Aegina : rapport de soutenance n° 1

Florian Amsallem (amsall_f) , Théo Issarni (issarn_t),
Julien Mounier (mounie_a), Romain Mounier (mounie_r)

AIM^2

Table des matières

Introduction

1 Univers du jeu

1.1 Aegina

Le monde d'Aegina est constitue d'une quasi infinité d'ile volante se trouvant au dessus d'un vide béant appelé les *abysses*. Se monde contrairement au notre est instable, il ne partage pas nos lois de l'équilibre et de nombreux phénomènes étranges s'y déroule sans que l'on puisse les expliquer la présence des *abysses* n'étant qu'un de ses phénomènes.

La principale source de l'instabilité de ce monde sont les cristaux que l'on peut trouver sur certaines îles. Ceux-ci sont capable des déchirer les trames de l'espace et du temps permettant de réaliser des miracles comme le retour dans le temps ou la téléportation. Malgré toute ces instabilité un écosystème ses tout de même développée dans ce monde. Des animaux comme ceux que l'on connait tel que des lapins, papillons et moutons peuple les îles d'Aegina. Mais d'autres espèces plus agressifs et plus exotiques sont apparue suite à l'instabilité du monde comme les slimes. Un instinct naturel pousse toute ces créatures agressifs à s'attaqué aux cristaux si ceux-ci sont activé pour les empêché de rendre le monde plus instable qu'il ne l'est actuellement.

On pourrait donc penser que sans personnes pour activer les cristaux et avec une telle armée de créature près à défendre ce monde celui-ci aurait trouver une certaine forme d'équilibre. Mais c'étais sans compté la présence d'envahisseurs, d'être extérieur à ce monde qui utilise celui-ci à des fins barbares, les êtres humains. Bien que ne vivant pas directement dans ce monde et n'ayant aucune information sur celui-ci les êtres humains on trouvé un moyen de « téléporter »leurs déchets dans un néant qui n'est autre que Aegina. Lorsque les déchets sont seulement des restes et que ceux-ci tombent directement dans les *abysses* cela ne rend pas Aegina plus instable, au contraire cela peut créer un apport de ressource à ce monde si ils ne tombent pas directement dans les *abysses*.

Mais certaines autorités humaines utilise se moyen pour se débarrasser de déchets plus embêtant : d'autres humains. Malheureusement une fois dans le monde d'Aegina ceux-ci tente désespérément de survivre et ils font alors appelle aux cristaux et à la pouvoir quasi divin leur permettant d'échapper à la faim et à la mort. certains voyagent alors d'île en île créant quelques infrastructure telle que des ponts pour s'aider, laissant leur trace dans ce monde mais l'amenant petit à petit à sa fin sans le savoir. Heureusement jusqu'a présent ils ont tous finit par périr sous l'assaut des créatures d'Aegina ou par suicide ne pouvant rentrer chez eux mais il arrivera bien un jour ou l'un d'eux emportera ce monde avec lui dans sa chute.

1.2 Ille

Ille est le personnage principale de notre jeux. Il a l'apparence d'un bucheron roux et barbus. En vérité Ille n'est pas son vrai nom mais en arrivant dans le monde d'Aegina Ille a perdu la mémoire et ne s'en souviens donc plus.

1.3 Les minerais

1.3.1 Mithril

Le mithril dans le monde d'Aegina et un minéral vert clair un peu plus solide que le fer. Assez rare sur la surface des îles on en trouve toujours en petite quantité à l'intérieur de celle-ci puisque le mithril et un des deux minéraux nécessaire à la survie d'une île. Le mithril à l'étrange particularité de réagir avec les phénomènes étranges d'Aegina parfois les amplifiants d'autres fois les stabilisant. Il est impossible de savoir comment celui-ci va réagir avant d'avoir testé.

A l'état naturel il catalyses ces instabilités et les relâchent sous formes de vagues d'énergie rendant le minéral chaud au touché. Une fois épurée le mithril continue de catalyser les instabilités mais peut relâcher son énergie sous de nombreuses autres formes que la chaleur comme le ferait un cristal mais à bien plus faible échelle.

D'ailleurs l'énergie catalysé et stocké par le mithril semble être la même que celle se trouvant dans les cristaux car les deux matériaux peuvent échanger cette énergie entre eux. Ainsi quelques chercheurs hérétiques qui se sont trouvés sur Aegina on supposés que les cristaux était formé d'un alliage composé en majorité de mithril. Ces chercheurs ont trouvé la mort quelques jours plus tard tués par des sangliers après avoir déchargé leur cristal à l'aide de mithril.

