

Rapport de projet - Gumber



Groupe : Tox

Tom COULLIAUD-MAISONNEUVE (Chef de projet)
Nicolas BEC
Elliott BEGON
Hugo MOUTINHO

Juin 2022

Table des matières

1 Reprise du cahier des charges	2
1.1 Structure	2
1.1.1 Fonctionnel	2
1.1.2 Technologique	3
1.1.3 Méthodologique	4
1.1.4 Opérationnel	4
1.2 Planning	5
2 Présentation chronologique individuelle	6
2.1 Tom	6
2.2 Elliott	22
2.3 Hugo	34
2.4 Nicolas	43
3 Conclusion	69

1 Reprise du cahier des charges

1.1 Structure

1.1.1 Fonctionnel

-Déplacement libre de base en deux dimensions avec une vue en plongée sur toute la carte, seul ou avec des compagnons au clavier, des moyens de déplacement plus rapides pourront être rajoutés tels que le déplacement sur une monture ou bien de la téléportation longue distance (refaire la carte entière dans le chemin inverse peut vite devenir assez redondant).

-Un système de classes avec chacune des aptitudes et styles de combats différents sera disponible, l'objectif est d'apporter une diversité dans le style des personnages et dans le gameplay pour ne pas toujours se retrouver face aux mêmes adversaires durant les phases de PvP (player versus player).

-Un système de sorts et aptitudes spéciales sera également disponible et déblocable suivant la progression du joueur et la classe que celui-ci aura choisie. Certains sorts seront à distance ou au corps à corps également suivant le style de jeu et l'arme du joueur.

-Le site web contiendra une page détaillant l'histoire du monde et éventuellement un codex des différents personnages, pnj, monstres avec des détails dessus

-Mode multijoueur (en ligne) :

Il y aura la possibilité de jouer à plusieurs pour affronter entre amis des monstres plus puissants ou simplement pour passer un bon moment ensemble à explorer la carte et ses possibilités.

Participer à des donjons en équipe, établir une stratégie afin de parvenir à triompher à plusieurs contre les nombreux adversaires qui se mettront sur leur route.

Les combats ne seront pas seulement possibles contre les monstres mais également contre d'autres joueurs lors de duels et à plus grande échelle avec des affrontements entre équipe.

Les combats seront au tour par tour sur une zone qui définira la portée des attaques ou la distance à combler si les attaques se font au corps-à-corps.

-Les différentes récompenses et objets seront obtenables durant toute l'aventure du joueur, générés aléatoirement lors de la fin d'une quête, d'un combat ou bien d'un donjon. Ils seront donnés selon les caractéristiques actuelles du joueur, pour qu'il puisse continuer son aventure sans problème et

qu'il n'y est pas d'incohérence de niveau.

-Une interface sera aussi visible en bas de l'écran regroupant toutes les caractéristiques du joueur comme notamment ses points d'expérience, ses sorts ainsi que ses aptitudes, les différentes zones et donjons du jeu, et même voir la liste de ses amis.

1.1.2 Technologique

Nous manipulerons, comme conseillé lors de la remise du cahier des charges, le moteur Unity. De plus, nous utiliserons Visual Studio Code pour lier la partie Script en C# aux différents assets de ce moteur. Concernant le site web, nous partirons sur du HTML et du CSS, nous voulions le programmer nous-même afin de pouvoir donner au site une forme qui nous correspond vraiment. Pour la gestion du projet en lui-même, nous utiliserons Git qui permettra de travailler chacun de son côté et de pouvoir s'organiser très facilement.

Après quelques recherches, nous pensons utiliser Unity 3D pour créer le launcher et de pouvoir télécharger, lancer ainsi que désinstaller Gumber en toute simplicité. Nous utiliserons dans un premier temps patchkit.net, un site qui facilite grandement la mise en place d'un launcher, bien que très rudimentaire. Nous prévoyons donc d'utiliser cette première solution au moins jusqu'à la deuxième soutenance et nous verrons par la suite, si le temps nous le permet, si nous faisons une version plus propre du launcher à l'aide d'un autre outil que nous choisirons au moment venu.

Nous allons utiliser Mirror afin de mettre en place la partie multijoueur, en hébergeant le serveur sur l'ordinateur d'un joueur.

1.1.3 Méthodologique

Pour ce qui est de l'aspect méthodologique, nous utiliserons GitHub afin de pouvoir accéder rapidement aux modifications de chacun et pour gérer facilement l'avancement du projet.

Nous utiliserons Discord comme messagerie instantanée pour communiquer le plus simplement possible lorsque nous ne nous voyons pas en présentiel.

Nous envoyons régulièrement des messages sur cette plateforme pour informer le groupe de notre avancée, sur ce qu'on fait ou encore apporter des modifications sur des parties déjà existantes.

1.1.4 Opérationnel

On pense utiliser des assets libres de droits pour notre jeu car il est très facile d'en trouver actuellement.

Au niveau du site web, nous achèterons l'hébergement de ce dernier et son nom de domaine.

Enfin, concernant le serveur, il sera hébergé directement sur l'ordinateur d'un des joueurs.

1.2 Planning

Concernant l'avancement du jeu :

L'objectif, au niveau de la programmation du jeu, sera d'avancer de façon qu'à chaque soutenance, le projet soit environ un tiers plus avancé. C'est le cas également de la partie chara design.

Certaines tâches avanceront différemment suivant les soutenances mais le tout est, selon nous, assez cohérent.

Notre but est globalement de pouvoir avancer en travaillant régulièrement de façon à atteindre les objectifs fixés suivant le planning.

Tâches	1re	2e	Finale
Programmation du jeu	30%	65%	100%
Chara design	33%	66%	100%
Level design	15%	50%	100%
Sound design / VFX	0%	40%	100%
Site web	50%	90%	100%
Launcher	0%	75%	100%
Conception IA monstres	0%	60%	100%
Réseau	0%	50%	100%
Lore / storyelling	80%	90%	100%

2 Présentation chronologique individuelle

2.1 Tom

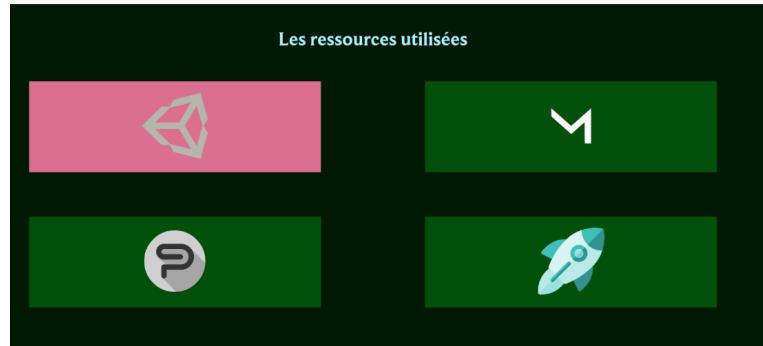
Je suis donc en charge de la partie réseau ainsi que du développement du site web et comme map mentionné plus tôt, nous avions initialement prévu d'avancer dès la première soutenance l'aspect multijoueur du jeu.

Cependant, nous étions des débutants en la matière et rien que le développement du mode solo n'avancait pas forcément aussi vite qu'escompter. De ce fait, j'ai décidé de commencer le site dès le début afin de pouvoir me concentrer pleinement sur la mise en place du multijoueur dès que possible.

Il a donc fallu me replonger dans la programmation hmtl et css pour laquelle j'avais de maigres bases bien insuffisantes pour développer un site digne de ce nom. J'ai donc suivi un cours openclassroom en ligne afin d'avoir les connaissances nécessaires pour commencer à créer le site et quand cela fut terminé, j'ai enfin pu me lancer dans le développement du site en lui-même. J'ai débuté en essayant de faire la page d'accueil mais je me suis rapidement rendu compte que se lancer dans le grand bain sans réfléchir au préalable allait s'avérer très compliqué surtout pour un débutant et constituerait une perte de temps énorme. J'ai donc arrêté le développement de la page d'accueil et j'ai tout de suite créé les fichiers correspondant à l'ensemble des pages que je comptais inclure dans le site : il y avait donc à l'origine la page d'accueil, une page pour présenter le groupe et une page contenant les documents demandés (manuel d'installation, d'utilisation et rapport) ainsi que le lien de téléchargement du jeu et enfin une dernière page qui aurait dû être une page sur laquelle on pouvait se connecter et avoir accès aux informations de son personnage, de son compte, des autres joueurs etc. Cette page a néanmoins été abandonnée pour des raisons que je détaillerai plus tard. D'un autre côté en reprenant la création de la page d'accueil, je me suis dit qu'il valait peut-être mieux de créer une page à part qui détaillerai les grandes étapes du développement du jeu, au lieu de l'intégrer à la page d'accueil et pour éviter ainsi de la surcharger. Après avoir créé tous les fichiers correspondant aux pages html ainsi qu'à leur page css associées, je me suis lancé dans la conception de ce qu'on pourrait appeler un menu de navigation qui est en fait une bande fixée en haut de la page qui nous donne accès aux autres pages du site. J'ai donc pour cela utilisé une balise nav couplée à une liste à puces

pour créer la liste des liens, mais lorsque je suis passé à la mise en page sur css, la manipulation de la liste à puces me posait problème et j'ai donc tout simplement utilisé des balises les unes à la suite des autres, ce qui m'a semblé plus simple à manipuler.

Un autre problème rencontré lors de la mise en place de ce nav a été, lors de la mise en forme des liens, de retirer le changement de couleur et de forme des liens une fois cliqués. Je suis allé chercher me documenter pour savoir comment faire et cette modification ne me paraissait pas difficile, mais lorsque j'ai voulu tester après avoir changé les propriétés des liens après avoir été cliqués, aucun changements n'apparaissaient. J'ai alors essayé diverses choses sans grand succès et j'ai enfin réussi en définissant les propriétés que je souhaitais pour tous les états des liens, ce qui heureusement me convenait du point de vue visuel. La suite du développement de la page s'est ensuite déroulée sans grand soucis jusqu'à ce que j'arrive à la partie des liens vers les ressources que nous utilisons pour le projet. Je voulais en effet avoir des cases (en l'occurrence 4) avec les logos des ressources et en faire des liens vers le site de ces logiciels et outils. Comme je savais que nous en aurions 4, j'ai décidé d'en placer deux côte à côte sur deux lignes et pour cela je leur ai attribué un identifiant commun en fonction du côté sur lequel ils étaient situés, ce qui m'a causé par la suite de nombreux problèmes lors de l'agencement des blocs. J'ai donc tenté une autre approche en les regroupant par lignes, ce qui a tout de suite réglé les soucis que je rencontrais, mais j'ai néanmoins dû renoncer à faire changer de taille les blocs lorsque on place notre curseur dessus, car cela détruisait complètement la mise en page à cause du bloc du dessous qui se déplaçait pour "laisser la place" au bloc du dessus, problème que je n'ai pas réussi à régler même en plaçant un peu plus bas les cases du bas, pour au final obtenir le résultat suivant :



La dernière grosse étape de cette première page a été de créer le pied de page, second élément commun à toutes les pages, après le menu de navigation.

Cet élément n'a pas été très compliqué à créer en soi, mais a causé lui aussi son lot de problèmes : tout d'abord, il ne prenait pas toute la largeur, peu importe les efforts que je faisais pour faire en sorte qu'il remplisse bien la page, ce qui m'a pris un temps non négligeable à résoudre (on notera par ailleurs que c'est un souci que j'ai rencontré plusieurs fois, ainsi quand j'ai trouvé la solution pour le pied de page, plusieurs problèmes ont été réglés d'un coup). Il se trouve en fait que par défaut le body ne prend pas toute la largeur de la page car la valeur prédéfinie pour la propriété "margin" , à savoir la marge sur les côtés, n'est pas nulle. De plus, le second problème est que le footer refuse de s'afficher bien tout en bas de la page, ce qui me force à le décaler manuellement vers le bas à l'aide de la propriété "top". Je n'ai par ailleurs pas compris la raison exacte de pourquoi il ne se place pas par défaut en dessous des autres éléments pourtant placés au-dessus dans le code de la page html, mais je pense qu'il est possible que ce soit à cause du fait que ces blocs sont eux aussi décalés, ou alors parce que je n'ai pas défini de taille pour ces éléments et je laisse le site calculer cela tout seul (j'ai essayé de modifier la taille des blocs au-dessus et du body et il y a bien eu quelques changements, mais m'ai permis de tirer des conclusions dont je suis absolument sûr). Après le long périple que fut la création de la première page, qui fut d'ailleurs ma première expérience avec le html et css depuis longtemps, je me suis lancé dans le développement de la page du groupe. Là rien de bien compliqué et pratiquement aucun problème rencontré, si ce n'est le body qui ne prenait pas toute la largeur et que j'ai réglé quand j'ai trouvé la solution pour le footer de la première page, cependant, c'est en développant cette page que je me suis rendu compte d'une chose terrible : je n'ai aucun sens des couleurs et à cause de ça le rendu du site était grandement impacté. J'étais à la base

partie sur des jaunes et vert assez clair pour être dans la même gamme de couleur que le logo, mais le visuel était vraiment mauvais. J'ai donc demandé de l'aide à un ami, et après une ou deux heures de tests et de dur labeur, on en est enfin arrivé au stade où j'étais fier de présenter un beau site, et pas un site qui donne envie de se rincer les yeux à la javel après deux minutes passées dessus. Enfin, la dernière chose dont je me suis occupé est la page qui servira à décrire l'avancée étape par étape du projet. J'avais pour celle-ci envie de faire une frise chronologique verticale, et après avoir un peu réfléchi à comment je pourrai implémenter ça, je me suis rendu compte que cela était au-dessus de ce que j'étais capable de faire à l'heure actuelle, ou du moins dans un intervalle de temps raisonnable (cela aurait peut-être et seulement peut-être faisable, mais aurait été une perte de temps énorme au point que la page ne serait pas terminée à ce jour).

J'ai donc fait mes recherches et j'ai trouvé des exemples de templates pour des frises verticales. J'en ai alors sélectionné une que j'ai par la suite modifiée (taille, couleur et positionnement) pour la faire correspondre à ce que je souhaitais. J'ai par ailleurs trouvé très intéressant de prendre du code fait par quelqu'un d'autre pour essayer de m'y plonger et réussir à le modifier à ma guise, bien que je n'aie pas non plus décortiqué ligne par ligne le contenu du code que j'avais trouvé. Je voudrais maintenant revenir sur pourquoi j'ai abandonné la page sur laquelle on avait prévu de faire se connecter les utilisateurs pour leur afficher leurs données de jeu. Il se trouve en fait qu'en me renseignant sur comment faire le système de connexion, j'ai découvert qu'il m'aurait fallu apprendre à programmer en php ainsi qu'à lier le site à une base de données stockant les informations des utilisateurs, et cela se serait avéré trop chronophage, surtout étant donné que cela m'aurait imposé d'utiliser la même base de données lors de l'implémentation future du mode multijoueur du jeu. Je suis donc allé voir comment faire et en toute honnêteté, cela m'est apparu hors de portée car beaucoup trop complexe. D'un autre côté, j'ai entendu parler d'une API pour unity appelée Mirror qui permettrait de simplifier les échanges clients serveur, en les réunissant en un seul projet.

J'avais donc un choix à faire, utiliser une base de données pour le multijoueur, ce qui me donnait la possibilité de faire la page que nous souhaitions originellement faire, ce qui avait de grande chance de s'avérer trop difficile pour moi et de demander un temps supérieur à celui que je possède, ou de me

tourner vers mirror qui me semble quant à lui bien plus simple d'utilisation et augmentant grandement mes chances de réussir à produire un travail à la hauteur de ce que j'espérais lorsque nous nous sommes décidés sur le choix du sujet. C'est donc la seconde option qui m'a semblé convenir le mieux à la situation, d'autant plus que je devais attendre que le développement du jeu en lui-même soit assez avancé pour me permettre de me lancer dans l'implémentation du multijoueur.

Après la première soutenance, j'ai donc repris mon travail et ai effectué plusieurs tâches diverses.

