sera pas un vrai boss si il n'avait pas plusieurs phases, c'est pourquoi à partir de la moitié de sa vie il sera plus rapide dans ses attaques.

2.1.4 Personnage

Pour conclure avec l'élaboration du game design, je vais parler du personnage, il n'y a pas eu grand chose à ajouter car la plupart avait été dit dans le cahier des charges. La seule chose restante a donc été de faire la liste des sorts, parmi eux nous avons décidé d'en implémenter 8 par contrainte de temps qui sont : Boule de feu, Jet d'eau, Bouclier de terre, Eclair, Laser de feu, Pluie éphémère, Gros Rocher et Chaine d'éclair. J'ai donc écrit pour chacun d'entre eux : leurs types, portées, vitesse, taille, nombre de cibles, temps d'incantation, temps de recharge, dégâts, description et illustration.

2.2 Level Design

Pour continuer sur le design, je me suis également occupé du level design de l'île du housing ainsi que du premier niveau. Comme établi dans le cahier des charges celui-ci est divisé en 3 îles comportant 4 zones différentes.

Pour l'île du housing la tâche a été plutôt rapide car j'avais presque tous les assets dont j'avais besoin à ma disposition. L'île est donc terminée, il manque seulement les dessins des différents établis et du portail. (voir figure 1)

Pour le premier niveau, la tâche n'a pas été aussi simple, en effet il y a énormément d'assets à dessiner que ce soit pour le terrain, les camps de gobelins, la décoration, les bâtiments etc... J'ai donc commencé par dessiner des assets de pierre (que je modifierai à l'avenir car je ne les trouve pas encore satisfaisant) afin de pouvoir dessiner la forme de ce qui constitue l'île où le joueur apparaîtra. La forme étant dessinée, il ne me manque qu'à attendre et dessiner les assets manquants pour terminer cette île. (voir figure 2)

Pour le reste des îles, j'ai décidé de seulement tracer le terrain et les routes sans relief afin que l'on puisse déjà implémenter tout ce que nous voulons, tout en attendant les assets manquants. J'ai construit la seconde île comme un chemin tout tracé pour le joueur qui lui laisse tout de même la liberté de se balader en dehors de la route, cela permet de guider le joueur vers là où nous voulons qu'il aille sans qu'il se sente forcé à y aller. La partie de ruine est petite, remplis d'obstacles et tous ses camps sont rapprochés pour donner un aspect étroit au lieu. Tandis que la partie forêt est construite de manière très écartée avec énormément d'espace, pour que la forêt qui y sera ajouté puisse donner une impression de densité et que le joueur ait l'impression de pouvoir s'y perdre si il ne suit pas les routes. (voir figure 3).

Pour la dernière île, je n'ai pas encore commencé sa construction car étant principalement constitué de montagnes, sans les assets adéquats je ne serais pas allé très loin.