1.3.2 Floatium

Le floatium et un minéral quasi transparent que l'on peut trouvé dans Aegina. Les humains qui arrive dans Aegina le confonde souvent avec du quartz lorsqu'ils en voient dans la roche mais celui-ci est bien plus transparent et s'apparenterait plus à du verre ou de la glace. La vrai particularité du floatium se remarque une fois le minerai épurée de toute impureté. La première constatation et que celui-ci est bien plus solide que ce à quoi on peut penser, sa dureté égale celle du fer, mais surtout il est impossible de le faire tombé par terre. En effet une fois celui-ci lâché il flotte.

Ce minéral est encore un mystère d'Aegina car personne ne sait pourquoi il flotte. Il n'est pas plus léger que l'air puisqu'il reste à une distance fixe du sol mais il ne semble pas s'agir

non plus de force magnétique. De toute façon peu d'expérience on eu lieu sur le floatium car ,dans Aegina, en plus d'être un minerai très rare, les personnes capable de faire des expérimentations scientifique dessus le sont encore plus.

La seule information sur est que le floatum est, avec le mithril, l'un des deux composants permettant aux îles de voler. Cependant les cristaux et le floatium peuvent réagir entre eux , le cristal semble pouvoir enlever ces capacités au floatium tandis que le floatium est responsable de la rotation et de la lévitation du cristal, mais tout cela reste de la théorie.

Néanmoins il est possible de conjecturer que si un cristal à une trop grande influence sur une île il pourrait provoquer sa chute. Mais les seuls personne connaissant la vérité sur cette hypothèse sont surement mort.

1.3.3 sunkium

Le sunkium et un minéral d'une couleur mauve très sombre. C'est un minerai légendaire qui renferme de nombreuse qualité car il est le minerai le plus rare, le plus lourd et le plus sombre, le plus solide, le plus cool, le plus classe, le plus *dark* et le plus unique d'Aegina. Pourtant malgré sa solidité il ne pas si difficile à extraire des roches car il semble repousser les autres minerai lors de sa formation ce qui permet de le trouver dans des alvéoles formé par la roche. Cependant une fois extrait son poids reste tout de même un problème pour le transporter.

Mais ce qui fait de ce minerai le minerai le plus cool est qu'il peut être utiliser très simplement pour la création d'épée ou d'armure. En effet ce minerai nécessite la même température que le fer pour fondre et une fois mélanger avec du mithril il chauffe et refroidie à une vitesse exceptionnelle. De plus la couleur sombre des armures formé avec le sunkium apporte un coté cool et *dark* à toute personne les utilisants. Mais malheureusement cet impression ne reste que quelques secondes car rare sont ceux capables de se mouvoir avec une telle armure, le plupart se retrouvent coincé dans celle-ci incapable de bouger ou de l'enlever amenant à la mort la plus stupide d'Aegina.

Le sunkium, comme le mithril et le floatium, réagit avec les cristaux mais sa réaction est peut être la plus impressionnante des trois, après tout c'est le minerai le plus cool. Le sunkium change de taille et de masse sous l'influence du cristal mais pas comme tout le monde s'y attendrait car plus le sunkium devient petit plus il devient lourd. Il se met alors à absorber de la lumière assombrissant les alentours. Le cristal quand à lui peut absorber le sunkium mais personne ne sait réellement ce que ça lui apporte.

Certaine théorie stipulerait que les pouvoir du cristal viendrait de minuscules morceau de sunkium qui déformerait l'espace temps ce qui permettrait d'expliquer tout les pouvoirs

du cristal. Mais cette théorie s'appuyant sur d'autres théories qui n'ont pas été prouvé sa véracité est plus que discutable. Sans la présence des cristaux le sunkium serait aussi le minerai le plus mystérieux d'Aegina.

2 Réalisation

2.1 Pré-janvier

2.1.1 Cycle jour/nuit (Julien)

Dans un premier temps, on a du déterminer la durée d'une journée que l'on a subdiviser en 30 phases. Ces phases servent à changer d'image de skybox. Seulement cette méthode à causé quelque problèmes. Entre autre, le changement de skybox était trop brusque et les différentes skybox n'étaient pas homogènes. C'est pourquoi nous avons du trouver une autre méthode : La *directional light*.
La *directional light* est un gameobject propose par unity. Son intérêt majeur est que l'orientation de cette lumière influence la skybox du monde. De plus nous pouvons choisir la couleur de cette lumière. Pour ce faire nous avons utilise une fonction qui converti la longueur d'onde des rayons lumineux en couleur RGB et nous avons représente le spectre solaire a l'aide d'une fonction mathématique.