Tout d'abord, pour ce qui est du site internet, j'ai continué d'avancer dessus et il est désormais pratiquement terminé, ce qu'il reste à faire n'est en fait que quelques détails et surtout cela ne touche plus vraiment au code ou très peu seulement. J'ai en fait ajouté 2 pages supplémentaires au site internet : la première est celle sur laquelle on peut retrouver tous les documents importants et demandés dans le dossier du projet. Elle ne m'a pas posé énormément de problèmes et était relativement facile à faire mais je trouve quand même pertinent de relever que si elle fut si facile à faire, c'est grâce aux techniques utilisées dans la template que j'avais trouvé pour faire la frise chronologique dans la soutenance précédente. En effet, j'ai réutilisé l'idée d'utiliser des blocs `div` englobant simplement pour positionner les éléments d'une certaine manière et en créant ce qu'ils appelaient des « containers ».

```
<div id = "dldoc">
    <h3>Autres documents</h3>
    <p>Retrouvez ici les documents concernant le projet</p>
    <div id="container"><a href="https://youtu.be/dQw4w9WgXcQ" target="_blank" id="link">Manuel d'installation</a></div>
    <div id="container"><a href="https://youtu.be/dQw4w9WgXcQ" target="_blank" id="link">Manuel d'utilisation</a></div>
    <div id="container"><a href="https://youtu.be/dQw4w9WgXcQ" target="_blank" id="link">Rapport du projet</a></div>
</div>
```

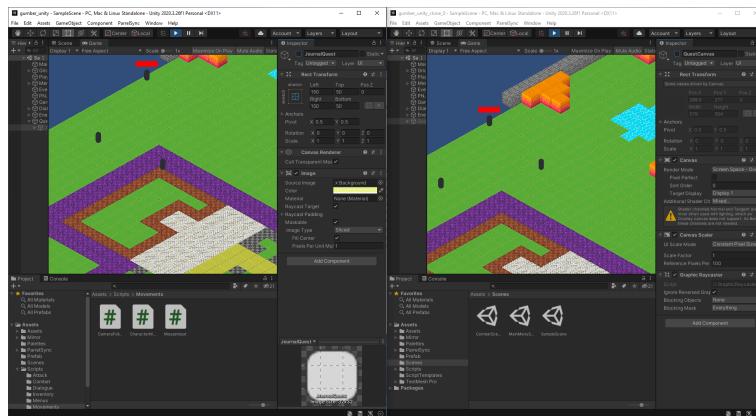
Comme on peut le voir dans la capture d'écran, j'ai utilisé des `div` supplémentaire avec l'id container pour pouvoir faciliter le placement de mes liens, ce qui m'a permis d'arriver très facilement au résultat ci-dessous, chose qui m'aurait probablement pris beaucoup plus de temps auparavant.



J'ai ensuite ajouté une seconde page au site, qui est une page que l'on a décidé d'ajouter pour donner un peu plus de vie au site, puisque on peut trouver dessus dans les grandes lignes l'histoire de l'univers de Gumber écrite par Eliott ainsi qu'un premier aperçu des classes jouables dans le jeu. Elle n'a pas non plus été très compliqué à faire puisqu'elle ne contient qu'un peu de textes et quelques images, mais il nous tenait à cœur de faire apparaître dans le site une page en plus de ce qui était explicitement demandé et qui venait ajouter un peu à l'immersion dans l'univers de Gumber.

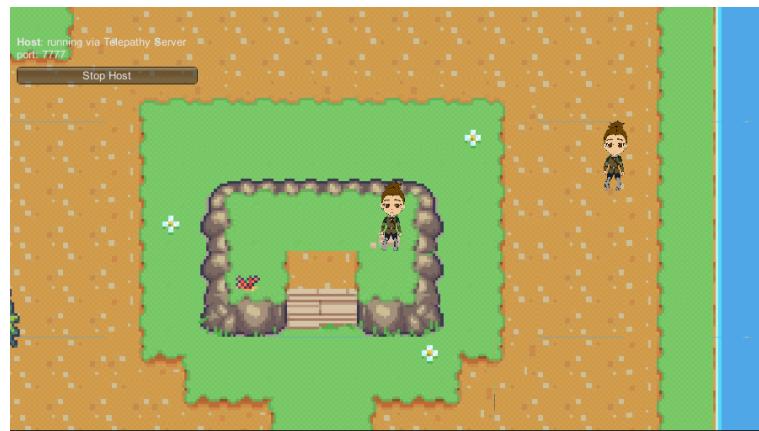
On arrive maintenant à la partie multi-joueurs qui pour le coup avait été totalement délaissée avant la première soutenance car on a jugé qu'il valait mieux attendre d'avoir un peu plus de contenu au niveau du jeu en solo avant de l'adapter pour le multi-joueurs. Je suis donc parti de zéro mis à part que je savais quelle API utiliser pour m'aider à implémenter le multi-joueurs, à savoir Mirror. Un avantage dans le choix de cette API est que sur le site internet où on peut la trouver, il y a aussi une bonne quantité de documentation, ce qui m'a permis de comprendre un peu les grandes lignes du fonctionnement de Mirror avant de me lancer dans le grand bain. Ils mettent aussi à disposition des liens vers des tutos sur youtube qui détaillent un peu l'utilisation des différents composants et l'un d'eux m'a permis de commencer à mettre en place le réseau dans le projet.

Un autre outil qui m'a été très utile et qui a même servi aux autres membres du groupe est une extension pour unity appelée « ParrelSync ». L'utilité de ce package est de pouvoir créer des clones du projet sur lequel on est en train de travailler et de pouvoir les ouvrir dans un nouvel éditeur unity, ce qui est très pratique pour tester le multi-joueurs sur un seul ordinateur puisque sans ça, j'aurai été obligé de build le projet à chaque fois que je voulais tester ce que j'avais fait, ce qui aurait constitué une perte de temps monstrueuse, alors que grâce à cet outil, lorsqu'une modification est faite sur l'éditeur du projet principal, elle se met automatiquement à jour sur les autres et il n'y a plus qu'à lancer pour tester. Voici un exemple de comment se présente ParrelSync.



On va maintenant pouvoir rentrer dans ce que j'ai fait pour pouvoir mettre en place le multi-joueurs. Donc tout d'abord, il faut créer un nouveau gameObject et lui ajouter le composant network manager mis à disposition par Mirror, c'est en fait cet objet qui va gérer tout ce qui touche au multi-joueurs et notamment linstanciation des différents joueurs car avec Mirror, lorsqu'un client se connecte au serveur, le serveur instancie un personnage pour se joueur, il faut donc pour ça lui indiquer quel prefab il doit faire apparaître. Cela m'a amené à faire la première modification sur le projet car à l'origine pour jouer en solo, on avait mis de base un gameObject player dans la scène et il a donc fallut le retirer, ce qui a alors posé d'autres problèmes qu'on verra un peu plus tard. Une autre chose importante à faire est d'ajouter le composant « networkIdentity » au prefab du joueur pour pouvoir linstancier avec le network manager, de plus, le composant « network transform » permet de rendre le déplacement d'un joueur visible par les autres aussi.

Pour revenir maintenant au problème évoqué plus tôt, une fois que tout fut mis en place, on a bien pu voir apparaître et se déplacer les différents joueurs sur la carte, mais il y avait alors 2 soucis à régler : tout d'abord il y avait un personnage en trop qui n'était contrôlé par personne comme je l'ai expliqué plus haut, mais le second problème est que la caméra ne suivait pas notre personnage et restait fixée à ce personnage qui n'était le personnage de personne comme on peut le voir ci-dessous.

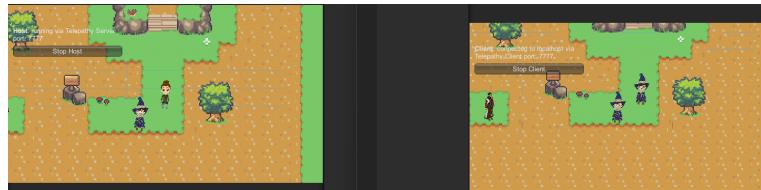


Ici, c'est le personnage de droite qui est celui du joueur et non celui au milieu, pourtant la caméra ne le suit pas.

Pour régler ça, il a déjà fallu retirer ce player de la scène, mais cela a eu pour effet de causer des plantages dans le jeu puisque la caméra ne savait plus quoi suivre et ajouter le prefab player dans l'attribut « target » du script qui gère le suivi de la caméra n'a rien réglé. J'ai alors beaucoup cherché sur internet sans grand succès, car certain conseillaient d'utiliser un package pour la caméra qui s'appelle « cinemachine » et d'autres donnaient des solutions intéressantes mais qui ne s'appliquaient pas pour nous, j'ai donc dû me débrouiller moi-même. Pour cela, j'ai en fait repris certains éléments vu lors de mes recherches en les adaptant à notre cas et cela a fini par fonctionner. J'ai en fait créé une nouvelle méthode « setTarget » dans le script qui gère le suivi de la caméra qui, comme son nom l'indique, donne une valeur à l'attribut target. Grâce à cette méthode, j'ai ajouté quelque ligne dans la méthode « start » du player pour que si l'attribut « isLocalPlayer » qui est donné par Mirror et qui est un booléen, est à True, alors on appelle la méthode

setTarget et on lui donne en paramètre le gameObject qui l'a appelé, pour ainsi se définir comme cible du suivit de la caméra.

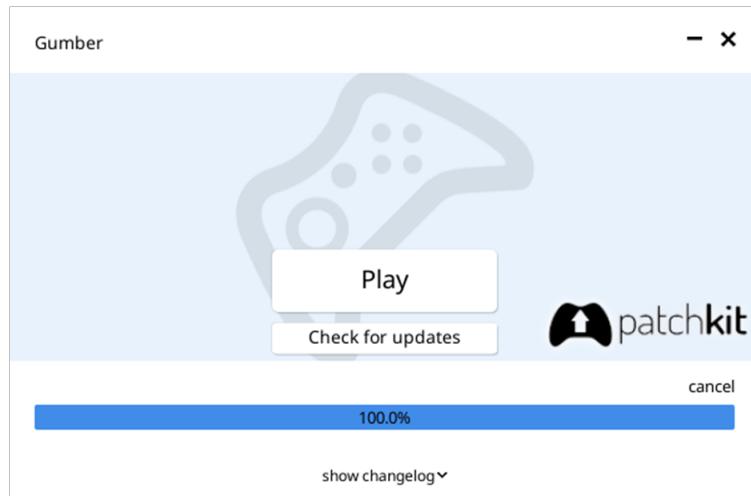
Avec ça, la synchronisation des déplacements entre les joueurs était terminé, ce qui était l'objectif que je m'étais fixé pour cette soutenance, mais on a découvert en testant le multi-joueurs quelques bug qui vont devoir être fixés par la suite comme l'apparence des personnages qui diffère selon qu'on se place du point de vue du serveur ou du client comme on peut le voir ci-dessous, bien que je ne donnerai pas plus de détails puisque à l'heure actuelle, le problème n'est pas résolu et qu'il sera traité lors du rapport de projet.



A gauche, le point de vue du serveur où tout fonctionne bien, et à droite celui du client qui affiche tous les personnages avec le skin de son personnage à lui.

La seconde soutenance était alors passée et on entrait alors dans la toute dernière ligne droite du projet. Le planning était cette fois-ci très simple : il fallait tout finir et rendre un produit terminé ainsi que tout ce qui tourne autour (site, installateur, multi-joueurs en ligne etc). Etant donné que j'étais en charge de tout ce qui touchait au réseau, je me suis fixé comme première tâche de mettre le site en ligne. Après quelque recherche, j'ai trouvé que github offrait la possibilité pour chaque compte de créer un projet spécial qui ferait office de site internet, ce qui m'a paru être une bonne opportunité de par la simplicité qu'elle semblait présenter. Pour cela, il fallait simplement avoir un répertoire git portant le même nom que le pseudo de notre compte, j'ai donc essayé de modifier le répertoire que nous utilisions jusqu'à présent pour le projet, sans grand succès. J'ai par la suite tenté de créer un nouveau projet de zéro pour cette fois-ci tester en suivant de plus près le guide que j'avais vu et cela a fonctionné sans-soucis. Seul bémol, le nom du site n'était pas idéal puisqu'il était identique à mon pseudo et était donc sans grand rapport avec le projet. J'ai donc cherché comment modifier le nom du site et je suis tombé

sur des guides qui m'ont eu l'air d'être plus ou moins compliqués et surtout pas forcément gratuit. C'est pour cela que je me suis contenté de me recréer un compte github avec pour pseudo le nom que je souhaitais donner au site et tout à parfaitement marché, c'était donc une première chose de faite. La deuxième tâche que je me suis donné a été de créer un launcher pour le jeu. J'ai pour cela utilisé patchkit qui est un outil très puissant permettant de créer un launcher pour son jeu unity en donnant par ailleurs la possibilité de faire des mises à jour du jeu sans pour autant imposer au joueur de télécharger à nouveau ce dernier, puisque les mises à jour se font automatiquement au lancement.



Comme on peut le voir ici, on a la possibilité lors du lancement du jeu de vérifier si une mise à jour a été mise en ligne ou même de vérifier l'historique des changements, ce qui est une fonctionnalité qui permet à notre projet de se rapprocher d'un vrai jeu qui est tenu à jour régulièrement. Ce launcher fait donc office de procédure d'installation et je vais en profiter pour expliquer un choix que nous avons fait par la suite. Comme je l'ai dit, ce launcher fait office de procédure d'installation puisqu'il s'occupe à la place de l'utilisateur d'installer le jeu, cependant il n'offre pas la possibilité de désinstaller le jeu, ce qui ne correspond pas aux attentes formulées dans le dossier du projet. Quand nous nous sommes rendu compte de ce problème, 2 possibilités s'offrent à nous, soit nous arrêtons d'utiliser le launcher pour nous conformer aux consignes, soit nous gardons le launcher car la possibilité de mettre à jour le jeu était un vrai plus, quitte à déroger un peu aux consignes données.

Nous avons donc un peu discuté entre nous et nous avons finalement décidé que la seconde option était la meilleure pour une simple raison : étant donné qu'il est imposé de fournir un manuel d'installation, nous avions la possibilité de prévenir l'utilisateur de bien télécharger le launcher là où ils souhaitent stocker les fichiers du jeu pour pouvoir par la suite le supprimer aisément. En effet, puisque pour supprimer le jeu il suffit de supprimer le dossier dans lequel se trouvent les fichiers, ce qui est une manipulation encore plus rapide que d'utiliser une procédure de désinstallation à condition de savoir où se trouve le dossier contenant le jeu, nous avons jugé que le point positif offert par le launcher était suffisamment important pour compenser ce petit défaut qui n'en est finalement pas un à condition d'être prévenu de ce soucis lors de l'installation du jeu.

Nous passons maintenant à la troisième et avant-dernière tâche que je me suis donnée, qui n'est pas l'une des plus évidente puisqu'elle consiste en la création d'un serveur de jeu permettant de jouer véritablement en multi-joueurs en ligne. Pour cela, je me suis aidé de la plateforme d'amazone AWS (Amazon Web service) qui propose de nombreux services d'hébergement ou autres en ligne.

Parmi tous les services proposés, deux formaient de solides candidats pour permettre à notre jeu d'enfin dépasser les frontières du local : lightsail, un service d'hébergement de serveurs virtuels, et EC2, un service permettant de lancer une machine virtuelle. Je me suis dans un premier temps dirigé vers lightsail puisqu'il semblait répondre plus précisément à nos besoins, mais puisqu'il ne semblait pas y avoir de possibilité pour l'utiliser gratuitement, je suis alors allé voir du côté d'EC2. J'ai alors trouvé un guide en ligne détaillant la mise en place d'une machine virtuelle dont on aurait accès à l'adresse ip. J'en profite maintenant pour préciser que ce qui a été fait sur cette machine virtuelle aurait pu être fait sur l'un de nos ordinateurs, mais il y avait tout de même un problème de taille, celui d'ouvrir nos ports et de récupérer l'adresse ip publique de notre ordinateurs, chose que je n'ai pas réussi à faire même avec quelques recherches et qui n'était pas nécessaire en utilisant EC2, ce qui en a fait au final la parfaite solution à notre problème. Une fois la configuration de la machine virtuelle terminée et le mot de passe récupéré grâce à ma paire de clé, j'ai alors eu accès à un ordinateur virtuel stocké dans un serveur et dont les ports étaient ouverts. Lorsqu'on lance la connexion à cet ordinateur, l'interface se présente comme ceci :



On a donc un ordinateur prêt à l'emploi et auquel on peut se connecter à condition d'ajouter quelques règles supplémentaires dans les paramètres du pare-feu comme on peut le voir ici :

Name	Group	Profile
<input checked="" type="checkbox"/> Mirror Port		All
<input checked="" type="checkbox"/> Program Mirror		All

Ces deux règles visant à ne pas restreindre le port 7777 utilisé par mirror et à autoriser la connexion lors de l'utilisation de notre programme.

Une fois ces étapes préliminaires passées, il ne restait plus qu'à créer un build serveur pour notre jeu, à le transférer sur la machine virtuelle et à le lancer pour que les autres puissent se connecter en utilisant l'ip publique que j'ai pu trouver sur la plateforme Amazon.

Un peu de temps s'était alors écoulé ce qui a laissé aux autres le temps d'avancer sur le cœur du projet à savoir le jeu, j'ai donc pu une fois ces trois tâches terminées me mettre à travailler sur le projet avec les autres et notamment Hugo pour pouvoir ajouter des fonctionnalités multi-joueurs à notre jeu.