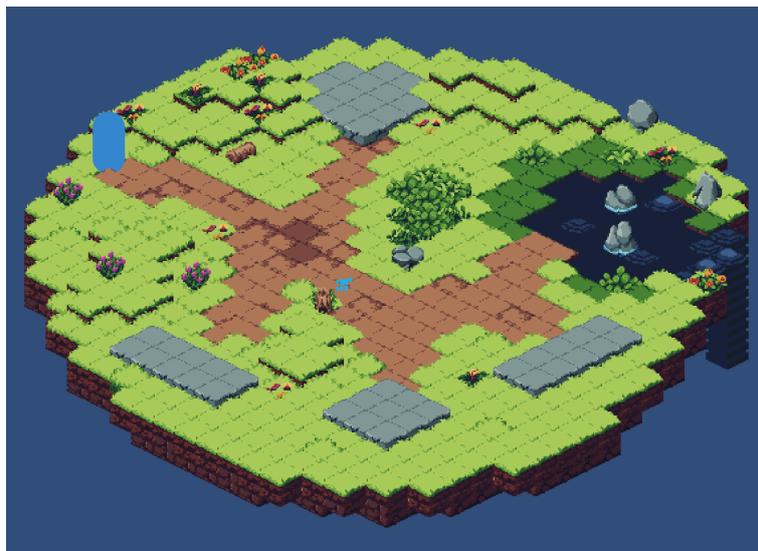


FIGURE 1 – Île du housing

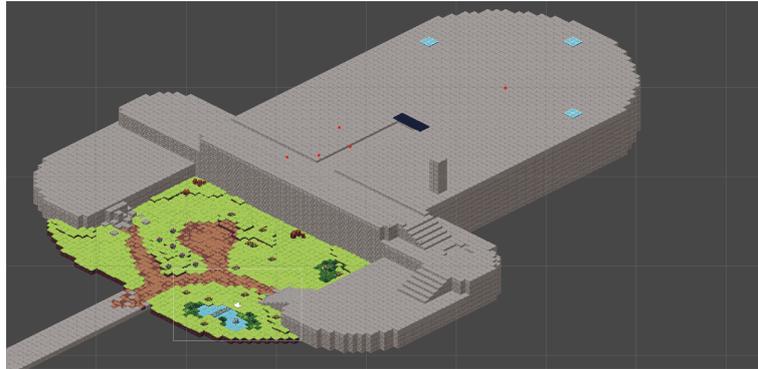


FIGURE 2 – Première île

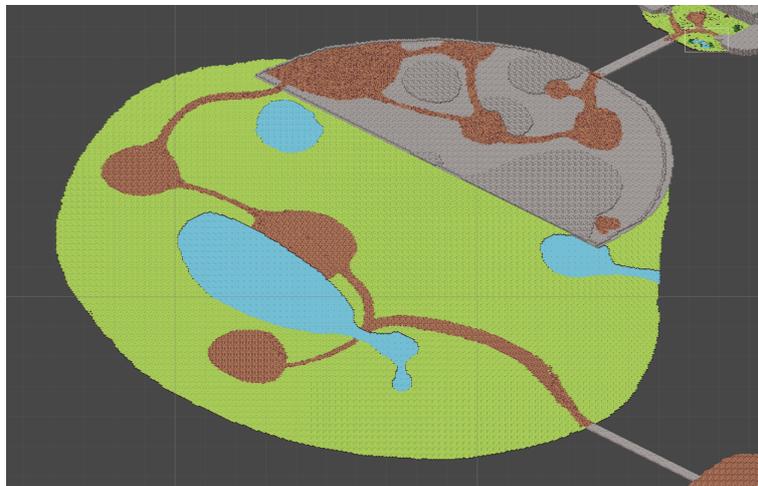


FIGURE 3 – Deuxième île

2.3 Implémentation des items

Suite à l'implémentation de l'inventaire d'Antoine, j'ai pu me lancer dans l'implémentation de tous les items du jeu. Pour cela, j'ai fait le choix de commencer par dessiner les sprites des items. Initialement ce n'était pas prévu que je m'occupe des dessins des items, mais comme cette tâche n'avait été attribué à personne, je m'y suis essayé et ça m'a plu. J'ai donc dessiné tous les items hormis les bâtons et les capes que je dessinerai d'ici la prochaine soutenance. (voir figures 4,5,6)

Après cela, j'ai implémenté tous les items dans le jeu en créant une instance de Item pour chaque objets dans une base de donnée en suivant les caractéristiques indiquées dans le documents liste d'items que j'avais créé.

2.4 Intégration du craft

La chose que j'ai commencé en dernier a été l'intégration de l'établi, j'ai sous-estimé la difficulté de créer une interface de craft et par manque de temps je suis seulement arrivé au stade de pouvoir ouvrir un menu avec une liste de crafts sur la gauche.

Cependant, j'ai également intégré la totalité des crafts de l'atelier, tout comme les items, à l'aide de la classe créé par antoine j'ai pu incorporer au jeu la liste de mon document de liste de



FIGURE 4 – Potions



FIGURE 5 – Poudres

crafts au jeu. Pour cela j'ai simplement créé des instances de Recipe pour tous les crafts dans une database.

2.5 Prochaine soutenance

D'ici la prochaine soutenance la première chose que je vais faire va être de terminer l'intégration de l'établi que je n'ai pas pu terminer pour cette soutenance, l'implémentation d'un système de craft sur la durée va probablement me demander pas mal de recherche.

Une fois que cette tâche sera terminée je commencerai à créer les combats de boss, j'aimerais pouvoir m'y mettre le plus tôt possible car je veux qu'ils soient parfaits. Et en parallèle de cela, je vais aider Dylan à dessiner les différents assets de map et d'ennemis afin de pouvoir continuer le level design.

3 Dylan

3.1 Procreate

Utilisant Procreate depuis plusieurs années, j'ai décidé d'utiliser ce logiciel pour réaliser le graphisme de notre jeu, puisqu'il permet non seulement de dessiner les différents assets, mais aussi de réaliser les animations. Cependant, celui-ci n'est pas conçu pour réaliser des dessins



FIGURE 6 – Perles

en pixel art dans sa version de base. J'ai donc, grâce à de nombreux tutoriels que l'on peut aisément trouver sur internet, créés des pinceaux permettant la réalisation de ces dessins.

3.2 Logo

Le logo, qui est l'une des premières choses que l'on voit sur un jeu, a deux buts principaux. Le premier est de commencer à raconter l'histoire du jeu avant même de débiter la partie. Pour rappelle, le mage s'est écrasé sur l'île avec son dirigeable. Ainsi, quoi de mieux que de représenter un dirigeable en guise de logo ! Le second était de montrer le style de notre jeu : le pixel art. Ainsi, dans cette idée de représentation du style, le logo Hermita est passé par une certaine phase d'évolution, voyant ainsi le nombre de pixels réduire au fil des versions.

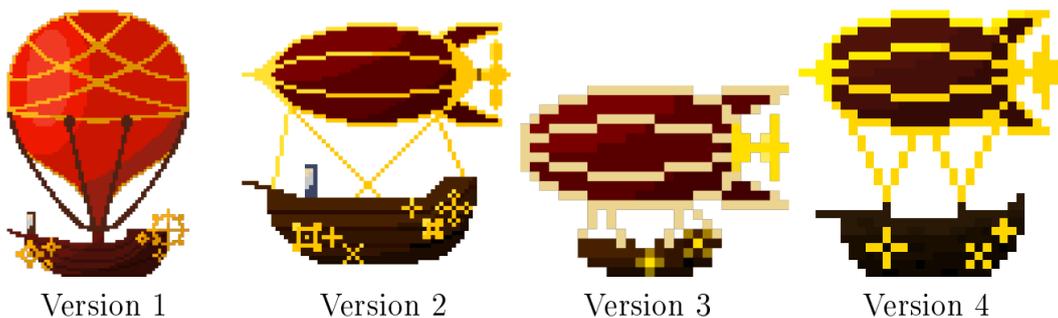


FIGURE 7 – Évolution du logo Hermita

A noter que les couleurs utilisés pour ce logo sont similaires à celles du mage.

3.3 Naissance d'un mages

Ne maîtrisant pas entièrement les codes du pixel art, le processus de création du mage a été un peu long au début. Je suis donc passé par une phase de dessin plus réaliste (puisque c'est un genre auquel je suis plus habitué). Voici le premier croquis réalisé :

C'est seulement par la suite, après de nombreuses tentatives, que j'ai réussi à créer la toute première image en pixel art de notre mage, c'est-à-dire celle-ci :



FIGURE 7 – Perles

Cette première étape m'a beaucoup appris sur la manière de dessiner en pixel art, un monde totalement nouveau si l'on compare cela aux autres genres de dessin. Cela m'a aussi appris à travailler les ombres d'une manière différente. Ces nouvelles méthodes m'ont permis par la suite de débiter la réalisation des différentes animations, afin que le mage puisse exécuter différentes actions dans le jeu. La première animation était la suivante : le faire marcher.