$$255((3.25t - 4)x^2 + (-4t + 4)x)$$

2.1.2 Inventaire (Romain et Florian)

Pour que notre personnage puisse posséder un grand nombre d'objets de toute sorte et se déplacer avec facilement nous avons créé un inventaire. Celui ci est constitue de 24 cases sous forme de 4 lignes et 6 colonnes. Chaque case peut contenir un élément, qui peut être déplacé grâce a un *drag and drop*. De plus le joueur peut avoir une description de chacun des éléments en passant sa souris sur celui ci. Cet inventaire s'ouvre lorsque l'on appui sur la touche *inventaire* (touche *i* par défaut)et peut ensuite être ferme avec la même touche *inventaire* ou la touche *escape*.

Lorsque l'inventaire est ouvert le joueur et la camera ne peuvent plus bouger via les touches de déplacement ou la souris (mais le joueur continuera de tomber si celui-ci était entrain de chuter). L'inventaire se sauvegarde automatiquement en local lorsque le jeu est quitté normalement ou lorsque qu'une modification de son contenue s'effectue (quand un objet est jeté ou récupéré). Ce même inventaire est alors récupéré d'une partie sur l'autre.

2.1.3 Animation 3D (Théo)

Ille est un personnage que nous avons télécharger depuis l'asset store qui possédais de base deux animations, "marcher" et "rester immobile" cependant ces deux seules animation ne nous suffisais pas. A partir de ce moment nous avons décidé de faire les animations des nos

personnage nous même. Le premier problème que nous avons rencontre : notre personnage n'était pas accessible depuis notre version de blender, trop récente pour le modele 3D que nous avons. Il nous a donc fallut trouver une ancienne version de blender compatible avec notre model 3D. Depuis nous essayons d'améliorer nos animation dans le but d'être le plus réaliste possible mais s'en expérience dans ce domaine il est difficile de faire des animations convenables et celle-ci sont systématiquement remise en question et modifié.

2.1.4 Déplacement du personnage et de sa camera (Théo et Julien)

Nous avons permis à notre personnage de se déplacer en fonction de ce que l'utilisateur souhaite faire. Celui-ci peut avancer (avec la touche "w" ou "up"), reculer (avec la touche "s" ou "down"), s'orienter vers la droite (avec la touche "q" ou "right") et s'orienter vers la gauche (avec la touche "d" ou "left"). Notre personnage peut courir si ce dernier se déplace et que l'utilisateur appuie sur la touche "shift", il peut aussi sauter si l'utilisateur appuie sur la barre d'espace.

Le saut à d'ailleurs été un élément problématique à implanté car notre personnage ne devait pouvoir sauter que quand celui-ci est sur le sol pour éviter qu'il s'envole en enchainant des sauts dans les airs. Nous avons donc créer des restriction permettant au personnage de ne sauter que depuis certain éléments du décors (qui regroupe presque tout les éléments).

A partir de ce moment notre personnage était devenue opérationnel mais à quoi cela servirait il si nous ne pouvions pas le voir ?

Nous avons alors décider de créer une camera qui suivrais notre personnage où qu'il aille. Cette caméra n'est pas une simple caméra fixe, elle possède des caractéristiques variées grâce auxquelles une grande liberté est accordé au joueur. Celle-ci s'oriente grâce à la souris et même si on dit qu'elle suit le joueur l'orientation des mouvements du joueur ne dépendent que de l'orientation de la caméra par exemple faire reculer le joueur le fait en réalité se diriger vers le caméra.