J'avais auparavant réussi à synchroniser les déplacements et le suivi de la caméra en multi-joueurs ce qui était un bon début mais était loin d'être suffisant. De plus, le jeu comportait de nombreux bugs liés au multi-joueurs qu'il fallait régler. Tout d'abord, les sprites des personnages autres que le siens n'étaient pas les bons, ils étaient par ailleurs tous les mêmes et en l'occurrence celui du personnage que le joueur choisissait. La cause de ce problème était simple, nous laissions mirror instancier à notre place les joueurs mais les

données utilisées pour linstanciation étaient seulement locales. Pour fixer ce problème il n'y avait pas réellement de solution puisque notre manière de faire était mauvaise, nous avons donc entrepris avec Hugo de faire notre propre système de spawn. Pour cela la première étape était de créer notre propre networkmanager hérité de celui fournit par mirror pour y modifier certaines fonctions et en ajouter d'autres. Nous avons donc réécrit les méthodes OnStartServer et OnClientConnect qui se lancent respectivement au lancement du serveur et à la connexion d'un joueur pour modifier leur comportement. Dans la première, nous avons utilisé le système de message mis en place par mirror en utilisant ce qui est appelé le RegisterHandler et qui sert d'après mon ma compréhension à gérer tous les messages envoyés au serveur pour en faire quelque chose. Nous avons donc fait en sorte que lors de la connexion d'un client, un message soit envoyé au serveur avec les bonnes informations concernant linstanciation du personnage, comme on peut le voir dans le bout de code ci-dessous :

```

public override void OnStartServer()
{
    base.OnStartServer();
    Debug.Log("Server started!");
    NetworkServer.RegisterHandler<CreateMMOCharacterMessage>(CustomInstantiate);
}
public override void OnStopServer()
{
    Debug.Log("Server stopped!");
}

public override void OnClientConnect()
{
    base.OnClientConnect();
    Debug.Log("Connected to server!");
    CreateMMOCharacterMessage characterMessage = new CreateMMOCharacterMessage
    {
        race = CharacterManager.race,
        pseudo = CharacterManager.pseudo
    };
    NetworkClient.Send(characterMessage);
}

```

On remarque par ailleurs que le RegisterHandler fait appel à la méthode customInstantiate qui est une méthode que nous avons ajouté au network manager pour instancier les personnages comme nous le souhaitons. Cette méthode prend en argument un message et une connexion et va en fait simplement instancier le personnage décrit par le message et donner au joueur à qui appartient la connexion le contrôle de ce personnage. On a pour cela créé 5 variantes de notre gameObject player dans lequel on a pré rempli les caractéristiques des différentes classes et il n'y avait alors plus qu'à instancier le bon et à lui attribuer le bon pseudo selon le contenu du message avant de

l'attribuer comme gameController au client décrit par la connexion comme on peut le voir ci-dessous :

```
void CustomInstantiate(NetworkConnectionToClient conn, CreateMMOCharacterMessage message)
{
    (string pseudo, string race) = (message.pseudo, message.race);
    switch(race)
    {
        case "Paladin":
        {
            GameObject Player = Instantiate(Paladin);
            PlayerData data = Player.GetComponent<PlayerData>();
            data.pseudo = pseudo;
            data.race = race;
            NetworkServer.AddPlayerForConnection(conn, Player);
            break;
        }

        case "Support":
        {
            GameObject Player = Instantiate(Support);
            PlayerData data = Player.GetComponent<PlayerData>();
            data.pseudo = pseudo;
            data.race = race;
            NetworkServer.AddPlayerForConnection(conn, Player);
            break;
        }

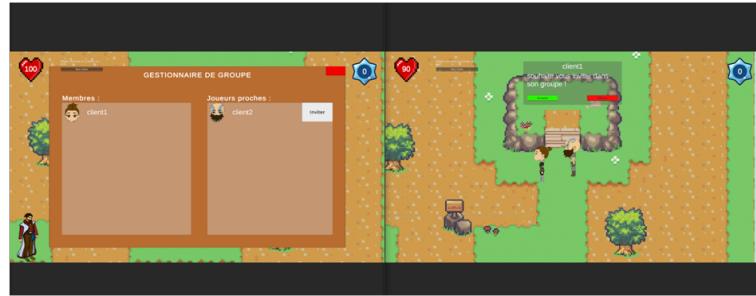
        case "Hunter":
        {
            GameObject Player = Instantiate(Hunter);
            PlayerData data = Player.GetComponent<PlayerData>();
            data.pseudo = pseudo;
            data.race = race;
            NetworkServer.AddPlayerForConnection(conn, Player);
            break;
        }
    }
}
```

Il n'y a par ailleurs en réalité pas tout dans ce bout de code, mais le reste est simplement la même chose appliquée aux deux autres classes jouables du jeu. Grâce à cela, il n'a plus fallu que modifier quelques petites choses mineures pour voir enfin apparaître les bons personnages lors du multi-joueurs, ce qui nous a pris au final beaucoup de temps puisque pour réussir ceci, nous n'avons utilisé aucun guide ou tutoriel car nous n'avions rien trouvé de pertinent sur le sujet à part une vidéo expliquant en détail le fonctionnement de Mirror.

Dans la même continuité, je me suis attaqué avec Hugo à ce qui constituera ma dernière tâche sur ce projet que nous avions alors commencé depuis des mois. Cette dernière tâche était en fait de faire un système de menu de groupe qui permet donc aux joueurs de s'inviter et d'appartenir au même groupe. Je tiens d'ailleurs à préciser que ce système de groupe est probablement ce qui nous a posé (ou en tout cas ce qui m'a posé) le plus de difficulté de tout le projet. L'idée était donc comme dans n'importe quel jeu du genre, d'avoir un menu affichant les joueurs proches et nous donnant la possibilité de les inviter. La première étape a donc été de créer ce menu qui devait afficher la classe et le nom des joueurs proches, ce qui a pu se faire assez facilement grâce à notre système de spawn personnalisé, puisque les gameObject des

joueurs contenaient toutes les informations sur ces derniers. La partie compliquée a été de faire en sorte que lorsqu'on clique sur le bouton inviter, une interface s'ouvre sur le jeu du joueur que l'on souhaite inviter et que lorsqu'il clique sur accepter, les groupes soient actualisés comme on peut le voir ici :

-invitation



-après acceptation



Pour faire ceci, nous avons à nouveau utilisé ce qui est mis à disposition par mirror, à savoir les attributs de méthodes [Command], [ClientRPC] et [TargetRPC] qui ont respectivement pour rôle d'exécuter une méthode sur le serveur et d'exécuter une méthode sur tous les clients ou sur un client en particulier. On a alors créé 4 méthodes : invite, receiveInvitation, AcceptInvitation et AddplayerToGroup dont je vais un peu détailler le fonctionnement. Invite s'exécute sur le serveur et lui demande d'appeler sur le client souhaiter la méthode ReceiveInvitation en lui transmettant nos informations. La méthode ReceiveInvitation ouvre l'interface qui permet d'accepter ou refuser et qui va dans le cas de l'acceptation appeler la commande AcceptInvitation. Cette dernière s'exécute sur le serveur et va faire en sorte d'appeler sur tous

les clients du groupe la méthode Addplayer qui va actualiser le groupe avec le nouveau joueur. Ce qui donne au final ceci :

```
[Command]
public void Invite(NetworkIdentity identity, string pseudo, PlayerData[] playersInGroup)
{
    Debug.Log("Inviting Player...");
    ReceiveInvitation(identity.connectionToClient,pseudo, playersInGroup);
}

[TargetRpc]
public void ReceiveInvitation(NetworkConnection target, string pseudo, PlayerData[] playersInGroup)
{
    Debug.Log("Invitation received");
    GameObject InvitePanel;
    InvitePanel = GameObject.Find("/InviteCanvas/Invitation");
    InvitePanel.transform.position = new Vector3(960,1030,0);
    InvitePanel.transform.Find("Pseudo").GetComponent<TextMeshProUGUI>().text = pseudo;
    GameObject.Find("GroupCanvas").GetComponent<GroupManager>().playersInGroup = playersInGroup;
}

[Command]
public void AcceptInvitation(NetworkIdentity identity, PlayerData newMember)
{
    Debug.Log("Coucou je suis le serveur et j'invite un joueur");
    AddPlayerToGroup(identity.connectionToClient, newMember);
}

[TargetRpc]
public void AddPlayerToGroup(NetworkConnection target, PlayerData newMember)
{
    Debug.Log("Récupération du membre invité...");
    int i = 0;
    PlayerData[] group = GameObject.Find("GroupCanvas").GetComponent<GroupManager>().playersInGroup;
    while (i < 4 && group[i] != null)
    {
        Debug.Log($"Le {i}eme joueur dans le groupe est {group[i].pseudo}");
        i++;
    }
    if (i < 4)
    {
        Debug.Log("i < 4");
        group[i] = newMember;
    }
}
```

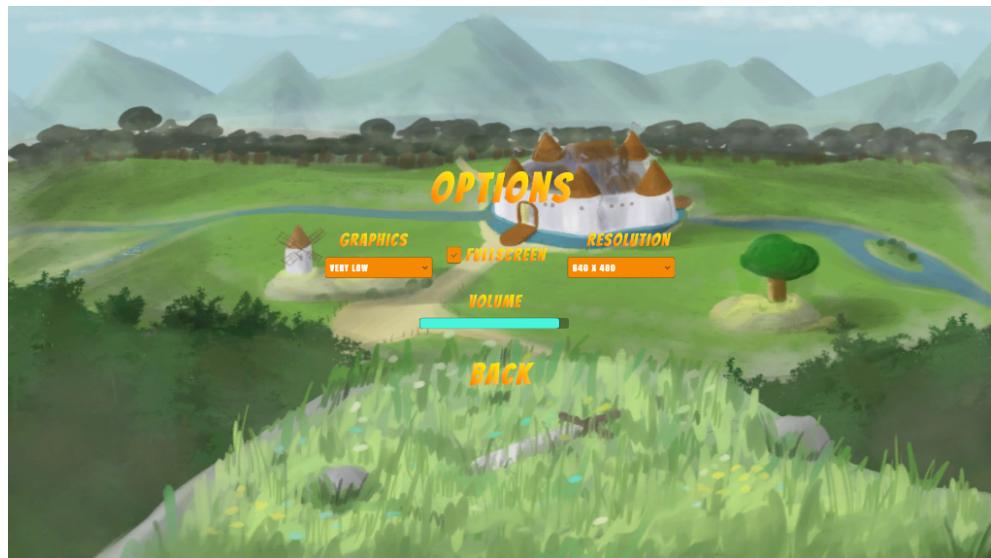
Après tout ça, j'ai finalement aidé Nicolas à synchroniser dans un contexte multi-joueurs les animations de déplacements des joueurs qu'il venait d'implémenter, ce qui conclue finalement mon long parcours sur ce projet de plusieurs mois et qui m'a permis d'apprendre réellement de nombreuses choses et surtout de me permettre de me familiariser avec le fait de faire des recherches sur un sujet à partir de zéro, ce qui je pense sera nécessaire dans la suite de mes études ainsi que dans ma vie professionnelle.

2.2 Eliott

J'ai commencé à travailler sur Gumber en créant un dossier TEST, pour prendre connaissance et me familiariser avec les différents menus et aspects de Unity, tout en regardant une vidéo Youtube explicative pour les débutants. Avec ces connaissances et les connaissances acquises par le premier TP de NTS, j'ai réalisé un croquis d'une carte isométrique afin de créer quelque chose d'assez concret où je pourrai faire mes premières expériences. J'ai ainsi trouvé une vidéo sur YouTube expliquant parfaitement mes besoins avec des collisions et une première petite map en 2D isométrique. En parallèle j'ai cherché sur cette même plateforme comment déplacer une « capsule » qui sera par la suite le joueur qu'on incarnera. J'ai alors eu quelques problèmes de collisions causées par des assets invisibles, derrière des blocs voyants qui ont vite été rectifiés. Pour créer ce croquis, j'ai trouvé et téléchargé des assets gratuits sur le site officiel de Unity grâce à un lien de la vidéo en question. Hugo et moi sommes alors partis de ce croquis et de ses assets pour développer la carte de notre jeu. Un de nos premiers problèmes était donc les assets ; il en existe trop peu qui sont gratuits et qui sont adaptées pour notre jeu et l'isométrique en général. Malgré cela, on a trouvé quelques assets d'arbre en 2D qui nous serviront à combler les forêts et la plaine et peut-être trouverons-nous d'autres qui nous serviront à combler la ville ou autre endroit...

Début février, j'ai cherché et trouvé quelques tutoriels pour faire des boutons fonctionnels pour changer de scène et choisir ce que l'on veut faire. J'ai passé énormément de temps à comprendre le fonctionnement de ces derniers et à les lier entre eux et entre les différentes scènes. J'ai malheureusement trouvé que très peu de design de boutons gratuits et aucun qui n'allait réellement avec l'environnement et l'ambiance du jeu. J'ai donc implémenté des boutons basiques avec un texte de couleur pour rendre le tout sympathique à l'écran avec notamment une ombre en arrière lorsqu'on met le curseur de la souris sur le bouton en question pour voir s'il est bien sélectionné. Dans la continuité de la semaine, j'ai créé une scène complète avec 3 boutons : le bouton PLAY, le bouton OPTIONS et le bouton QUIT. Le bouton PLAY permet de basculer sur la scène principale : la « SampleScene » où tout le jeu est présent. Le bouton OPTIONS sert quant à lui à changer les options du jeu : on reste alors sur la même scène mais l'affichage de l'« OptionsMenu » s'active et prend le dessus par rapport au « MainMenu ». On a alors une panoplie de boutons : le bouton FULLSCREEN qui permet de

mettre en pleine écran, le bouton BACK qui permet de revenir au menu principal, le niveau sonore du jeu et deux menus déroulants qui permettent de changer soit la qualité du jeu, soit la résolution que le jeu prend sur ton écran. Pour ce dernier, les résolutions proposées sont les résolutions possibles de l'écran de l'utilisateur, obtenu grâce à une fonction qui ordonne les différentes résolutions dans une liste sous forme : « largeur x longueur ». Le bouton QUIT est évidemment le bouton qui permet d'arrêter complètement le jeu, et de le fermer. Pour finir, j'ai intégré l'image dessinée par Nicolas en tant que fond d'écran, pour être dans l'ambiance dès le lancement du jeu ; ce fond d'écran est trouvable aussi sur le site internet du jeu, pour retrouver le même aspect du jeu. C'est une vidéo d'une vingtaine de minutes qui m'a permis de réaliser toutes ces fonctionnalités, que j'ai reproduit sur plusieurs jours car plusieurs bugs m'ont découragé le premier jour à cause de toutes les liaisons.



Concernant la carte en tant que telle, c'est Nicolas qui a dessiné plusieurs croquis de la ville, des différentes zones comme la plaine, la jungle ou encore le marais qu'il a envoyé sur le Discord du groupe pour avoir une première idée sur comment construire la carte. Ainsi, avec Hugo, on a établi un premier prototype de la carte, très utile pour faire nos premiers tests pour tester le système de dialogue et qui sera utile pour lancer des futures quêtes et des

futurs combats ou encore implémenter des designs fait par Nicolas sur le personnage incarné, etc. On distingue alors le contour de la ville, le placement de certains bâtiments qui seront dans le futur des bâtiments importants, des maisons lambda ou les différents chemins pour entrer et sortir de Omiri, la capitale.

Pendant la semaine de vacances de février, n'étant pas chez moi, j'ai écrit toute l'histoire autour de l'univers de Gumber, histoire qui pourra être lu directement sur le site internet du jeu. Je me suis inspiré de l'histoire d'un jeu de mon enfance en reprenant l'idée de certains aspects comme l'arrivée des démons. Ce dernier sera utile pour comprendre l'univers de notre jeu et pourquoi on fait face à des créatures hostiles par exemple.

Pendant la semaine des contrôles, j'ai implémenté le suivi de la caméra qui fixe la « capsule » qui représente le joueur, assez complexe vu qu'on se déplace sur une carte en isométrique, mais j'ai vite réussi en ne changeant que le déplacement de la coordonnée x et y, car le tuto que je regardai suivait un personnage 3D qui lui bougeait sur les 3 axes.

Peu de temps après, j'ai mis en place un système de dialogue, appris par le biais de deux vidéos différentes. Pour ce faire, il faut que le joueur soit dans un périmètre défini en cercle autour du PNJ, qui est lui aussi une capsule pour l'instant. Le périmètre du PNJ peut être changé selon nos envies et suivant le placement du PNJ. A l'avenir, le PNJ, comme le joueur, aura un design spécifique et reconnaissable. Il suffit alors d'appuyer sur la touche « e » pour commencer le dialogue entre nous et le PNJ. Lorsqu'on déclenche cette action, il suffit de cliquer sur le texte qui apparaît en bas pour enclencher le second texte, et ainsi de suite. Durant ce temps, il est impossible de bouger sur la carte, le but étant d'être concentré sur les explications du personnage jusqu'à la fin du dialogue. Les mouvements sur la carte sont donc possibles après les échanges de paroles du PNJ.