Dans une telle animation, la difficulté réside surtout dans le fait que chaque image soit cohérente avec la suivante. Sachant que l'animation doit se répéter indéfiniment de façon cyclique, il faut aussi que la dernière image de l'animation soit cohérente avec la toute première. Le but étant que le mouvement soit fluide et qu'il n'y ait pas de décalages flagrants entre 2 images consécutives. J'ai donc obtenu ceci :

Nous avons décidé que le mage pourrait avancer sur 8 côtés, il m'en restait donc 7 à réaliser. Ici, de nouvelles difficultés sont apparues. Pour pouvoir réaliser le personnage sur plusieurs côtés, même s'il est en 2D, il faut pouvoir se le représenter en 3D pour ensuite le projeter dans un monde 2D. Les questions à se poser à ce moment présent pour chaque sens de marche du personnage sont les suivantes : Qu'est-ce qui doit être visible ? Ou caché ? Est-ce que les proportions corrélerent entre chaque axe de mouvement ? Les ombres sont-elles bien placées ? Après un certain moment j'ai pu aboutir et obtenir les 8 directions, c'est à dire un total de 64 images :

3.4 Le goblin : un premier ennemi !

Ayant beaucoup appris avec la création du mage, ce nouveau personnage m'a semblé au départ plus facile à appréhender. Même si un détail change un peu la donne. En effet, sur le mage, la cape venait cacher ses jambes, je n'avais donc pas tellement à prendre en compte le mouvement des jambes. Mais ici, c'est différent, puisqu'elles sont bien visibles. C'est en raison de ce détail que cela a mis un certain temps à réaliser cette animation, mais j'ai fini par y arriver. Par ailleurs, pour dessiner le goblin, je me suis inspiré de ceux présents dans Clash of Clans. Voici le résultat :



FIGURE 8 – Perles



FIGURE 9 – Perles

3.5 Retour au mage

Un mage qui marche, c'est bien ... Mais le voir courir, c'est encore mieux. En effet, un personnage qui ne ferait que marcher en jeu rendrait le tout un peu long et ennuyant. C'est pourquoi il fallait que notre mage puisse aller plus vite en courant. Sachant que le gobelin avait lui aussi été animé en train de courir, je savais déjà comment m'y prendre pour le mage. De plus, en ajoutant du mouvement à la cape, j'ai pu rendre le mouvement plus réaliste, et apporter un effet de rapidité. Le résultat ci-dessous :

3.6 Prochaine soutenance

- animer les attaques du mage
- animer les attaques du gobelin
- créer de nouveaux ennemis
- commencer à programmer le système de craft

4 Antoine

4.1 Réalisation du site web

La conception de ce site a subi un grand nombre de va-et-vient, avec plus de lignes de codes supprimées que de lignes écrites... Ayant de bonnes bases en développement web, les difficultés



FIGURE 10 – Perles

rencontrées ne consistaient pas en des problèmes de code mais plutôt des problèmes de goût. Après être passé par des transitions qui donnaient une impression d’animation 3D powerpoint, j’ai décidé de refaire tout le site de manière très sobre pour que ce soit les images du jeu qui donnent l’esprit du site.

4.1.1 Choix des librairies

Le choix du framework svelte a été fait dès le début, principalement parce que c’est mon préféré. De plus, il est très simple à utiliser et à comprendre, ce qui est un plus pour un projet de groupe. Pour la partie serveur, j’aurais bien pris SvelteKit mais je voulais rendre l’écriture des posts le plus simple possible pour motiver toute l’équipe à en écrire. Astro est alors apparu comme une solution attrayante. C’est un générateur de sites statiques qui permet d’utiliser différents frameworks tels que Svelte ou React pour programmer. Mais une fonctionnalité séduisante consiste à utiliser un dossier de fichiers au format markdown comme une collection, ce qui est adéquat pour implémenter un système de blog. Parmi les aventures, je suis passé par une SPA (Single Page Application) mais contrairement à SvelteKit, ce n’est malheureusement pas géré nativement par Astro. C’est pourquoi il a fallu passer par un plugin pour Astro : astro-spa. Cependant, après de nombreuses batailles pour correctement exécuter mes scripts et plus de code consacré à gérer les problèmes de la spa que de code en soi, j’ai pris la décision d’enlever la spa malgré les avantages en termes de performance qu’elle apporte.

Il reste un petit détail maintenant, pour rendre le site un peu plus vivant, j’ai décidé d’utiliser Lenis, une librairie pour rajouter des animations avec le scroll. Malgré quelques connaissances avec la librairie locomotive scroll, son manque d’accessibilité et ses besoins gourmands en performance m’ont fait essayer cette nouvelle librairie. Lenis est la base de la page « blog » mais la librairie étant pauvre, toute la logique a dû être implémentée.