Mon dernier essai avant la première soutenance était l'implémentation d'un ennemi avec une barre de vie, barre de vie qui diminue suivant son nombre de vie en fonction de son nombre maximal de vie.

Après la première soutenance, je me suis directement penché sur l'implémentation d'un PNJ qui permet de donner une quête au joueur, avec un système d'avancement et de récompense à la fin une fois la quête accomplie. Je suis donc parti de mon PNJ avec lequel j'ai fait un système de dialogue pour créer ce dernier. YouTube a alors été ma principale source d'informations avec plusieurs vidéos qui m'ont permis de faire exactement ce que j'avais en tête.

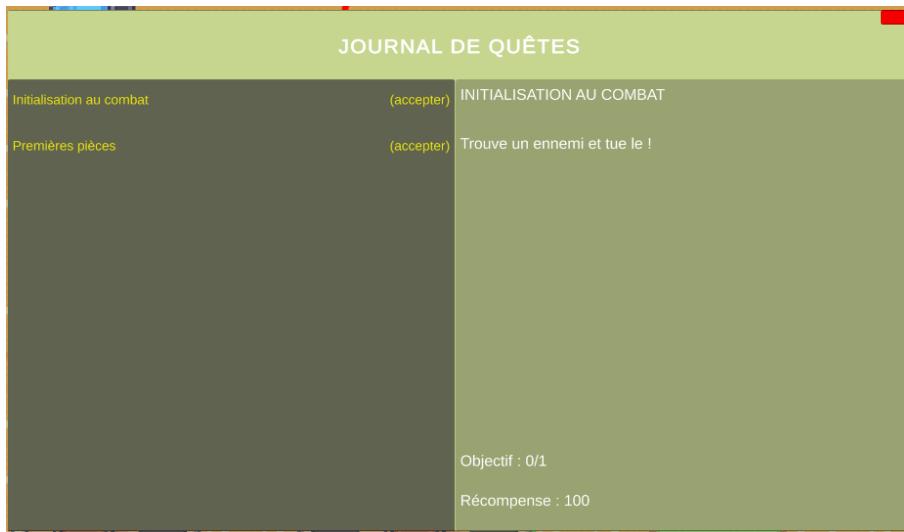
Tout d'abord, la quête en elle-même : elle possède un titre, une description et plusieurs phrases qui lui sont propres ; phrases utilisées pour transférer les informations de la quête du PNJ au joueur. Chaque phrase représente la progression de la quête : on a la phrase du début qui met le joueur dans le contexte et il apprend par la même occasion le but de la quête. S'il accepte cette quête, le PNJ lui fait part de la deuxième phrase et le joueur apprend plus en détail sur ce qu'il doit faire, ce qu'il doit chercher et à quel endroit être pour compléter ce dernier. Au contraire, s'il refuse la quête, il peut simplement reparler au PNJ qui lui dira exactement la même chose avec deux boutons : Accepter et Refuser. Une fois la quête finie, Une phrase de remerciement est donnée par le PNJ et la récompense est donnée. Le joueur peut encore interagir avec le PNJ en question qui lui donnera comme simple

réponse qu'il a fini la quête et donc il ne pourra plus la recommencer.

Pour l'instant, j'ai implémenté deux PNJ dédiées aux quêtes. Une première quête qui demande à l'utilisateur de récupérer une pièce d'or qui se trouve sur la carte et une deuxième qui demande au joueur de tuer un monstre. Le joueur a aussi la possibilité de voir directement la progression de la quête avec des icônes différentes suivant l'avancée de ce dernier au dessus du PNJ en question : le point d'exclamation jaune représente un PNJ qui a une quête à donner mais qui n'a pas encore débuté, le point d'interrogation rouge correspond à une quête acceptée par le joueur qui n'a pas encore répondu aux attentes du PNJ. Enfin, le point d'interrogation jaune communique au joueur qu'il a bien obtenu ou réussi ce que le PNJ lui demandait. Ainsi, il doit simplement aller lui parler pour obtenir la récompense.



En parallèle, j'ai mis en place un journal de quêtes qui répertorie toutes les quêtes acceptées par le joueur :



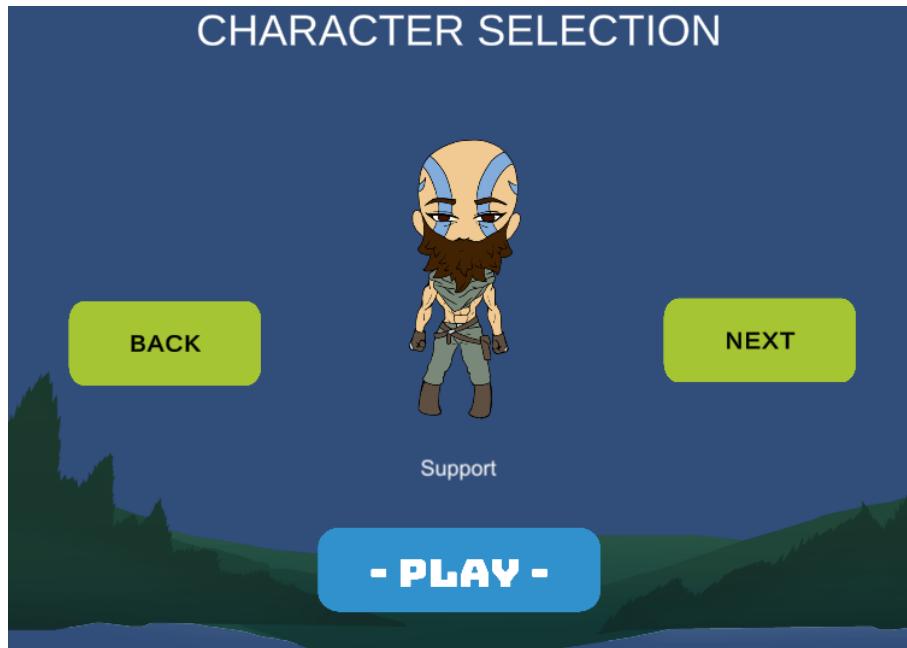
Il s'actualise en temps réel et ajoute les quêtes qui sont acceptées en appuyant simplement sur la touche « J ». Si une quête n'a pas été acceptée ou si elle est finie, elle n'apparaît pas dans le journal. On peut alors très bien voir le titre, la description, l'objectif donnée par le PNJ et la récompense (en gold) qui est propre à chaque quête et donc ces derniers changent lorsqu'on clique sur une autre. Pour l'instant, la quête “Premières pièces” est finie à 100% et permet, une fois ce dernier réussi, de gagner 50 gold. Si cette quête est finie, le joueur verra son compteur d'argent s'actualiser en temps réel en haut à droite de son écran.

Cependant, j'ai eu un problème avec ce système de quête : mettre en pause lorsqu'on entre dans un dialogue. J'ai pas mal cogité de mon côté et j'ai essayé de réussir à mettre en pause avec mes propres connaissances car trop peu de vidéos expliquaient concrètement mes attentes. Cependant, je n'ai pas encore réussi à le faire pour les PNJ donnant des quêtes. De plus, j'ai rencontré un autre problème : comment rendre l'affichage lisible de ce journal pour chaque résolution possible de l'utilisateur mais malheureusement elle n'est lisible que pour les personnes qui ont pour résolution 1920 x 1080.

Par la suite, avec Hugo, on s'est concentré sur le design de la carte et nous sommes tombés sur un très gros problème : il n'y avait pas assez d'assets pour faire une carte belle et complète en perspective isométrique malgré le fait que

c'était notre idée de base. Nous avons cherché des assets sur le site officiel de Unity mais que ça soit en 2D ou en isométrique, trop peu étaient adaptés pour notre jeu et quasiment aucune étaient gratuites. On s'est alors penché sur une carte en 2D et nous avons trouvé par la même occasion énormément d'assets de particulier, tous regroupés sur un même site, qui nous a permis pour l'instant de faire un village et quelques chemins menant aux différents PNJ et éléments de la carte. Ce changement nous a permis dans un premier temps d'avoir énormément de faciliter pour construire la carte avec des collisions et d'implémenter le mouvement et les différents affichages pour notre jeu. Avec ce changement, nous avons décidé de nous déplacer avec les touches ZQSD pour faciliter l'implémentation des animations, car nous détectons maintenant quelle touche est utilisée et donc quelle animation mettre.

Pour finir, j'ai réussi à implémenter un menu pour sélectionner sa classe (son personnage) avant d'être réellement dans le jeu. On accède à cette sélection après avoir appuyé sur le bouton PLAY du menu principal :



On a alors le choix de prendre une classe parmi les 5 existants : support, mage, paladin, assassin et hunter. Les designs (les sprites) utilisés pour ce menu ont été dessinés par Nicolas. Pour changer de classe, il suffit simplement de cliquer sur un des deux boutons BACK ou NEXT pour dérouler la

liste des personnages et lorsqu'on appuie sur le bouton PLAY, l'utilisateur incarne le dernier personnage qui était sur l'écran de sélection. J'ai réussi à lier ce système entre les différentes scènes en sauvegardant le dernier personnage sélectionné dans un script.

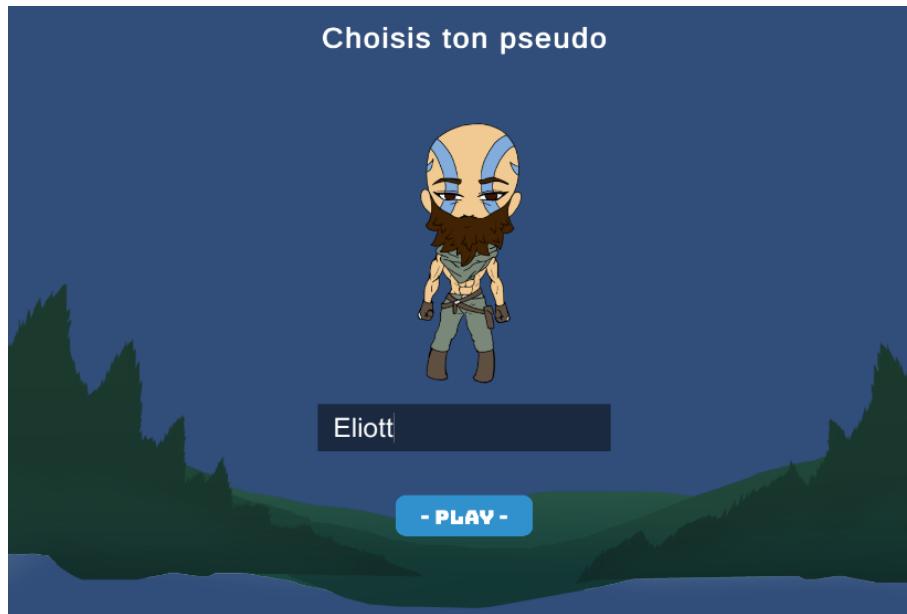
C'est à partir de la deuxième soutenance qu'on s'est réellement penché sur l'implémentation des différentes mécaniques du jeu pour le multijoueur et c'est à partir de ce moment-là qu'on a rencontré énormément de difficultés. Depuis le début, toutes les implémentations des différentes mécaniques du jeu étaient destinées à un jeu en solo, car tous les tutoriels YouTube que je suivais était conçu pour créer un jeu solo et non multijoueur. On a alors dû énormément changer les choses, que ce soit le fonctionnement de sélection de la classe, les interactions, ou encore les scripts.

Par la suite, je me suis penché avec Hugo pour faire un système de groupe, pour donner une réelle utilité au mode multijoueur. J'ai surtout implanté l'affichage de ce système plutôt que le multijoueur derrière, avec les différents panels et l'actualisation de ce dernier. Ainsi, en appuyant sur la touche « G », on ouvre un panel qui où on peut voir les joueurs dans notre groupe et les joueurs proches de notre personnage, joueurs que nous pouvons inviter à rejoindre notre groupe grâce à un bouton. Pour ce faire, je me suis principalement basé sur ce que j'avais fait pour les quêtes, car le système d'affichage pour le groupe et pour le journal de quête était très lié, que ce soit au niveau de l'architecture dans Unity ou dans les fonctions en C#. Tout d'abord, il fallait détecter les joueurs à proximité pour les inviter dans notre groupe. On a donc fait une liste « closePlayers » qui se trouve dans le GameManager et qui s'actualise à chaque nouvelle collision entre deux joueurs. Après cela, pour voir les différents joueurs dans le groupe, il fallait les distinguer : on a dû lier au joueur la bonne tête de la classe qu'il a sélectionné ainsi que son nom. La tête de la classe est définie lors de la sélection du personnage, lorsqu'on choisit pour la première fois notre classe.



Les joueurs dans le groupe sont situés à gauche et les joueurs proches de notre personnage (ceux dans la liste closePlayers) sont situés à droite : ce panel est actualisé à appui de la touche "G"

Ainsi, pour implémenter tout ce mécanisme qu'est le groupe, j'ai créé un menu qui permettait de rentrer le nom que l'on souhaitait, son pseudo. On peut mettre au maximum vingt caractères pour qu'on puisse correctement le reconnaître si on l'invite dans un groupe.



Cette sélection se trouve juste après la sélection de sa classe et elle se trouve sur la même scène. Une fois le bouton PLAY appuyée, on accède à la scène principale, la carte du jeu.

Après tout cela, je me suis intéressé aux récompenses, notamment aux récompenses lorsqu'on arrive à tuer un ennemi ou lorsqu'on finit une quête. Concernant les quêtes, on gagne un nombre d'expériences défini par moi-même. Pour l'instant, on gagne 200 xp, soit l'équivalent de deux niveaux. A chaque gain de niveau, on gagne 15 points de vie en plus qui servent à mieux se défendre contre les ennemis. Ainsi, plus tu gagnes de combats, plus tu gagnes d'expériences, plus tu gagnes de points de vie. De la même manière, si on finit un combat, on gagne entre 80 et 130 points d'expériences, donnés et actualisés directement dans l'affichage du personnage.

```
1 reference
private void HandleVictory()
{
    player = GameObject.FindGameObjectWithTag("Player").GetComponent<PlayerData>();
    sprite = player.GetComponentInParent<SpriteRenderer>().sprite;
    TextMeshProUGUI pseudo = pseudopanel.GetComponent<TextMeshProUGUI>();
    int nbxp = (UnityEngine.Random.Range(80,130));
    pseudo.text = player.pseudo + " a gagné " + nbxp.ToString() + " xp !";
    player.AddXp(nbxp);
    foreach (GameObject text in MenuManager.instance.InGameInfo)
    {
        text.SetActive(false);
    }
    MenuManager.instance._victory.SetActive(true);
    isWaiting = true;
}
```



Chaque classe a un nombre de points de vie unique et dépend notamment de la spécificité de la classe. Par exemple, la classe paladin aura plus de vie que la classe Hunter au début du jeu.

Enfin, entre la seconde et la dernière soutenance, je me suis mis sur l'IA des monstres de notre jeu et notamment l'implémentation de celle-ci dans notre système de combat. Ainsi, quand on lance un combat, on se retrouve face à un ennemi. Le combat commence et on a deux coups possibles : on peut soit bouger de deux cases, soit attaquer deux fois si on est proche de l'ennemi, soit faire l'un et l'autre. De l'autre côté, l'IA cherchera en premier lieu à attaquer le joueur s'il est proche de lui ; sinon il s'en approchera. Le monstre n'a qu'un seul coup. S'il ne détecte aucun joueur, alors il va chercher à être sur la même ligne que le joueur. S'il est déjà face à lui (sur la même ligne) alors il s'en approchera en avançant vers lui. S'il ne peut pas à cause des cases d'eau, alors il contournera cette case. Pour ce faire, nous détectons et sauvegardons les coordonnées du joueur ainsi que du monstre sur le plateau de jeu. La case en bas à gauche a comme coordonnées 0,0 et celle en haut à droite a comme coordonnées 8,5. Avec ces coordonnées, on peut calculer la distance horizontale et verticale entre les deux entités en faisant simplement la coordonnée x ou y du monstre moins la coordonnée x ou y de celle de l'utilisateur. Ainsi, avec cela, on peut comparer quelle distance est la plus faible et donc donner à l'IA les bons mouvements à adopter.

Une attaque fait 50 de dégâts, le mouton lambda a 100 points de vie et le joueur a un nombre de points de vie défini à sa création mais non fixe

car il en gagne à chaque fois qu'il monte de niveau. Ainsi, plus on augmente de niveau, plus on a de points de vie et donc combattre plus facilement le mouton possédé. De plus, lorsqu'on clique sur le monstre, on peut connaître sa vie en temps réel :



2.3 Hugo

La première chose que j'ai appris à faire a été de créer une carte sur Unity, afin qu'à terme, je puisse retranscrire le monde imaginé par mes camarades. M'aventurant dans un terrain inconnu, j'ai commencé par apprendre les fondamentaux du 2D isométrique : créer un terrain plat, des blocs en hauteur et des collisions. Par chance, beaucoup de tutoriels sont disponibles sur YouTube et la plupart sont bien expliqués, ce qui permet de comprendre ce que je fais en même temps que j'apprends.

Ensuite, ayant entendu parler de Blender, j'ai cherché des modèles que je pourrais utiliser pour la carte et je me suis renseigné sur comment exporter un projet Blender vers Unity ; j'ai cependant abandonné cette option car je me suis rendu compte que cela ne correspond pas vraiment à ce dont nous avons besoin pour ce projet.

C'est donc vers l'Asset Store d'Unity que je me suis tourné et cela a été assez frustrant de voir que peu d'assets sont disponibles gratuitement. Avec une collection trouvée par Eliott et une autre que j'ai trouvé, j'ai conceptualisé un morceau de la carte à partir du croquis conçu au préalable. Je me suis ainsi rendu compte des problèmes posés par l'isométrie : la ville – qui est l'élément central de notre carte – telle quelle ne pouvait être fidèlement représenté dans Unity car des éléments en cachaient d'autres. Afin de représenter de la manière la plus juste possible la ville, je me suis emparé d'une règle que je plaçais contre mon écran afin d'avoir une échelle constante. Il y a eu un réel effort artistique et pratique derrière la conception de ce squelette car il doit être aisément compréhensible ce à quoi correspondent chaque blocs utilisés et il doit être beau visuellement parlant. Ayant pour objectif de présenter une première version du système de combat, une autre partie de la carte finale que j'ai représenté dans Unity est la première zone contenant des ennemis, ainsi que bien évidemment, notre premier ennemi.

La première difficulté rencontrée est sans aucun doute avec Git, lorsque la carte se faisait modifier, le pull était compliqué : pour la première fois j'ai dû effectuer un merge manuel car il y avait des conflits... La solution à laquelle j'ai pensé a donc été de supprimer les fichiers problématiques afin de récupérer les nouveaux sans difficulté... mais il n'était alors plus possible d'effectuer un pull. Suite à ça, j'ai finalement supprimé l'entièreté du projet

stocké sur ma machine et réeffectué un clone. A l'avenir, nous pensons nous envoyer les modifications à part – en plus de les push – afin de simplifier les pull.

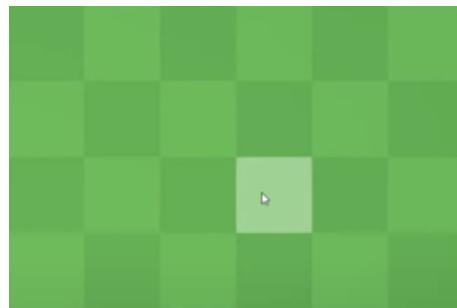
Pour ce qui est des combats, j'ai essayé de trouver un tutoriel correspondant à notre idée de base : c'est-à-dire avec un plateau de cases vue de haut directement relié à une matrice. Partant du fait que les attaques aient chacune une portée spécifique et pouvant avoir des actions sur plusieurs cases, la matrice est idéale. De plus, il se trouve que la représentation de cette dernière est assez simple.

Entre la première et la deuxième soutenance, mon objectif principal a été d'implémenter un système de combat (qui fait partie de l'essence du jeu) . Lors de la soutenance précédente, je ne savais pas encore exactement comment m'y prendre : j'avais une idée en tête en m'inspirant de Club Penguin mais ça n'allait pas plus loin... .



Comme précédemment, la première étape était de chercher des tutoriels afin d'avoir une base solide. Il fallait trouver le bon : permettant d'implémenter une IA sans trop de difficulté à l'avenir et se rapprochant au mieux de l'idée que je m'étais faite. Par chance, j'ai fini par trouver « La » vidéo ! Il n'y avait plus qu'à implémenter tout ça. Dans un premier temps, j'avais donc une grille avec des cases, chacune étant interactive : le passage de la souris sur l'une d'elle la mettait en évidence.

L'étape suivante a été de trouver un moyen de passer de notre scène principale à cette fameuse scène de combat. Ne trouvant pas exactement ce que je souhaitais sur internet, j'ai réussi à mixer les informations de plusieurs tutoriels pour réaliser mon idée : détecter les collisions d'un ennemi en étant dans une zone de contact, sans pour autant pouvoir traverser l'ennemi et changer de scène avec l'appui sur une touche.

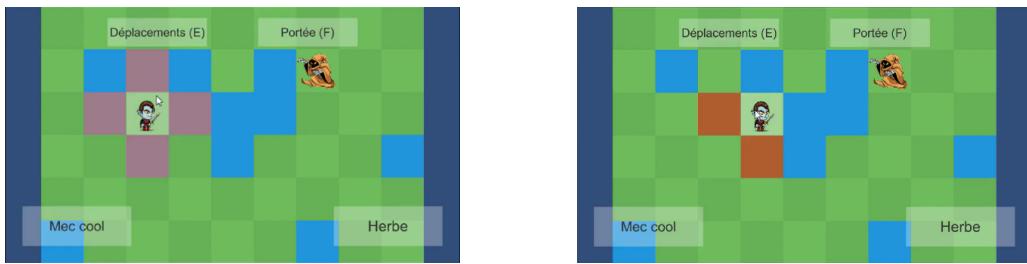


Bien, ensuite il a fallu développer le cœur du système de combat, donc faire apparaître des entités et les faire interagir entre elles. Le multijoueur n'étant pas encore implémenté, j'ai donc un premier temps conçu une version à un joueur (ce qui important de noter et qu'il n'y a donc qu'une seule personne qui agit sur le/les héros présents). Cela ne m'a pas empêché de réfléchir à d'éventuels moyens de modifier les scripts afin d'adapter ces derniers au multijoueur. Afin d'éviter d'implémenter « inutilement » des choses, j'ai préféré ne pas gérer toute la partie récupération de « sprites », notre joueur étant encore une capsule à ce moment. Afin d'éviter d'avoir trop de redondance entre deux combats, quelques règles ont été fixées : le héros apparaît toujours sur la gauche et l'ennemi toujours sur la droite mais ces derniers à des positions aléatoires, de plus, des cases servant d'obstacles (ici de l'eau) apparaissent aussi aléatoirement, en petit nombre, afin de créer de la diversité sans pour autant ruiner l'expérience du joueur.

Après avoir ajouté un déplacement et une attaque rudimentaire (destruction de l'objet avec un clic), vient l'arrivée des conditions d'actions et le côté pratique/esthétique : chaque personnage ayant ses propres attaques avec leurs spécificités, le moyen « simple » de faire a été d'utiliser des vecteurs à deux dimensions : cela permet d'avoir une information sur la portée d'un coup sur l'axe horizontal et une autre sur le vertical (la gauche étant

symétrique de la droite et vice-versa pour le haut et le bas).

Cela a permis d'ajouter des fonctionnalités supplémentaires telles que l'affichage des cases de déplacements valides (qui apparaissent en surbrillance en violet) et des cases où l'attaque est possible (en surbrillance rouge). Je suis parti sur des touches de claviers plutôt que des boutons car cela permet d'éviter d'encombrer l'écran sachant qu'il est déjà rempli de cases sur lesquelles cliquer.



Ensuite, vient la vérification de fin de partie : à l'aide d'un compteur d'ennemis initialisé au nombre d'ennemis qui apparaissent dès le départ, il suffit de vérifier s'il est à zéro après l'action du héros. Le nombre d'actions à effectuer par tour est de deux : soit deux mouvements, soit deux attaques ou bien une action de chaque. L'IA n'étant pas encore présente, il est donc impossible de perdre, c'est pourquoi seulement l'écran de victoire a été créé.



Au même moment, le multijoueur commençait à prendre forme et comme on peut le voir ici, une fenêtre de gestion d'un serveur apparaît en haut à gauche. Par chance, l'ajout du multijoueur n'a causé presque aucun souci avec le combat.

Vient le changement de scène, j'ai réussi à << faire attendre >> la fonction 5 secondes entre l'affichage de victoire et le changement de scène grâce à une astuce avec un compteur et la fonction << Update >> qui est appelée automa-

tiquement à chaque image.

Afin de finaliser cette version 1.0 du combat, il restait un souci : conserver certains « GameObject » d'une scène à l'autre, et c'est à l'aide de la création de singleton et de la fonction « DontDestroyOnLoad » que cela se réalise.

En plus du combat, un autre élément majeur entre ces deux soutenances a été la modification de la carte que nous possédions. En effet, la « tilemap » isométrique était trop contraignante (pas assez d'assets gratuits notamment) ce qui nous privait de construire un monde comme nous le concevions. Nous avons pris le pari de reprendre cette partie à zéro : nouvelle carte et nouveau système de déplacement. Eliott s'est chargé de présélectionner des packages et nous avons choisis ensemble ceux à conserver. La création/conception de la carte et l'implémentation du combat ont donc été mes deux activités. A présent, j'ai conscience des possibilités que m'apporte ces assets et j'ai essayé de créer le plus de choses originales possibles afin de donner une vie à cette carte, comme il était question lors de la première soutenance.

A l'avenir, les objectifs sont d'implémenter un système de combat multijoueur (qui récupère toutes les informations de chacun et qu'un joueur ne puisse interagir qu'avec son héros), une file d'attente sur la scène principale pour que n'importe quel joueur puisse donner son aide, du butin en récompense (points d'expériences, or ou autres objets collectionnables dans le sac à dos), un système d'apparition d'ennemis (après avoir été battu, attendre un certain temps avant qu'il revienne).

Après la deuxième soutenance, j'ai beaucoup plus collaboré avec le reste de l'équipe qu'auparavant : avec Eliott nous nous sommes très souvent donné rendez-vous afin de se mettre d'accord sur ce qu'il fallait développer en premier lieu. Nous avons tous deux nous adaptés à l'autre : l'un étant du matin et l'autre du soir... L'après-midi a donc été le juste milieu idéal. En reprenant l'affichage du journal de quête d'Eliott, nous avons pu très rapidement le remodeler afin de concevoir notre panel de groupe suivit d'un panel qui agit comme un pop-up lorsque le joueur est invité dans un groupe, lui proposant ainsi d'accepter ou refuser l'invitation.



Nous désirions faire un système de groupe vraiment abouti et qui affiche les pseudos de chacun des membres ainsi que la tête de leurs sprites (comme une sorte de photo de profil).

Cette idée qui peut paraître simple n'a cessé de requérir de nouvelles choses à implémenter et nous étions loin d'imaginer cela en le commençant ! Il a donc fallu créer une nouvelle interface pour que le joueur puisse définir son pseudo ainsi que gérer sa récupération et aussi récupérer le sprite de la tête.

Un problème majeur avec le multi-joueur est ensuite survenu : les personnages étant de préfabs, il n'y avait aucune difficulté à gérer la synchronisation des sprites car celle-ci était gérée automatiquement par Mirror ; cependant nous n'avions aucune idée de comment transmettre le pseudo du joueur aux autres joueurs connectés. C'est à partir de ce moment que Tom est venu en renfort. La page de documentation de Mirror ainsi que notre instinct nous ont permis de comprendre comment manipuler les "SyncVar" dans un premier temps. Nous avons donc décidé de créer notre propre script pour instancier les joueurs (ce qui n'était pas une mince affaire). A l'aide d'une classe servant de message pour transmettre des informations et de méthodes servant à communiquer entre clients et serveur nous sommes parvenus à gérer nous-même l'instanciation des joueurs. Ce fut un grand moment de satisfaction pour tous : nous n'avions suivis aucun tutoriel pour y parvenir. Cela a été très enrichissant car nous avons pu ensuite comprendre comment faire en sorte d'inviter un joueur dans un groupe et que tous les membres puissent recevoir l'information dans le cas de l'arrivée d'un nouveau membre.

Un petit peu avant d'avoir commencé à s'occuper du système de groupe, j'avais créé une interface de joueur qui affiche ses points de vie ainsi que son niveau. Cependant je ne savais pas comment transmettre au serveur un gain d'expérience. Réaliser ce système de groupe m'a donc fais acquérir les compétences nécessaires afin de corriger ce problème.

A ce niveau de l'avancement du projet, je commençais à être fier de notre jeu car au lancement du projet, je ne m'imaginais absolument pas capable de réaliser un jeu en multi-joueur. Enfin je commençais à comprendre comment fonctionnait ce côté du jeu vidéo. Tellement motivé à aller le plus loin possible, ma première pensée au réveil était souvent une idée de comment essayer de résoudre un bug (j'ai rencontré assez de bugs lors du développement pour avoir de quoi réfléchir dès la première heure). Je vivais pour ce jeu et mon temps se divisait grossièrement en trois activités durant cette période : manger, dormir et Gumber.

Ensuite est survenue ma plus grosse déception lors de ce projet : l'idée étant de former un groupe pour combattre ensemble les monstres, il fallait donc trouver un moyen de déplacer tous les joueurs vers une même scène. Le premier problème étant qu'avec mes connaissances actuelles, je n'étais capable de changer la scène qu'en local... Le deuxième problème (si le premier est résolu) et de s'assurer que chaque joueur est déplacé dans la même scène. J'ai donc commencé à me renseigner sur internet et voir ce que disaient les forums, rien ne correspondait à mes attentes donc j'ai essayé du côté de la documentation de Mirror mais je n'ai pas réussi à trouver ce que je désirais. Dans un premier temps, j'ai donc essayé -toujours grâce à ce que j'ai appris avec le système de groupe- de m'occuper du premier problème : à savoir, déplacer tous les joueurs d'un même groupe vers la scène de combat. J'ai eu beau essayé une multitude de techniques différentes (même essayer d'utiliser les thread) mais toujours le même résultat : essayer de lancer un combat faisait complètement planter Unity. Après plusieurs heures à avoir essayé ces différentes méthodes et avec le besoin permanent d'utiliser le gestionnaire des tâches pour stopper Unity et le redémarrer à chaque échec, j'ai finalement abandonné car je n'étais normalement pas responsable de cette tâche et je devais m'occuper d'autres aspects du développement du jeu.

Honnêtement triste et déçu car ce système de groupe n'avait donc plus de réelle utilité, je devais tout de même repartir de l'avant car même sans combat à plusieurs, il fallait finir de développer le système de combat qui nous attendait Eliott et moi. Après une réunion pour expliquer à Eliott mon implémentation de ce système, nous avons commencé par nous occuper de la récupération du sprite du joueur ainsi que de son pseudo. Grâce au DontDestroyOnLoad du joueur, il a été très simple de s'en occuper car il suffisait de

récupérer le composant (script) associé au joueur qui stock ces informations. La seule difficulté a été de pouvoir accéder au joueur tandis que ce dernier n'était pas présent dans la scène de base, la solution a été d'attendre un temps infime avant de récupérer les informations du joueur car ainsi il est alors possible d'accéder à ses composants (le faire dans la fonction Start causait une erreur nulle) et j'ai donc appris à manipuler les « Coroutine » qui permettent d'ajouter du délai dans des fonctions. Je me suis même rendu compte qu'attendre 0.01 seconde était suffisant !

```
private void Start()
{
    StartCoroutine("WaitToSetSprite");
}
IEnumerator WaitToSetSprite()
{
    yield return new WaitForSeconds(0.01f);
    hero.sprite = NetworkClient.localPlayer.gameObject.GetComponent<PlayerData>().combatSprite;
    hero.hp = NetworkClient.localPlayer.gameObject.GetComponent<PlayerData>().maxHealth;
}
```

Une fois fais, nous nous sommes penchés sur l'IA avec Eliott et nous avons pu créer quelque chose de fonctionnel mais qui rencontre des difficultés à faire le meilleur choix de déplacement lorsqu'il rencontre une case d'eau (obstacle dans le combat). Grâce à ce que j'avais implémenté lors des précédentes soutenances, il n'y a pas eu de soucis à récupérer les cases autour du monstre car cela reprenait la même logique que les déplacements du joueur.