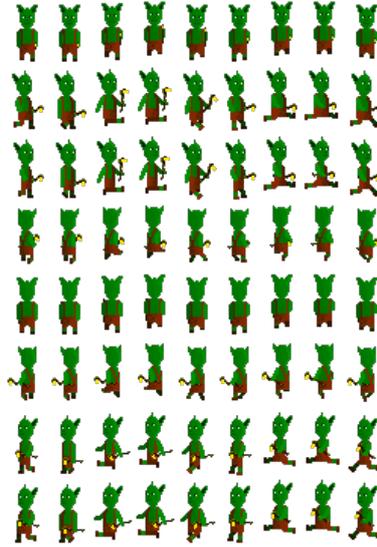


FIGURE 11 – Animation deu gobelin

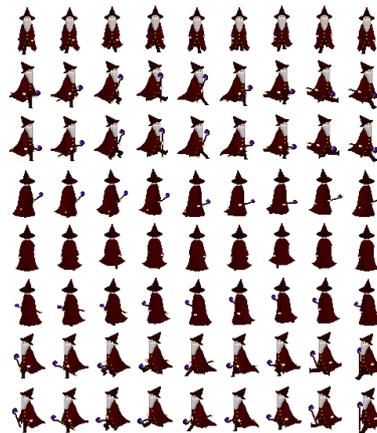


FIGURE 12 – Perles

4.1.2 Le style

Des essais précédents, la leçon a été retenue : Ne pas en faire de trop. Cependant, même si le site se veut sobre, j'ai incorporé quelques touches qui rappellent le jeu. Par exemple, pour mettre l'utilisateur dans l'ambiance d'un jeu vidéo, l'utilisation d'une police en pixel art est pertinente, précisément quand le jeu lui-même est en pixel art. Pour rester dans cet esprit du pixel art, beaucoup d'éléments rectangulaires sont implémentés en évitant d'arrondir les coins des objets. De cette manière, tout est un petit peu « carré », à la manière du pixel art. En plus de rappeler la forme du jeu, je voulais aussi en rappeler l'esprit. Pour évoquer la magie présente dans Hermita j'ai pensé à une fumée bleue qui suit la souris de l'utilisateur. (visible dans la figure ci-dessous)

Et finalement, avec la forme et l'esprit du jeu présents dans le site, il ne manque plus qu'à montrer un bout du monde d'Hermita. Pour cela, le monde étant peuplé de golems, le tout dans un style légèrement steampunk, j'ai pensé à mettre des roues dentées en bas à droite de l'écran ; qui tournent avec le scroll. En plus de cela, je compte implémenter des îles flottantes

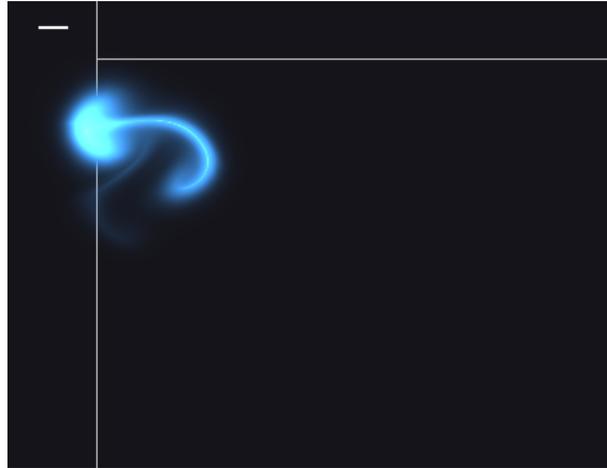


FIGURE 13 – Effet de fumée

tirées de notre jeu qui bougeraient avec un effet de parallaxe, comme sur la page d'accueil (qui a des exemples avant de recevoir des images plus finies du rendu du jeu).

4.1.3 La page de blog

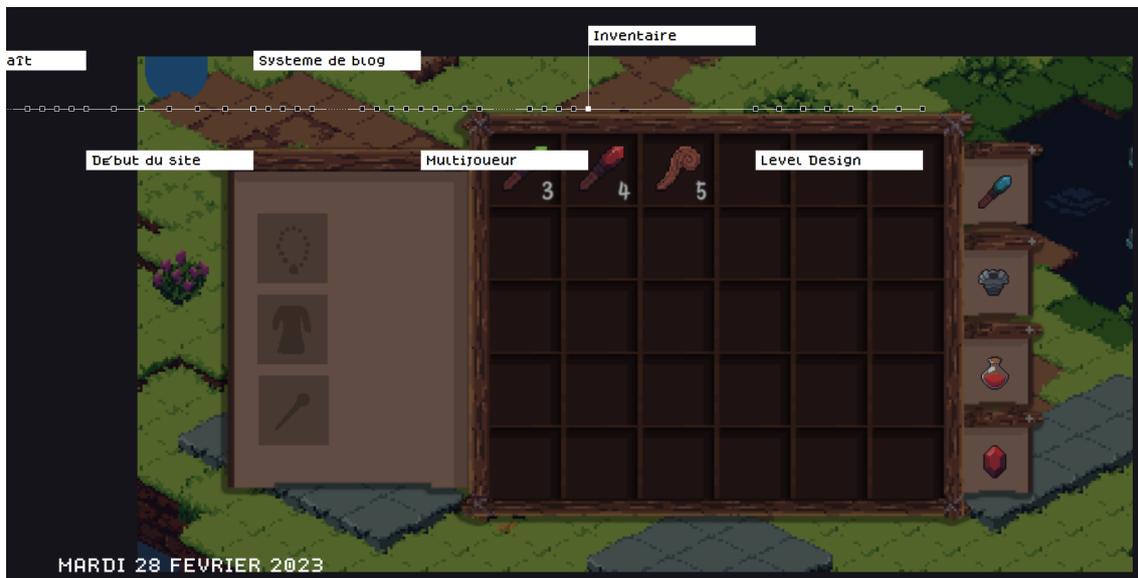


FIGURE 14 – Frise chronologique

Le blog prend la forme d'une frise chronologique avec un scroll horizontal, courant dans les sites modernes qui se veulent « classe ». J'ai dû beaucoup bidouiller avec Lenix en écrivant une fonction qui devinait quelle poste était visible à l'écran à tel endroit du scroll. Ceci est nécessaire pour faire la petite animation et pour mettre la date et l'image de chaque poste dans la zone d'affichage commune à tous les posts. Entre les posts se trouvent des carrés de petite taille, ils représentent le nombre de jours passés entre chaque post et une ellipse est représentée lorsque le nombre de jours dépasse 10. On peut ensuite cliquer sur chaque étiquette de post

pour lire le texte. Cette ouverture se fait avec une transition poussée pour donner un effet plus vivant.

4.2 Architecture du projet

Ayant été nommé chef d'architecture et ayant touché aux tâches qui avaient besoin des fonctionnalités de base en premier, j'ai eu le plaisir de faire cette intégration. Ce travail ressemblait un peu aux TP qui nous sont donnés l'EPITA mais sans sujet et sans moulinette.