Comme on peut le voir, il y a plusieurs étapes clés au déroulement de l'IA : Dans un premier temps on récupère la liste des cases avoisinantes au monstre ; Si un joueur se trouve dans la portée d'attaque du monstre, alors l'IA l'attaque ; Sinon elle se déplace.

```
private void HandleEnemiesTurn()
{
    UnitManager.instance.Coups = 0;
    UnitManager.instance.GetAvailableTiles(enemy);
    Tile possible_attack = UnitManager.instance.CanAttack(enemy);

    if (possible_attack != null)
    {
        Debug.Log("IA : Attaque possible");
        UnitManager.instance.Attack(enemy, possible_attack.OccupiedUnit);
        UnitManager.instance.Coups += 1;
    }
    else
    {
        Debug.Log("IA : Essaie de se déplacer");
        UnitManager.instance.MoveEnemy(enemy);
        UnitManager.instance.Coups += 1;
    }

    ChangeState(CombatState.Decide);
}
```

La fonction «Attack» sert aussi bien au joueur qu'au monstre, en effet lors des précédentes soutenances il n'y avait ni point de vie ni point de dégât donc l'entité était simplement supprimée au click. Dans cette fonction on vérifie si après le coup l'entité ayant reçu les dégâts est toujours en vie et de réagir en fonction des situations. Cette méthode permet aussi l'implémentation d'une possible animation d'attaque mais n'a pas été utilisé.

2.4 Nicolas

Après avoir définis les classes et leurs caractéristiques, je me suis mis à la recherche de leurs potentiels chara-design (character-design, l'apparence des personnages).

Au tout début, nous avions décidé de partir sur un jeu en 2.5D, ou en vue isométrique. J'ai donc, en premier lieu, dû prendre un temps pour m'habituer à dessiner avec une vue isométrique. Ayant commencé à dessiner il y a peu (un an et demi maintenant) mes bases ne sont pas très solides puisque j'ai encore du mal avec tout ce qui touche à la perspective et l'anatomie lorsqu'il y a du foreshortening. Le foreshortening correspond au fait d'appliquer de la perspective aux membres, par exemple, un bras tendu vu de face est totalement différent de si on le voyait de profil.

Je me suis donc forcé à utiliser des lignes d'horizon et de perspective pour apprendre et comprendre un peu mieux comment cela fonctionnait. En effet, n'ayant commencé que récemment j'ai principalement dessiné des personnages sans vraiment varier de contenu. Cela m'a, au départ, quelque peu déstabilisé. Pour apprendre et m'exercer j'ai d'abord travaillé au traditionnel (sur papier, ou de façon générale pas au digital) mes carnets en sont d'ailleurs témoins. Je n'ai pas vu l'apprentissage, bien que très superficiel, de la perspective comme une corvée, au contraire j'ai trouvé très intéressant et stimulant car j'aime beaucoup apprendre dans les sujets qui m'intéressent. En parallèle, j'essayais de trouver différentes idées de chara-design pour les classes du jeu. Rien de très concluant cependant.

Une fois que j'ai considéré que j'avais assez compris les bases je suis passé au digital, sur le logiciel Clip Studio Paint, un logiciel payant sur lequel il est possible de dessiner et animer avec une grande variété d'outils et d'assets. Je me suis donc lancé dans la création de la première des cinq classes que l'on va retrouver dans le jeu, la classe d'assassin.

Pour son physique de base j'ai emprunté Ren, l'OC (original character) d'un ami à moi parce que j'ai pensé que son personnage irait bien en tant qu'assassin. Une fois son accord obtenu j'ai proposé l'idée du personnage à mes camarades avec un fan art que j'avais déjà fait de ce personnage. Le chara-design de base du personnage a été validé par le groupe et je me suis donc mis au boulot.

Pour commencer j'ai utilisé une image de planche présentant déjà des lignes pour de l'isométrie, car cela m'aurait pris beaucoup de temps de le faire à la main, sur laquelle j'ai placé, pour m'aider un pantin 3D, un modèle 3D

d'humain que l'on peut manipuler à sa guise dans le but de s'aider à réaliser des poses dans certains angles. Cela est très pratiques car on peut également modifier ses proportions ce qui offre une grande variété de possibilité. J'ai donc retracé sur un calque différent le pantin du modèle 3D afin d'avoir la base à partir de laquelle j'allais pouvoir commencer à concevoir le personnage.

J'ai passé un peu de temps à chercher une tenue pour le personnage étant donné que j'avais déjà son physique de base en tête, j'ai donc demandé l'avis du groupe sur le sketch (brouillon/esquisse) que j'avais faite et suis donc passé à la couleur. Ce n'est pas la partie du dessin que je préfère parce que j'ai du mal à choisir et gérer les couleurs, je ne trouve pas la théorie des couleurs évidente. De plus, pour ne rien arranger, j'ai plus l'habitude du traditionnel où, personnellement je n'utilise jamais de couleur à ma grande honte. Avec le peu de bases dont je dispose dans la coloration, j'ai fait ce que j'ai pu. J'ai choisi de lui donner des couleurs peu vives qui sont le vert et le bleu foncé. J'ai de nouveau fait valider le chara-design par défaut de l'assassin et suis directement passé à la vue de dos.

Je ne dessinais jamais mes personnages de dos et j'ai donc, de nouveau dû repasser au traditionnel pour revoir l'anatomie du dos et principalement les différents muscles qui le composent et leur placement. Cela m'a pris plusieurs heures et plusieurs pages remplies de torses musclés vu de dos. Une fois que je me suis senti assez à l'aise avec l'anatomie du dos je suis repassé sur CSP pour m'attaquer à l'assassin vu de dos. J'ai tout de même eu un peu de mal pour le réaliser, il faut dire que ce n'est pas en quelques heures que l'on peut maîtriser totalement quelque chose. Cependant j'ai quand même été satisfait de ce que j'ai fait.



J'ai ensuite choisi de passer directement à l'animation de marche du

personnage. Je ne fais pas non plus d'animation, même si je me renseigne régulièrement dessus et qu'il s'agit d'un domaine que je trouve fascinant et très riche, le travail demandé par cette pratique est impressionnant et le rendu l'est d'autant plus lorsque l'on sait comment cela a été réalisé. Pour cette animation, j'ai d'abord recherché des GIF pour m'en inspirer. J'ai également regardé une vidéo décrivant les principes de bases d'une boucle de marche en animation (cycle walk). A l'aide des explications données par cette vidéo sur la marche de face et de profil, dommage pour moi il n'y avait pas d'explication pour la vue isométrique, je n'ai eu d'autres choix que d'improviser un peu. J'ai probablement passé plusieurs minutes à marcher sur place à différents rythmes en décomposant le mouvement pour m'en imprégner. Un regard vers mon reflet à ma fenêtre, et j'ai directement pensé que j'avais perdu toute crédibilité. Cependant, cette étape m'a permis réussir à obtenir un squelette de marche plus ou moins convaincant et c'est tout ce donc j'avais besoin.

J'ai ensuite pris le temps de passer le squelette en volumes simples, à savoir des cylindres, des sphères et des cubes. Cette partie m'a pris pas mal de temps, plus que la partie du squelette d'ailleurs parce que je m'efforçais de corriger les erreurs que je voyais. J'ai dû voir puis revoir puis corriger puis revoir encore plusieurs fois pour arriver à un résultat qui me convenait.



Une fois cette étape passée j'ai séparé les frames dans le but de réaliser des frames dites « in-between » ou intermédiaires dans le but de rendre l'animation plus fluide. Après avoir fini de faire ces frames j'ai enfin pu passer

au croquis du personnage en lui-même pour l'animation. Une fois satisfait je suis passé au line et à la coloration pour laquelle j'ai fait le choix de ne pas faire d'ombrage.

Cette première animation m'a pris plus d'une dizaine d'heures pour un résultat peu concluant avec beaucoup d'erreurs mais que l'on peut considérer comme passable.

J'ai ensuite voulu passer au cycle walk de dos mais j'ai eu un blocage, je n'arrivais pas le moins du monde à faire la base de cette animation. C'est à ce moment-là que Tom est venu me demander de réaliser un logo pour le jeu. Après lui avoir envoyé quelques croquis avec différentes idées (je n'avais jamais eu à faire de logo à ce moment-là, ce projet est d'ailleurs à ce jour la seule raison pour laquelle j'ai eu un logo à réaliser) nous avons décidé du logo qui allait devenir notre logo actuel que nous avons fait valider par les autres. J'avais pour le concevoir choisi de lui donner un côté nature en rajoutant un côté aérien pour renvoyer une idée de liberté. J'ai eu comme référence le logo du jeu Dofus.



Tom m'a ensuite demandé de réaliser la bannière du site, qui allait également servir d'écran titre pour le jeu. Je n'ai pas l'habitude non plus (oui ça doit devenir un peu trop habituel) de réaliser des paysages ou des backgrounds et j'ai donc cherché des références qui me parlaient un peu pour m'en inspirer. J'ai choisi comme principales références celles des jeux Genshin Impact et The Legend of Zelda : Breath of the Wild. J'ai pu réaliser le croquis sans trop de peine, mais une fois lancé dans la coloration, je me suis rendu compte que je n'aimais pas le rendu de ce que je faisais. Aussi ai-je dû demander de l'aide à l'ami à qui j'ai emprunté le chara-design de l'assassin. Il a gentiment accepté de faire la coloration ce qui a donné un rendu meilleur que ce que j'aurais réussi à faire même si j'y avais passé deux fois plus de temps aussi lui en suis-je reconnaissant.



J'ai également créé les croquis de la carte, en détaillant les différents éléments de celle-ci pour aider Eliott et Hugo pour sa conception dans Unity. J'ai réalisé le croquis global de la zone puis détaillé chaque sous-partie de la zone en essayant d'être assez cohérent dans les différents placements des éléments.

Après cela, je me suis relancé dans le cycle walk de dos. J'ai eu beaucoup de mal avant d'obtenir quelque chose de plus ou moins acceptable, mais je n'en demandais pas plus. Le nombre de frames par secondes me perturbait un peu, c'est pourquoi je me suis tourné vers mon groupe pour connaître leur avis et point de vue sur la question. On a fini par choisir après une petite discussion. Une classe était donc terminée d'un point de vue déplacement et apparence dans le jeu. J'avais de plus, les pantins de bases et les volumes des cycles walk faits et prêts pour les classes suivantes.



J'ai ensuite réalisé un mannequin pour l'apprentissage du combat, j'ai pris comme référence un épouvantail et des sacs de frappe pour lui. J'ai également réalisé son animation lorsqu'il prend des dégâts. Cette animation était marrante à faire bien que j'ai eu du mal à conserver les proportions du mannequin dans son mouvement de balancier. L'animation lui permettait d'ailleurs de ne pas être juste un sprite immobile lorsqu'il se fait frapper étant donné qu'il ne peut pas attaquer ni bouger.

Ce n'est qu'après que je me suis lancé dans la conception des chara-design des classes restantes. On retrouve des classes typiques de RPG, il y a le paladin, le chasseur, le mage et le support.

Pour chaque classe j'ai cherché des références sur Pinterest et me suis inspiré des images et chara-design qui me plaisaient le plus et qui me parlaient. J'ai à chaque fois pris mon temps pour également choisir des couleurs et trouver les inspirations que je voulais. Pour faire valider le chara-design des classes j'envoyais le flat (c'est-à-dire qu'il n'y a que le line qui a été fait et les premiers aplats de couleur).

Pour les classes de chasseur, paladin et mage, j'ai conservé l'idée de base que l'on se fait de ce type de personnage pour coller avec leurs rôles respectifs.

Pour le mage, je lui ai donné un chapeau pointu et une tenue ample proche d'une robe/cape. Je lui ai rajouté une sacoche pour renforcer son côté aventureux mais je ne suis pas allé beaucoup plus loin dans la recherche de chara-design. Il a une écharpe qui lui recouvre le bas du visage pour un côté mystérieux qui m'a également permis de lui rajouter une touche de couleur étant donné que nous ne sommes pas sur le personnage le plus coloré que l'on puisse trouver. J'ai choisi de lui donner une couleur bleu foncé.

Pour moi, un chasseur est quelqu'un d'assez svelte avec une armure légère, plus proche d'une tunique rembourrée que d'une armure telle qu'on l'imagine en général. Je lui ai donc donné une tenue comme je l'imaginais en utilisant des teintes de vert et de marron pour rappeler le côté proche de la nature. Le paladin lui est un tank, il dispose d'une armure lourde, d'une grande épée et en général d'un bouclier. En général il est proche d'un culte et dégage une impression de noblesse. C'est donc ce que j'ai recherché pour le créer. J'ai choisi de lui donner des couleurs qui iraient bien avec cette idée.

Enfin, pour la dernière classe, le support. Je ne savais pas vraiment vers quoi partir à la base. Avec le groupe on avait décidé que la classe serait assez sympa à jouer histoire que des personnes puissent potentiellement avoir envie de le jouer. Après avoir cherché différentes idées sur Pinterest. J'ai fini par partir sur l'idée d'un druide de type celte. J'ai essayé de lui donner un côté assez guerrier, assez sauvage aussi. Pour dénoter avec son rôle et garder le côté guerrier celte, j'ai choisi d'en faire un personnage bien musclé.



Ensuite, je me suis principalement concentré sur les animations de déplacements des classes restantes afin de pouvoir commencer à les implémenter dans le jeu pour qu'il puisse commencer à ressembler à quelque chose d'original.

Etant donné que j'avais déjà à disposition les squelettes et volumes nécessaires pour réaliser les cycles walk des classes restantes, les suivantes devaient me

prendre moins de temps que la première. Cependant, vu que je n'ai pas l'habitude d'utiliser les fonctionnalités d'animation de Clip Studio Paint, je ne savais pas du tout comment récupérer mon travail pour pouvoir l'appliquer à chaque classe. En effet j'avais choisi de retracer les frames par-dessus le sprite par défaut des classes. Ne trouvant pas le moyen d'avoir les images pour les appliquer à la chronologie, j'ai craint de devoir reprendre le processus depuis le début à chaque fois. Personnellement je me voyais pas passer plus de 6h par animation à chaque fois. Au vu de ce qu'il me restait à faire, j'en aurais eu pour plus de jours complets de travail. Heureusement pour moi, l'ami que j'ai déjà mentionné plus haut m'a montré comment faire. Finalement, il s'avère qu'un simple copier-coller faisait l'affaire, j'ignorais que cela fonctionnait sur le logiciel, mais ça m'a sauvé la vie. Je n'avais plus à recommencer à zéro chaque animation.

J'ai donc pu me lancer à fond dans l'animation. Néanmoins, j'avais sous-estimé la quantité de devoirs à venir et, naïvement, je me disais que je n'avais qu'à faire une animation par soirée. Ayant du mal avec l'organisation de mon temps je me suis retrouvé débordé à ne le faire que lorsque trouvais le temps de m'y mettre. Je n'aime pas devoir m'interrompre en plein milieu d'un dessin/animation. Je les ai donc faites petit à petit quand j'avais le temps et pendant les vacances.

Par soucis de temps et parce que cela commençait à devenir assez répétitif d'avoir à dessiner et redessiner les mêmes poses, les mêmes personnages dans des positions qui ne changeaient que légèrement, j'ai choisi d'ignorer la partie sketch du personnage et je repassais directement sur les volumes. Je dois dire que l'étape d'animation de marche a été pour moi très frustrante, comme je viens de l'expliquer, cette étape m'a obligé à dessiner et redessiner les mêmes choses pendant des heures et des heures, j'avais donc dû passer mon cerveau en mode pause. Je dessinais sans réfléchir, j'étais rentré dans un stade où je n'avais que le fait d'avancer dans mon travail. Machinalement, en devenant plus précis j'avais trouvé un bon rythme de travail. Bien que ça soit également grâce au temps que je consacre à m'entraîner au dessin, je me sentais de plus en plus à l'aise quand je dessinais. Dans tous les cas, cela m'a permis de finir les animations plus vite que je ne l'aurais pensé.

J'ai cependant eu du mal avec celle du mage car je n'ai pas l'habitude de détailler les tissus et de comprendre la façon dont cette matière bouge et réagit aux forces.

Le temps que j'ai dédié à chaque à animation était certes plus court que pour les premières en raison du croquis déjà présent, mais je me suis rendu

compte que les chara-design de base des différents perso étaient un peu trop détaillés et complexes pour mon niveau d'animation et le temps que j'avais à disposition. J'ai donc dû simplifier les chara-design des personnages sans pour autant que le changement soit drastique et que l'on ne reconnaissse plus le personnage. Je m'en suis rendu compte la première fois, lorsque j'ai commencé à animer de façon « industrielle ». J'ai donc après en avoir parlé avec mon groupe pris la décision de simplifier les chara-design des personnages afin la réalisation des animations ne me prenne pas trop de temps. J'ai donc choisi de simplifier ce qui serait trop complexe pour moi de faire ou trop long pour moi.

Pour l'animation du chasseur, je n'ai pas eu à modifier grand-chose. J'ai uniquement choisi de ne pas refaire les hachures présentes sur son gambison (tunique rembourrée que l'on porte en général sous une côte de mailles). Avoir à refaire les hachures à chaque fois en restant cohérent entre les images m'aurait demandé trop de temps et d'effort pour un détail auquel on doit faire attention pour le remarqué.

Dans le cas du paladin, j'ai fait le choix de ne pas reproduire tous les détails de son armure et les ceintures qu'il porte autour de sa taille. Petite remarque, les ceintures sont la principale chose que j'ai choisi de retirer dans les animations.