4.2.1 Pour l'inventaire

Les items sont découpés en 3 classes : Item, Equipment et Consumable. Item est la classe mère de ces deux dernières. Elle contient les propriétés communes à tous les items : nom, description, icône, etc. Equipment et Consumable contiennent les propriétés spécifiques à ces deux types d'items. Par exemple, Equipment contient les propriétés de l'équipement (armure, dégâts, etc.) et Consumable contient les propriétés du consommable (effet, durée, etc.).

J'ai aussi implémenté des recettes pour avoir un squelette de base mais ce sera une tâche à finir plus tard.

4.2.2 Les statistiques

Pour gérer ces statistiques, j'ai créé une classe Stats qui est un dictionnaire permettant de stocker des flottants associés à des chaînes de caractères tout en connaissant le type de statistique. En effet, il faut différencier hp +50 et hp +50% par exemple. Cela permet de faire des calculs plus complexes sur les statistiques tout en gardant une certaine flexibilité. Ainsi, chaque objet qui interagit avec les statistiques est une instance de Stats : equipment, consumable, player, etc. J'ai aussi implémenté une classe FlatStats assez similaire mais sans la notion de pourcentage. En effet, dans la base de donnée on peut avoir un équipement qui donne +10% de vie (hp) mais pour optimiser les calculs, +10% est « aplati » en fonction de la vie du joueur. Pour ces calculs j'ai défini les opérateurs + et - et d'autres fonctions telles que Flatten et Ceil.

4.2.3 Bases de données

Pour ajouter des items ou des recettes, il suffit de modifier les fichiers qui agissent comme des bases de donnée. Ces bases de données renvoient l'objet associé à son id. Chaque item (ou recette) n'est instancié qu'une fois puisque ses propriétés ne changent pas. Ce n'est juste pas le cas pour l'équipement dont une nouvelle instance est créée dès que l'objet est récupéré dans la base de données puisque ses stats subissent un passage à l'aléatoire. Ces scripts sont rattachés à un GameObject dédié aux bases de données. Les constructeurs d'Item et de Stats sont surchargés et simplifiés (voir exemple ci-dessous) pour rendre la tâche du game designer plus aisée.

4.2.4 Le joueur

Pour associer un réel objet à toutes ces statistiques, j'ai créé la classe Player qui permet d'avoir des statistiques de base, un équipement et des effets. Je gère aussi une boucle de régénération pour que toute statistique de la forme *-regen soit une valeur de régénération par

```
new Equipment(  
  id: 6,  
  name: "Bling bling king cape",  
  description: "The cape of the goblin king living in the ruins",  
  tier: 'C',  
  set1[83],  
  StuffType.Robe,  
  stats: "res +10%, spd +10%, atk +10%"),
```

FIGURE 15 – Constructeur Equipment

seconde. C'est ici que sont définies les fonction Equip pour l'équipement et UseConsumable pour les potions.

4.3 L'inventaire



FIGURE 16 – Inventaire

Toutes ces bases ont été développées dans le but de rendre la réalisation de l'inventaire possible. Bien que la tâche ne devait être que commencée, j'ai tout de même voulu rendre l'inventaire du joueur fonctionnel pour pouvoir implémenter les vraies fonctionnalités par la suite. Les joueurs n'ont pas de coffre, il est donc important de leur fournir un inventaire complet en fonctionnalités. C'est pourquoi déjà maintenant, on peut choisir de montrer "tout", "équipement", "consommables" et "ressources". En plus de ce filtre, on peut trier par ordre alphabétique, par rareté et par quantité, chacun dans l'ordre croissant ou décroissant. L'inventaire est infini et n'est donc pas constitué de *slots* comme dans la plupart des jeux mais simplement d'une liste chaînée. Dans l'UI, il a ainsi fallu mettre les items dans quelque chose où l'on puisse scroller de manière théoriquement infinie.

Les items sont représentés dans l'interface par des instances de la classe `UIItem` qui instancie un prefab dans l'inventaire. Pour interagir avec eux, on peut passer la souris dessus ce qui affiche un petit encadré avec plus d'informations sur l'item et éventuellement ses statistiques. En faisant un clic droit sur un équipement on peut l'équiper et en faisant un clic droit sur une potion on peut l'utiliser. Les statistiques du joueur sont affichées sur la gauche et actualisées grâce à un évènement déclenché dans la classe `Player`.

4.4 Multijoueur

Le multijoueur possède les fonctionnalités pour héberger et rejoindre une partie. Le travail a été plutôt simplifié par Unity Relay puisqu'il existe nativement un système de parties protégées par un code, exactement ce qu'il nous faut. La joie était présente lorsque ça a marché pour la première fois, mon personnage bougeait en temps réel chez mon colocataire Baptiste ; dont le personnage (un carré) bougeait en temps réel chez moi et vice-versa !

Avant d'arriver à cela, le debug était pénible. Unity ne permet de lancer 2 instances du jeu dans l'éditeur... Solution : faire un build à chaque modification pour lancer le .exe. Heureusement, beaucoup de choses se simplifient grâce à l'absence de sécurité. Aucune vérification de l'identité du joueur, pas de vérification de la validité des données reçues, le paradis des tricheurs ! On part du principe que les codes de parties ne sont partagés qu'entre amis, et que les amis ça se fait confiance.

Cependant, héberger et rejoindre n'est que le début du projet, la plupart du travail sur le multijoueur sera sûrement de gérer tous les bugs et les cas particuliers. Ces problèmes ont déjà été imaginés dès le début de notre aventure, c'est pourquoi il nous semblait important que le système du multijoueur soit implémenté en amont de tout le reste afin que les fonctionnalités se construisent dessus et non l'inverse. Pour pouvoir tester j'ai implémenté 2 simples scripts de mouvement avec les animations non correctement intégrées et j'ai eu un aperçu de la difficulté...

4.5 Console

Nous utilisons un Asset du Unity Store pour intégrer une console dans le jeu, que l'on peut ouvrir avec la touche entrée. J'ai toujours trouvé les codes de triches ennuyants à retenir alors j'ai alors proposé de rajouter des commandes pour tester des fonctionnalités. De cette manière, nous avons les commandes `give_item`, `use_consumable` et `set_stat`. À l'avenir peut-être une commande de téléportation ou des façons de débloquent des zones.

4.6 Prochaine soutenance

Pour la prochaine soutenance je prévois de finaliser l'hébergement du multijoueur et d'aider sur la synchronisation des combats. Ayant déjà fait des recherches pour l'apparition des ennemis et leur IA, je vais commencer à les coder avec pour tâche la plus dure de faire A* sur une tilemap. Pour l'inventaire j'aimerais bien presque le terminer. Et finalement, pour le site web, je voudrais finir la page d'accueil et améliorer le style global du site.

5 Anouar

Pour cette première soutenance j'ai dû m'occuper de réaliser le système de déplacement du joueur en plus de devoir implémenter ses différentes animations et je me suis également occupé de l'interface en jeu que le joueur verra durant ses sessions de jeu. Lorsque j'ai commencé à travailler sur ces tâches j'étais très loin de m'imaginer le nombre de difficultés que j'allais rencontrer.

5.1 Gestion des déplacements

Tout d'abord la tâche la plus importante que j'avais à réaliser pour cette soutenance était le fait d'avoir un personnage qui peut se déplacer. Ma première difficulté dans la réalisation de cette tâche était que je ne comprenais pas vraiment ce qui était attendu de moi. En effet, mon objectif était de respecter autant que possible la vision de notre game designer afin d'obtenir un résultat qui serait satisfaisant à ses yeux. J'ai donc commencé à travailler sur un système de déplacement au clavier qui marchait relativement bien et que j'avais commencé à tester sur une scène en vue top down (car à ce moment nous n'avions pas encore choisi de faire de l'isométrique).

5.1.1 Première difficulté

Tout allait basculer lorsque nous sommes passés sur de l'isométrique car dans cette perspective, le choix d'un déplacement à la souris a supplanté au choix initial de déplacement au clavier. Par conséquent à ce stade tout était à refaire pour moi. Je me suis mit à regarder de nombreux tutoriels sur internet afin de trouver une manière de mettre en place un premier déplacement à la souris. Une des première version que j'ai obtenu consistait simplement à déplacer le personnage à l'endroit où l'on cliquait avec la souris, cependant cette solution était loin d'être satisfaisante car elle ne tenant absolument pas compte des spécificité de l'environnement sur lequel le personnage évoluait. J'ai donc continué mes recherches et j'ai eu l'idée de mettre en place un système de pathfinding pour que le personnage s'adapte dynamiquement à son environnement. Mon calvaire a commencé précisément lorsqu'il s'est agi pour moi d'allier pathfinding et vue isométrique. En effet, les solutions proposé par unity comme le système de NavMesh n'était tout simplement absolument pas exploitable sur des tilemap de type isométrique ce que je n'ai pu accepter qu'après de très nombreux essais infructueux. Le navmesh ne prenait tout simplement pas en compte la manière dont les tiles étaient placés et ne faisait même pas la distinction entre zone praticable et impraticable.

5.1.2 Le renouveau

Je devais me rendre à l'évidence, il fallait que j'applique une autre méthode pour parvenir à mes fins. J'ai envisagé la mise en place d'un algorithme de recherche de chemin et j'ai commencé à faire des tests avec l'algorithme A star mais je ne suis pas parvenu à un résultat satisfaisant. Le problème était que le personnage ne voulait pas se déplacer en diagonale car le théorème de Pythagore implique que la distance en diagonale est toujours inférieure à celle d'un des côtés de l'angle droit et malgré toutes les formules de calcul de distance que j'ai tenté d'appliquer je ne suis pas parvenu à outrepasser cette difficulté. Après des semaines passés à m'acharner sur ce problème, j'avais atteint un stade où je ne prenais plus aucun plaisir à travailler sur le jeu et le burn-out me tendait les bras. Je suis allé jusqu'à proposer à l'équipe de revenir à un jeu en vue top down ce qui pour moi non seulement nous aurait simplifié la tâche pour les aspects technique mais cela nous aurait permis également d'améliorer (toujours selon moi) l'aspect graphique du projet. Cependant, au vu de l'état d'avancement du projet, revenir sur du top down aurait impliqué de devoir modifier de nombreux éléments ce qui nous aurait fait perdre bien trop de temps. J'ai donc dû me résoudre à continuer sur cette tâche en l'état. Je me suis alors posé une simple question qui n'avais l'air de rien mais qui pourtant allait tout changer. Quel jeu en vue isométrique purement en deux dimensions propose un tel système de

déplacement. Autrement dit : De quel jeu pourrais-je m'inspirer pour le système de déplacement. Face à l'incapacité de tout le groupe à citer un jeu correspondant à ces critères l'on s'est rendu à l'évidence. Le système que l'on me demandait d'implémenter était bien trop ambitieux pour un projet de cette envergure et en me rappelant du précepte « Vous n'allez pas révolutionner le monde du jeu vidéo avec ce projet de S2 » réaliser toute l'équipe s'est mise d'accord pour revenir sur un déplacement au clavier afin de me permettre tout de même d'apporter le système de déplacement qu'il m'incombait de réaliser.