Ensuite, pour le support, je ne sais pas pour quelle raison, probablement par inattention j'ai oublié le badge qu'il a normalement sur sa cape. Ses tatouages sur la tête m'ont donné du mal mais j'ai choisi de ne pas les retirer, cela aurait été trop étrange et radical. J'ai également moins fourni sa cape en plis et n'ai pas dessiné son deuxième œil par soucis de simplicité. Une nouvelle fois le nombre de ceintures a diminué. J'ai aussi moins détaillé ses muscles, dans le cas de la marche de dos cela m'a permis d'éviter les nombreuses erreurs que je fais encore puisque je n'ai pas l'habitude de dessiner des personnages vus de dos. Cela m'a également obligé à ne pas passer trop de temps à dessiner les muscles, chose que j'aime particulièrement faire.



Cela m'habitue également à faire des traits plus efficaces quand je dessine ce que je trouve pratique et amusant quand je dessine des choses hors projet (chose qui occupe la plupart de mon temps lorsque je ne suis pas en train de travailler). Je m'accordais tout de même quelques pauses pour éviter de surcharger à cause des animations. Pour l'anecdote, j'ai passé plus de temps au total à faire des animations pour ce projet que le temps pris par un ami en école d'animation à passé sur un projet d'animation.

Enfin, dans le cas du mage, je lui ai retiré sa sacoche qui me posait un problème dans l'animation. J'avais du mal à l'imaginer bouger aussi ai-je choisi de la supprimer pour l'animation. Cette sacoche étant la première raison pour laquelle j'ai choisi de simplifier les chara-designs, la sacoche et la cape pour être plus précis. Je lui ai également retiré sa ceinture et je n'ai pas dessiné ses yeux dans cette partie. J'ignorais à quel point le tissu en mouvement pouvait être compliqué à animer. Il est plus que probable que cela ne soit dû qu'à mon manque d'expérience. De dos j'ai également fait en sorte que l'on ne voit pas ses pieds, chose qui m'a permis d'épargner pas mal de temps.



Une fois les animations terminées, j'ai été sollicité par Eliott car il m'a demandé de redessiner les personnages en version simplifiée, ou chibi comme je les appelle en raison de ma forte tendance à lire des mangas et regarder des animes. Il m'a envoyé un modèle de ce qu'il avait en tête pour l'apparence des personnages dans l'écran de sélection des personnages. Après lui avoir envoyé un sketch de pantins dans la forme chibi et attendu sa validation je suis passé au travail.

Ayant déjà les chara-design de base des personnages, je n'ai eu qu'à les passer sous la forme chibi à l'aide du mannequin que j'avais présenté à Eliott. J'ai un peu l'habitude de ce type de dessin aux proportions simplifiées, je ne mentirais pas en affirmant que je maîtrise ce style de dessin mais je dirais que j'aime bien dessiner des chibi. J'ai passé un après-midi pour faire tous les personnages sous la forme de chibi. Je les envoyais toujours au fur et à mesure que je les faisais pour obtenir validation de la part de mes camarades.



J'ai beaucoup aimé les dessiner de cette façon car je les trouve assez mignons et plus attachant dans ce style. Je discerne toujours un certain nombre d'erreur pour chacun d'entre eux mais elles ne sont pas suffisamment importantes ou discernables si l'on ne se penche pas vraiment dessus. D'ailleurs, on voit souvent les erreurs après avoir envoyé son travail, ce qui est frustrant mais ce n'est qu'à ce moment là que l'on a le recul nécessaire pour voir ce qui ne va pas.

Quoi qu'il en soit j'ai beaucoup aimé cette partie car elle m'a permis de sortir de la période que j'ai passé à dessiner sans réfléchir à cause des animations. Pour l'anecdote parmi les 5 classes, l'une d'entre elle fait l'unanimité du fait de son chara-design. Il s'agit du mage qui je suis d'accord est très sympa. Même si je pose une mention honorable pour le support, le mage est également l'image que je préfère parmi les différentes classes sous la forme chibi.

Après cela, Hugo m'a parlé du système de combat qu'il était en train d'implémenter et m'a demandé si j'avais des images des personnages dans des positions de combat et regardant vers la droite. Puisque je n'en avais pas faite je lui ai dit que j'allais les faire aussi vite que possible. Je me suis donc lancé dedans en me renseignant en parallèle sur comment implémenter les animations de déplacement des personnages tout en conservant le choix de sélection de personnage fait avant de lancer la partie.

Pour ce qui est des images de combat, je les ai faites aussi vite que j'ai pu. J'ai choisi de les réaliser dans des proportions normales pour garder un côté plus sérieux et m'amuser un peu en dessinant des personnages dans des positions que l'on ne peut pas qualifier de neutre. J'ai également fait en sorte qu'il regarde tous dans la même direction.

J'ai commencé par la classe d'assassin. Pour ce faire, j'ai fait quelque chose que je n'avais pas eu à faire depuis longtemps pour le projet à savoir ouvrir Pinterest à la recherche de poses intéressantes pour une classe telle qu'un assassin car je n'arrivais pas à obtenir quelque chose de concluant quand j'y

allais de tête. J'ai donc cherché et tenté plusieurs poses avant d'en trouver une dont le sketch me plaisait et ne souffrait pas de trop de problèmes. Je cherchais une pose assez dynamique qui se prêtait à une personne maniant des dagues tout en mettant lesdites dague en valeur dans la position. Après avoir dessiné le personnage et en être plus ou moins satisfait, je l'ai envoyé à mes camarades pour le faire valider et pour qu'ils puissent le sauvegarder de leur côté, là où ça leur serait le plus utile.

Ensuite je suis passé au paladin. Pour cette classe, je n'ai pas eu à chercher longtemps, en effet le hasard a voulu que pendant mes exercices, je dessine un mannequin ayant une pose qui sied parfaitement au port d'une épée et d'un bouclier. Je l'ai donc copié-collé et j'ai redessiné par-dessus ce que je voulais, à savoir le paladin avec un bouclier et une épée. Une fois le line fait, je me suis dit que le bouclier faisait assez vide sans le moindre motif ou ornement quel qu'il soit. J'ai donc choisi de dessiné une fleur de lys sur l'écu, en effet je me suis dit que cela rappellerait le côté noble du paladin. La fleur ne rend pas vraiment comme je le voulais, elle n'est pas assez ressemblante et fidèle à l'originelle. Elle a cependant le mérite de combler le vide qui se faisait ressentir sur le bouclier sans elle. Affublé de cet ornement et avec le choix des couleurs, le bouclier me fait penser au bouclier Hylien de la série The Legend of Zelda, je ne me suis rendu compte de cette ressemblance que lorsque j'ai tourné la tête après avoir fini le motif. J'ai le bouclier dans ma chambre, en y repensant c'est peut-être pour ça que j'ai inconsciemment choisi ces couleurs-là.



Ensuite, je me suis attaqué au chasseur pour lequel je me suis lancé dans des recherches pour trouver une position inspirante. J'ai fait un premier sketch que j'avais assez détaillé mais qui ne m'a finalement pas satisfait, aussi ai-je choisi de chercher d'autres idées de poses qui irait bien avec un archer. J'ai fini par en trouver une qui m'a plu et je me suis donc lancé au travail. Une fois le dessin fini cependant je me suis rendu compte du grand nombre d'erreurs de proportions que j'avais faites. Je ne peux pas voir ce dessin sans m'énerver contre moi-même c'est pourquoi j'ai pris la décision de le refaire. Les erreurs m'énervent beaucoup trop pour que j'accepte de garder cette image dans le jeu, je ne peux pas voir ce dessin sans ses erreurs beaucoup trop visibles à mon goût.

J'ai donc enchaîné avec le mage. Je n'ai pas eu à chercher longtemps pour trouver la pose du personnage. C'est je pense, la pose sur laquelle je me suis le plus amusé. Je lui ai donné un bâton/sceptre qui fait, je l'espère suffisamment sorcier même si la couleur que j'ai choisi pour la sphère qui se trouve au bout du bâton ne me plaît pas vraiment. Je lui ai aussi donné un grimoire qui flotte dans les aires avec ses pages qui se tournent seules, sans avoir besoin d'être touchées par le personnage pour accentuer son côté magique. En y repensant maintenant, je pense que j'aurais dû ajouter un petit effet de glow au niveau de ses yeux pour toujours plus accentué le fait qu'il soit en train de faire de la magie. J'ai vraiment apprécié dessiner cette pose pour ce personnage, même si m'occuper des mains a été quelque peu fastidieux je l'avoue, j'ai même été obligé de prendre ma propre main en photo pour m'en servir de référence puisque je n'en trouvais pas qui allait avec ce que j'avais en tête.

Enfin, il reste le support, je n'ai pas eu à chercher longtemps pour lui non plus, j'ai néanmoins eu besoin de solliciter mes camarades afin que l'on décide ensemble quelle arme donner à ce personnage. On a fini par voter pour un bâton pour lui aussi. A l'instar du sceptre du mage, le bout de l'arme du support ne me satisfait pas mais au moins la pose me plaît.



Cette partie m'a plu même si les erreurs de proportions que j'ai faites m'énervent quelque peu. Au moins, cette partie m'a permis de dessiner de façon « normale » même si je ne pouvais pas me permettre d'aller à fond dans les détails. Par soucis de temps, j'ai également choisi de ne pas faire d'ombrage et de ne pas appliquer de lumières non plus.

J'ai implémenté les images des deux types de pnj (personnages non jouables) qui étaient présents sur la map à la base.



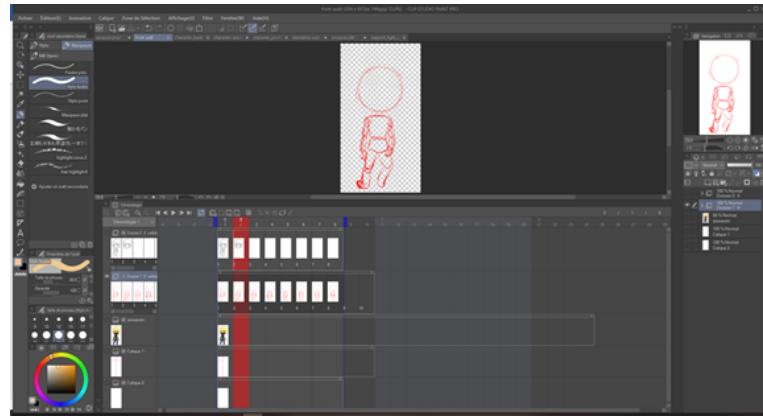
Je vais maintenant parler de l'implémentation des animations. J'ai commencé par effectuer des recherches, regardes des vidéos explicatives, des tutos pour voir comment faire. J'ai tenté plusieurs choses avec les animations de déplacement que j'avais déjà préparées et décomposées en séquences d'images. Le problème venait cependant avec la partie code, en le lisant et en cherchant comment faire je ne trouvais pas où et comment je pourrais implémenter les animations avec le code.

J'avais donc fabriqué des préfabs des classes pour voir le lendemain si, avec l'aide d'Elliott et d'Hugo on pouvait trouver comment implémenter les animations dans le jeu tout en conservant la sélection de personnage. Nous n'avions cependant rien trouvé à ce moment-là.



J'ai essayé de placer les codes d'implémentation nécessaires à l'implémentation des animations à différents endroits du code pour voir. Cependant, cela n'a pas marché. Je me suis donc retrouvé bloqué dans une impasse. Le lendemain, on m'annonçait une nouvelle pour le moins déroutante, en effet on venait de m'annoncer que le type de vue du jeu venait de changer et avec cela le mode de déplacement pour que cela soit plus simple pour tout le monde. Pour ce qui est du changement du mode de déplacement, je n'ai rien pas à me plaindre. Cependant, à ce moment-là, j'étais entré dans un état de panique. En effet, j'ai pensé que toutes les animations que j'avais faites jusqu'à présent n'allait servir à rien et que j'étais bon pour repartir de zéro. Après en avoir discuté avec les autres, nous avons décidé qu'on les garderait quand même en temps que déplacement dans les directions en diagonale. Cela voulait, cependant dire que j'aurais de nouvelles animations à réaliser pour les nouvelles directions de déplacement. En apprenant la nouvelle à cause mon état de choc, l'ami que j'ai déjà cité plus tôt avait gentiment accepté, une nouvelle fois de m'aider, mais pour les animations cette fois. Il allait donc m'aider à faire les animations pour les nouveaux déplacements.

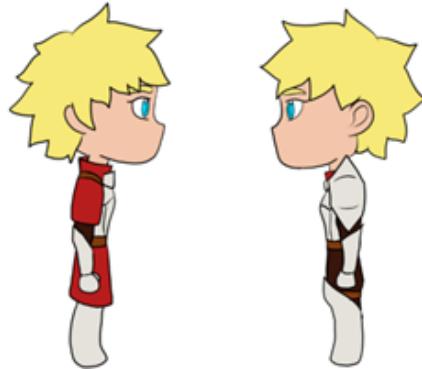
Dans le feu de la nouvelle, j'ai même commencé les volumes de l'animation que j'ai pu faire en moins de temps qu'avant, probablement par habitude et parce que j'ai trouvé plus facilement de bonnes références. De plus, on était sur le style chibi donc d'autant plus simple.



J'ai également commencé à réaliser les personnes dans la forme chibi de profil et de dos. Le lendemain, après avoir pull le nouveau code de déplacement j'ai quand même réessayé d'implémenter les animations pour être sûr de savoir comment m'y prendre et ç'a enfin marché après plusieurs tentatives. Enfin, c'était à prendre avec des pincettes parce que cela ne prenait pas en compte la sélection de personnage. J'aurais donc besoin des explications d'Elliott afin que cela fonctionne sans problèmes.

Plus tard, j'ai choisi de finir de faire les classes sous forme chibi dans toutes les directions. C'était plus marrant que lorsqu'ils avaient toutes leurs proportions en vue isométrique je ne peux pas mentir dessus.





Après avoir fini de réaliser les images des classes dans les quatre nouvelles directions, j'ai pris le temps pour refaire l'image de combat du chasseur. En effet, comme je l'ai déjà mentionné, l'image précédente me hante du fait des erreurs au combien nombreuses de proportions que je ne peux m'empêcher de voir dès que je pose les yeux dessus. Pour les autres classes les erreurs sautent moins aux yeux mais pour ce dessin-là. Je le considère définitivement comme un échec. J'aurais dû directement le refaire.

Je me suis donc attelé à le refaire. J'ai de nouveau cherché des images sur Pinterest afin de trouver des poses intéressantes. J'ai finalement d'abord choisi d'en réaliser une de tête. Après avoir fait un sketch que j'ai bien apprécié et dans lequel je ne voyais pas d'erreurs trop flagrantes j'a choisi de le faire valider par le groupe qui m'a donné le feu vert. Je me suis donc lancé dans le line, étape pour laquelle j'ai toujours du mal. J'ai comme pour les autres sprites de combat choisi de ne pas l'ombrer ou mettre de lumière. J'en suis quand même plus satisfait même si son visage ne me satisfait pas vraiment. Je trouve de plus son expression trop neutre, on peut cependant faire passer ça pour une expression de personne calme qui sait garder son sang-froid comme tout bon archer qui a besoin de concentration pour viser et tirer ses flèches.



Etant donné que je suis chargé de la partie chara-design, c'est également à moi qu'incombe la tâche de créer et concevoir les mobs. On m'a dit de m'inspirer des bouftous, des espèces de mouton en forme de boule provenant du monde de Wakfu/Dofus. J'ai donc commencé à réaliser des croquis pour la recherche de chara design. Je me suis dit que pour lui donner un côté plus agressif j'allais l'affubler de grosses cornes de bétail pour lui donner un air féroce et dangereux. Je lui ai donné également des yeux rouges et un regard qui laisse à penser qu'il a les sourcils froncés. C'est un méchant mouton et cela se voit. Après cela reste une boule de laine dont l'apparence est plus proche d'une sphère que d'un mouton.

J'ai donc fait valider mes croquis par le groupe pour savoir si ce concept leur plaisait. Il a été accepté et validé à l'unanimité. Parmi les croquis, j'avais fait une version avec une corne cassée que j'ai changé. Finalement, les deux cornes du mouton sont intactes et bien brillantes. En ce qui concerne les couleurs du mob, j'ai choisi une couleur proche de celle de la laine des bouftous (créature dont j'ai parlé plus haut). Pour les cornes j'ai choisi une teinte sombre. En ce qui concerne sa peau j'ai opté pour du gris. Enfin ses yeux, j'ai choisi de lui donner les yeux rouges. Heureux hasard, cela coïncidait avec le laure écrit par Eliott. Une partie que j'ai considérée comme marrant a été d'ombrer ces boules de laines. J'ai donc ensuite entrepris le réaliser dans les quatre directions. C'était tout aussi marrant même si la réalisation des cornes de profil m'a posé un peu de problèmes en raison de leur perspective.