5.2 Déplacement au clavier

Revenir sur un déplacement au clavier avait des airs de défaite pour moi d'autant plus en tenant compte de tous les temps et l'énergie investie dans le but d'aboutir au déplacement à la souris. Mais lorsque j'ai commencé à travailler sur les déplacements au clavier, le plaisir de travailler sur le jeu commençait peu à peu à me revenir et j'ai abouti à un système qui permet le déplacement du personnage dans huit directions. La difficulté pour ce système fut de faire en sorte que le personnage se maintienne en diagonale lorsque l'on se déplaçait en maintenant une des touches « haut ou bas » en plus d'une des touches « gauche ou droit ». En effet, une fois le déplacement terminé, une de touches était nécessairement relâchée avant l'autre ce qui faisait que l'on finissait par rester dans la position de la dernière touche relâchée. J'ai donc ajouté artificiellement un temps de latence durant lequel si même si l'on ne relâche pas les touches exactement en même temps, le personnage reste considéré comme étant en diagonale.

```
3 références
private IEnumerator StayDiag()
{
    if(_diag)
    {
        yield return new WaitForSeconds(0.05f);
        _diag = false;
    }
}
```

FIGURE 17 – Script de mouvement en diagonale

5.3 Implémentation des animation

Il me restait une fois mon script de déplacement terminé à implémenter les animations du personnage dans les huit directions respectivement : haut, haut droite, droite, bas droite, bas, bas gauche, gauche et haut gauche. Pour ce faire j'ai d'abord songé à gérer manuellement chaque cas dans mon script de déplacement à l'aide de la légendaire succession de « if else ». Mais lorsque j'eus fini, en contemplant mon code de plusieurs dizaines de ligne absolument indigeste je me suis demandé si une solution plus élégante n'existait pas car le déplacement était tout de même une des bases du jeu vidéo. J'ai donc entrepris des investigations en quête d'un système mis en place par unity pour simplifier la mise en place d'un tel système de déplacement. J'ai fini par trouver mon bonheur grâce au système de « blend-tree » disponible quand on crée un objet animator sur unity. Mais avant d'avoir accès à ce merveilleux outil, je

devais d'abord implémenter une par une les différentes animation créée par notre talentueux directeur artistique.

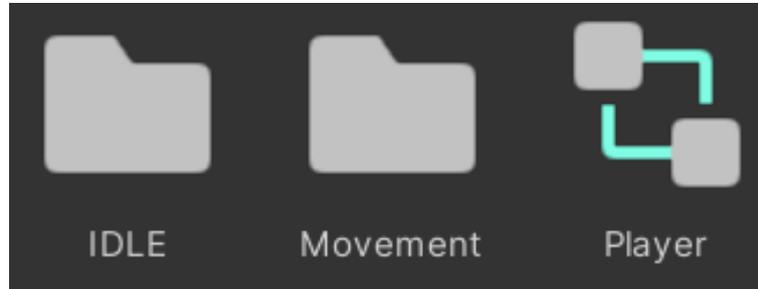


FIGURE 18 – Exemple implémentation d'une animation



FIGURE 19 – BlendTree

Une fois les différentes animations toute créée j'ai pu m'occuper de la manière dont elles s'articuleraient entre-elles au cours du jeu. Je devais m'occuper de deux questions majeures : comment passer d'une animation à l'autre et comment gérer l'arrêt du personnage une fois qu'il aura fini de se déplacer. Ces deux points peuvent être réglés à l'aide des blendtree

Pour le premier point j'ai utilisé deux variables « Movement X » et « Movement Y » qui à l'aide de l'appel à `input.GetAxis("Horizontal")` et `input.GetAxis("Vertical")` pour récupérer en temps réel si les touches qui permettent de se déplacer sur l'axe des abscisses et sur l'axe des ordonnées. Quant au deuxième point je l'ai réglé en ajoutant des animations dite « Idle » et en utilisant une variable booléenne « isMoving » qui permet de passer des animations de déplacement à l'arrêt de personnage dans la bonne direction.

5.4 Gestion de l'interface

Pour conclure j'ai également eu à m'occuper de l'interface que j'ai implémenté de la même manière que celle détaillé dans le cahier des charges. J'ai utilisé un canevas pour agencer les différents éléments de l'interface notamment la barre de vie que j'ai subdivisé en trois sous-sections ainsi qu'une barre de mana qui pour l'instant n'est pas utilisé car les sorts ne sont pas encore présents. De plus je n'ai pas encore ajouté l'inventaire de sort car ce dernier est aussi dépendant de l'ajout des sorts. Finalement j'ai ajouté le fait que l'appui sur une touche inflige des dégâts au personnage et que lorsque celui-ci n'a plus de vie, il disparaît.

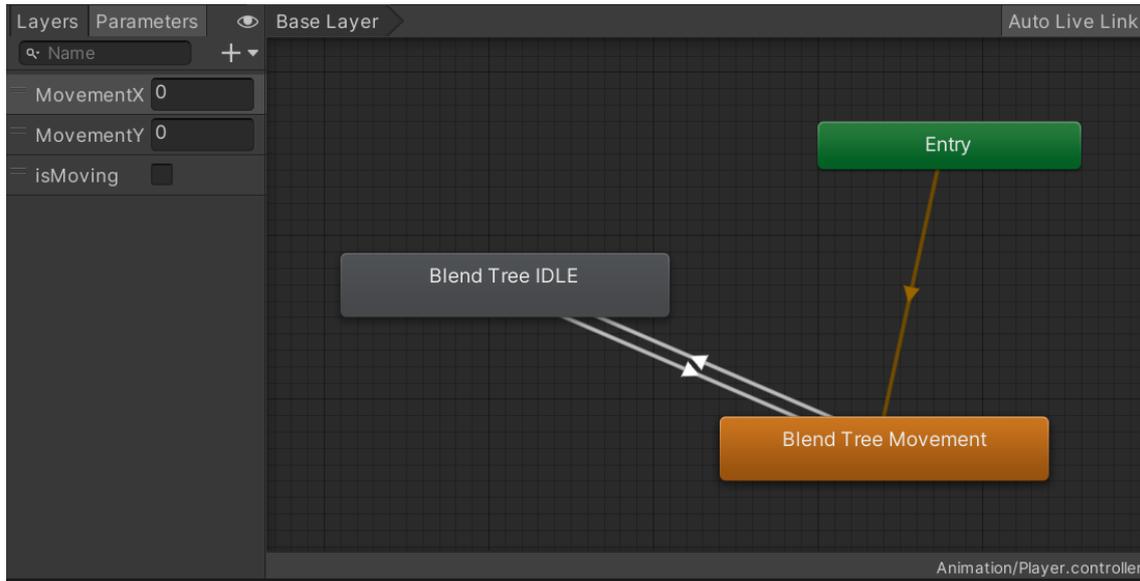


FIGURE 20 – BlendTree

5.5 Prochaine soutenance

Pour la prochaine soutenance, je vais continuer à améliorer le déplacement du personnage et je compte avancer sur l'interface. Ce qui me motive le plus sur ce projet sera d'implémenter un écran d'accueil et j'aurais également à m'occuper de l'implémentation des sorts.