Ensuite je me suis donc naturellement attaqué au boss. Le chef des moutons. Je suis repassé à des sketch pour des concept de boss. J'ai imaginé plusieurs concepts pour le boss afin que l'on ait le choix de son apparence et du type de créature qu'il serait. J'ai voulu garder le côté mouton. Encore une fois j'ai renoncé à lui faire une corne sur deux cassée.

J'ai donc d'abord pensé à une espèce de centaure. Il était doté d'un torse humanoïde, était armé d'un marteau et affublé d'une clochette autour de son cou qui était recouvert de laine. Je me suis dit pourquoi pas un mélange entre un centaure et un minotaure. L'idée et le concept de base me plait donc je me suis lancé dans le sketch de cette créature sortie de mon imagination. J'en ai fait une version avec et sans marteau. Celle sans marteau était de face n'avait pour différence que ses avant-bras, pour cette version là j'ai recouvert de laine ses avant-bras pour conserver le côté laineux du mob.

Le deuxième concept auquel j'ai pensé était tout simplement un giga mouton. Je ne vois pas comme le décrire autrement que de cette façon. Je l'ai fait plus gros, c'est assez compliqué à voir vu qu'il s'agit de croquis et qu'ils

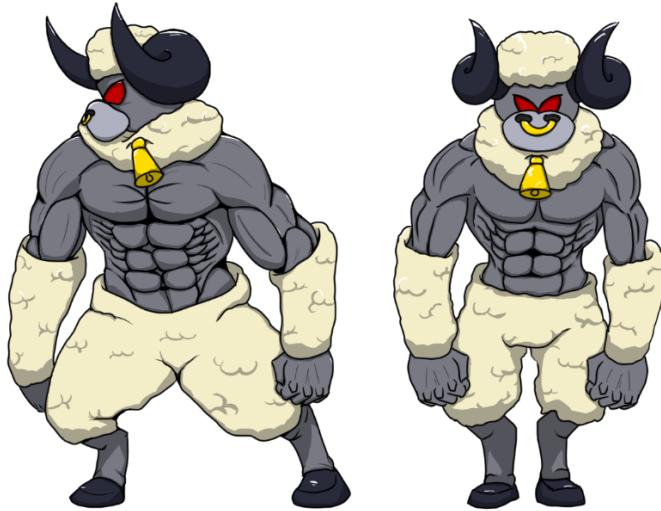
ne sont pas vraiment à l'échelle. Bref, pour le concept de celui-là j'ai fait en sorte que ses cornes soient bien plus grandes en termes de proportions que celles du mouton de base. J'ai également eu envie de m'inspirer de Sleipnir, le cheval à 8 pattes d'Odin, Père de toutes choses de la mythologies nordique/scandinave/viking. J'ai également pensé à Appa, le bison volant dans la série animée Avatar le dernier maître de l'air. Et j'ai donc choisi de l'affubler de 3 paires de pattes.

Enfin, il reste la version qui a été validée. Pour ce concept là j'ai juste pensé au minotaure, tel qu'il est décrit et représenté dans la mythologie grecques et les légendes l'entourant. J'en ai fait une montagne de muscles, avec un collier de laine autour du cou. Les avant-bras recouverts de laine et le bassin jusqu'aux cuisses également recouverts de laine. En ce qui concerne le bas de ses jambes j'ai plus pensé à un satyre (ou faune si son nom romain vous sonne plus familier). Cela contraste avec sa carrière de mastodonte et permet de renforcer son côté musclé. J'ai choisi dans son premier croquis de lui faire une sorte de cape de laine qui descend de son collier de laine. J'ai également fait sa version de profil histoire que cela ne soit pas trop compliqué de l'imaginer en 3D.



Sur les trois concepts que j'ai proposé à mon groupe, nous avons décidé de prendre l'idée du minotaure en tant que bosse du jeu. J'étais aussi d'accord avec la décision. Ensuite j'ai décidé de le mettre au propre. J'ai de fait le sketch de façon un peu plus. Parmi les détails qui ont changés par rapport

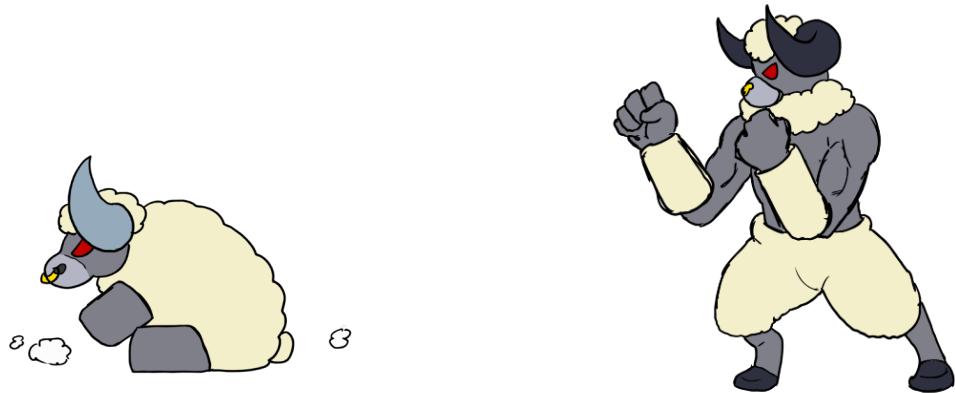
au concept de base qui a été validé, j'ai retiré sa cape que j'ai trouvé plus compliquée à faire que sur un sketch, surtout que ça devait être une cape de laine avec la bonne texture appliquée. Je lui ai également rajouté la clochette dans son collier de laine, idée qui vient du concept de centaure qui n'a pas été retenue. Pour ce qui est de la couleur, j'ai gardé la même palette de couleur que le mob de base.



Après que je m'ètre occupé de concevoir les mobs Tom m'a chargé de réaliser le logo du launcher, en effet, le logo que l'on a depuis le début du projet ne convient pas au format de logo d'application. Mes camarades m'ont conseillé de partir de la première lettre du jeu, à savoir le G. je me suis donc penché dessus, j'ai voulu garder le même style que pour le logo du jeu tout en ajoutant un clin d'œil au mob. J'ai envoyé un premier croquis à mes camarades qui m'ont chacun fait part de ce qui les dérangeait dans le croquis. En écoutant ce que l'on m'a dit j'ai modifié mon sketch et suis revenu vers eux pour leur proposer cette nouvelle version qui a, cette fois, été validée. J'ai donc mis le tout au propre en gardant la même palette de couleur que le logo du jeu. Dans le logo on retrouve les plantes présentes dans le logo du jeu, la laine et la clochette des moutons ainsi que le nom de notre groupe à savoir Tox.



Il faut savoir qu'entre temps j'ai décidé, plus ou moins avec le groupe de ne pas perdre de temps à faire les autres animations de déplacements (haut, bas, gauche et droite). Aussi suis-je passé aux animations que j'ai le plus appréciées faire depuis le début : les animations d'attaques de personnages. L'ami que j'ai déjà beaucoup mentionné a accepté de se charger des animations d'attaque des mobs, qui sont plus proches de son style que mes personnages.

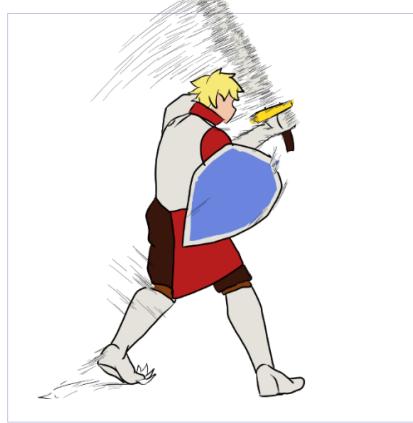


J'ai commencé par l'animation du chasseur qui est celle pour laquelle j'ai fourni le moins d'efforts et qui me plaît le moins. En effet, parce que je ne suis pas fan d'animation quand c'est moi qui dois en faire, je me suis dit que pour

gagner du temps je n'allais faire que le torse qui bouge et les jambes seraient immobiles, mauvaise idée, le résultat n'est pas très satisfaisant je trouve. J'ai donc juste réalisé un tir à l'arc. Pour ce faire j'ai essayé de me rappeler les fois où j'avais tiré à l'arc et les différentes personnes que j'ai observées tirer. Ce qui m'a le plus posé problème dans cette animation c'est l'arc et la flèche. En effet leur placement et la conservation de leurs proportions est vraiment compliqué je le trouve. Après avoir vu le rendu de l'animation précédente j'ai décidé de faire tout le corps à chaque fois. Je me suis donc attaqué ensuite à l'animation d'attaque du paladin. Celle-ci m'a posé pas mal de problème lors de la recherche du mouvement que j'allais animer. Après avoir passé environ une heure à chercher un mouvement sympa à réaliser et à faire des essais infructueux j'ai finalement décider de me créer une référence. Pour ce faire je me suis levé et suis allé chercher un katana pour chercher LE mouvement qui me convaincrait. A la question qui est de savoir si oui ou non je me suis « amusé » à trancher de l'air le katana dans une main et le fourreau dans l'autre la réponse est oui.

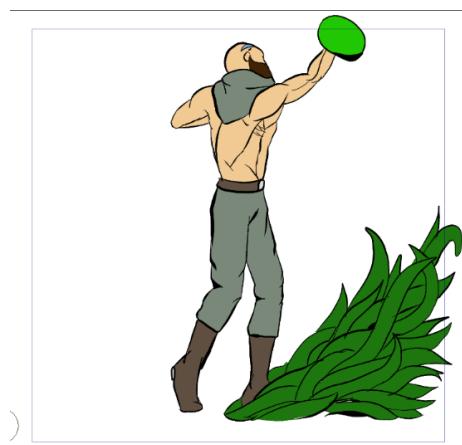
Une fois avoir trouvé le bon mouvement je l'ai répété et répété en le décomposant afin de bien m'en imprégner. J'ai même fini par me filmer pour avoir une bonne référence qui m'a beaucoup servi quand je suis enfin repassé sur CSP. J'ai donc pu faire l'animation du paladin sans trop de difficultés après cela. Je me suis d'ailleurs un peu amusé avec les effets et lignes de vitesse.

Je suis ensuite parti faire l'animation du mage. Pour lui j'ai recherché de nombreux GIF en pixel art pour m'inspirer de leur type d'attaque, j'ai passé moins de temps à rechercher ce que j'allais faire que pour le paladin. Après avoir fait un croquis très simplifié des poses clés pour me donner l'idée et me dirigé je m'y suis lancé. J'ai obtenu un résultat qui m'a beaucoup plu. Je l'ai donc fait en train de s'élevé dans les airs avant de lancer de la foudre par son bâton.



Je suis ensuite passé sur le support. Pour lui j'ai vite pensé à ce que j'allais faire. J'ai également dû me filmer pour celui-là, en plus j'étais torse nu à ce moment-là, c'était bien ridicule mais ç'a été utile. Tom m'a pendant que je faisais le sketch de l'animation qu'il souhaitait voir des ronces invoquées par le personnage. Les ronces ont été marrantes à faire d'ailleurs. J'ai bien aimé le concept d'attaque auquel j'ai pensé pour ce personnage.

Enfin je me suis attaqué à l'assassin. Pour son cas, c'est allé beaucoup plus vite que les autres. En effet, étant un personnage basé sur la vitesse, je l'ai fait disparaître à partir d'un point, trancher deux fois pour finir par réapparaître dans une position différente. Pour lui j'ai pensé à télécharger une brush qui faisait les lignes de vitesse pour aller plus vite et ne pas passer trop de temps à faire de tonnes de lignes pour pas grand-chose.



Enfin je me suis attaqué à l'implémentation des animations de déplacements des personnages. J'ai donc d'abord demandé à Elliott et Hugo de m'expliquer comment fonctionnait la sélection de personnage afin de trouver où est-ce que je pourrais implémenter les animations dans Unity, le code ayant déjà été préparé pendant mes tests. J'ai pu trouver les prefabs dans lesquelles j'ai placés les animations. Cela a été long et répétitif d'avoir à refaire la même chose 5 fois pour 8 types d'animations à chaque fois. Une fois fait j'ai remarqué que les animations n'étaient pas visibles en multijoueur. Grâce à Tom j'ai pu régler sans trop de problèmes ce détail là et c'est ainsi que cela a été réglé.

Je me suis donc ensuite penché sur une « amélioration » de la map. J'ai donc passé une après-midi et une soirée à modifier et remplir la map avec les assets que j'avais à disposition. Je me suis ensuite occupé du donjon sur la map. J'avais en tête de faire un simili labyrinthe mais j'ai n'est pas vraiment eu le courage de le faire, aussi suis-je resté sur quelque chose d'assez classique : une ligne droite vers le boss.

Finalement, à mon grand regret je n'ai pas réussi à implémenter les animations d'attaque dans le jeu. Je n'ai pas pu y consacrer suffisamment de temps pour réussir à les implémenter pour le combat. Faute de quoi, j'ai décidé de les implémenter en tant qu'emote. Pour ce faire j'ai eu à rajouter l'animation dans les prefabs et à rajouter un code de trigger pour les déclencher. Je suis heureux d'avoir quand même pu les intégrer dans le jeu.

Pour conclure ce projet m'a beaucoup appris, en effet j'ai passé du temps en appel avec les autres pour suivre un peu comment ils faisaient pour essayer de comprendre également et j'ai donc pu apprendre beaucoup de chose comme ça. Ma conclusion pour ce projet est que j'ai beaucoup appris que ce soit du point de vue du code et du point de vue d'attaque.

3 Conclusion

Pour conclure, nous voudrions maintenant revenir sur ce que nous a apporté ce projet aussi bien en termes de connaissances que de méthodologie. Tout d'abord, commencer un projet de cette ampleur avec nos connaissances semblait un peu être mission impossible, mais puisque nous n'avions pas le choix et que nous ne souhaitions pas réaliser un projet moins ambitieux mais qui nous attirait moins, nous nous sommes lancés dans la création d'un jeu semblable à des jeux comme Wakfu ou Dofus qui sont des MMORPG, à savoir des jeux dont la conception prend de longues années avec de très grosses équipes composées de personnes expérimentées, tout ce que nous n'étions pas. La seule arme que nous avions au début était au final notre ignorance du travail colossal qui nous attendait, ce qui nous a permis dans une certaine mesure de ne pas nous retrouver décontenancé par le travail à effectuer et de nous concentrer sur les tâches essentielles une à une.

Ensuite pour ce qui est de la méthodologie, nous n'avons pas eu d'autres choix que de prendre un bon moment pour trancher quant à l'ordre des tâches à réaliser, et c'est ainsi que nous avons décidé de diviser le travail en le plus de petites tâches unitaires possible pour ne pas avoir de grosses parties à réaliser d'un coup, ce qui aurait pu constituer une véritable perte de temps en rendant le travail très compliqué.

Ensuite pour ce qui est des connaissances, nous avons premièrement appris à utiliser unity. Bien que ce ne soit pas quelque chose qui est assuré de nous servir à l'avenir, la manière dont cet outil fonctionne a certainement su nous faire progresser en POO, notamment grâce au système de gameObject et de composants associés.

De plus, pour ce qui est de la partie réseau, nous avons appris beaucoup de nouvelles choses comme un exemple de comment peuvent dialoguer un client et un serveur, comment synchroniser des données ou même tout simplement comment développer un site internet.

Enfin, deux choses qui ne sont pas à négliger sont la capacité à faire des recherches et à les appliquer et la capacité à assimiler des connaissances trouvées en ligne. En effet, nous n'aurions bien évidemment pas pu accomplir tout ceci sans l'aide d'internet, mais nous savons que les recherches font partie intégrante de nombreux métiers liés à l'informatique, et c'est pour cela que nous n'avons pas hésité à avoir recours à des guides ou tutoriels pour trouver comment faire ce que nous voulions. De plus, ce qui est intéressant est que nous avons petit à petit réussi à assimiler les connaissances que nous

avions pour mettre en place certaines fonctionnalités sans suivre de guides. L'exemple le plus concret est sûrement celui du groupe pour lequel nous avons absolument tout fait de nous-même, sans suivre de guide ou de tutoriel, que ce soit l'interface, les scripts ou même la partie réseau.

Nous avons donc tiré de ce projet beaucoup de choses : des joies, des peines, des connaissances, de l'expérience et il est aujourd'hui sûr que ces quatre mois de travail n'ont pas servi à rien, et chose à laquelle nous ne nous attendions pas forcément, nous comptons continuer de travailler sur ce projet ponctuellement pour le rendre un peu plus complet et le partager avec nos amis, puisqu'en l'état, il peut sembler encore un peu vide malgré tout le travail accompli et tout l'investissement que nous avons pu fournir.