6 Conclusion

Le jeu Hermita est un projet de développement de jeu vidéo ambitieux qui vise à créer un monde unique pour les joueurs. Notre objectif est de créer un jeu qui soit accessible à tous et qui proposera une expérience captivante. Nous avons travaillé dur pour créer un univers unique et cohérent pour Hermita, axé sur la dualité entre la magie naturelle de ce monde et l'aspect artificiel des machines amenées par le joueur.

Pour la prochaine soutenance nous aurons à cœur de maintenir le cap afin d'aboutir à un jeu à la hauteur de nos ambitions.

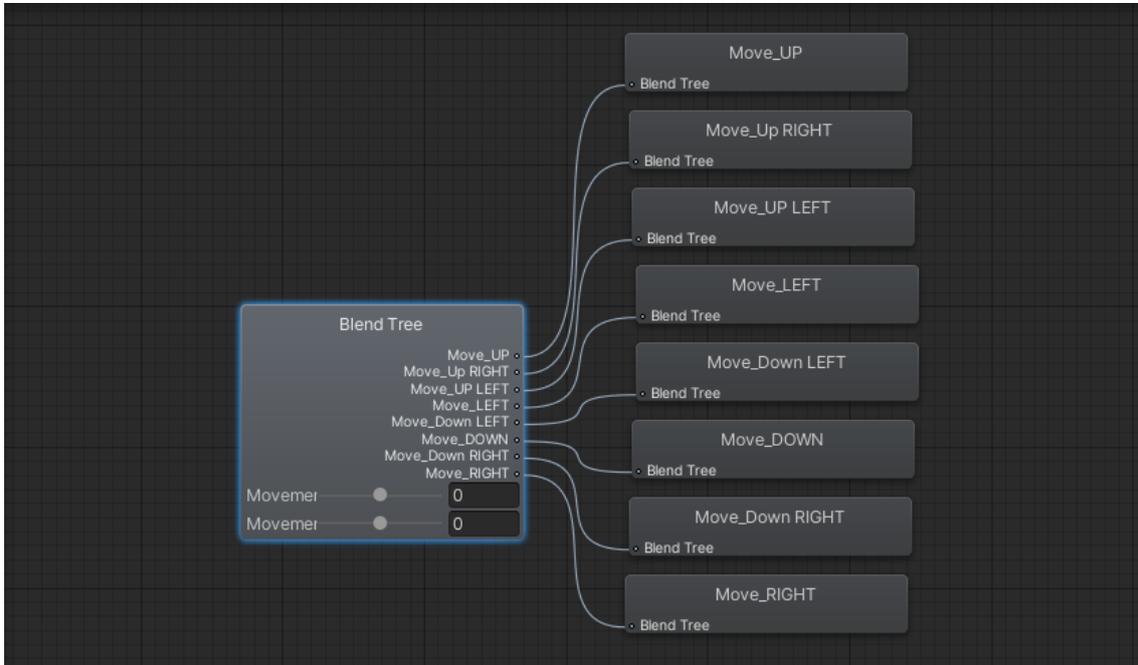


FIGURE 21 – BlendTree

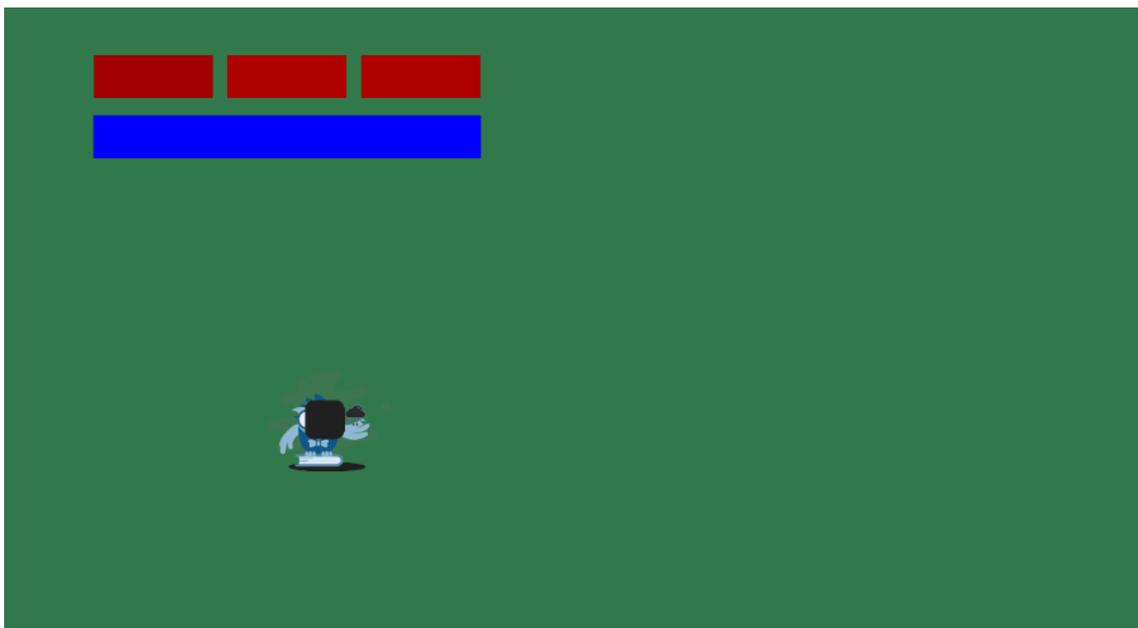


FIGURE 22 – BlendTree