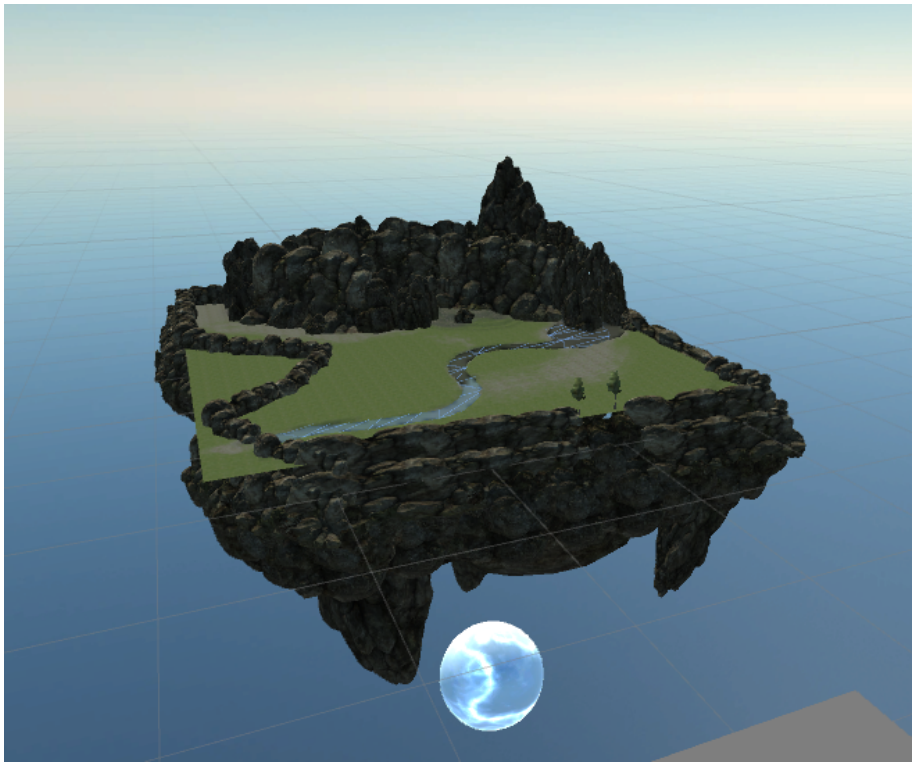
A la troisième personne le joueur peut vois son personnage de cotés, de devant ou encore de derrière tout cela en étant en étant proche ou éloigné de son personnage. cette éloignement de la caméra se contrôle avec la molette de la souris il est donc malheureusement indispensable de posséder une souris à molette pour profiter à 100% de la caméra (le trackpad d'une ordinateur portable peut lui aussi permettre de profiter de la liberté de la caméra).

A la première personnage notre camera permet au joueur de voir exactement ce que son personnage voit. Cette vue est accessible en rapprochant la caméra du joueur assez près de celui-ci en troisième personne et elle se quitte dés que l'utilisateur tente d'éloigné la caméra du personnage.

2.1.5 Models 3D (Julien)

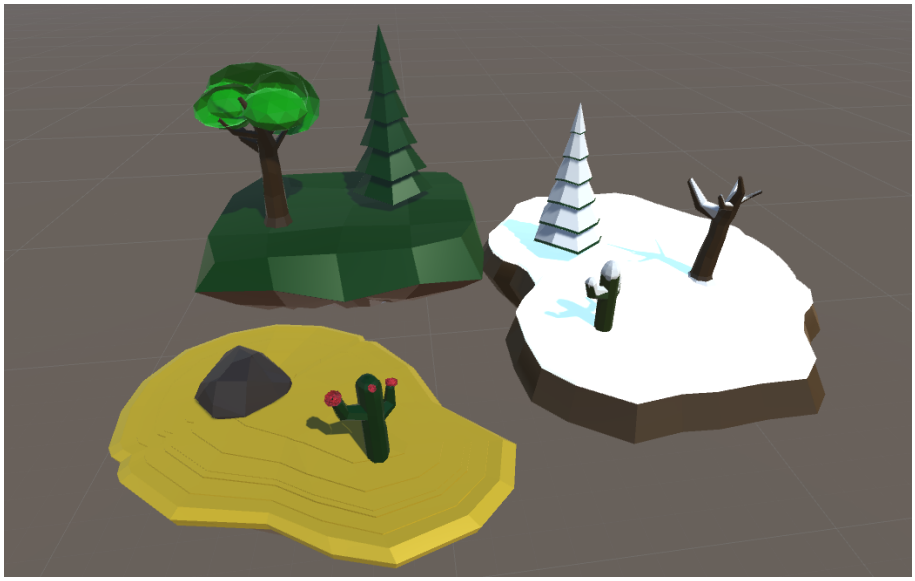
Nous avons décidé de doter notre jeu d'un style graphique original. Nous avons cherché l'inspiration en parcourant l'assetstore. Nous y avons trouvé le modèle de notre personnage principal et nous nous sommes accordés pour un univers cartoon.

Nous avons donc commencé la réalisation de notre première île mais ce fut un échec. Nous n'avions pas assez bien définie la forme que nous voulions lui donner et il a donc fallu recommencer le modèle car celui-ci n'était pas vraiment cartoon et n'était pas adapté à la génération procédurale du monde.



première île

Nous avons alors décidé de réaliser tous nos future modèle grâce au logiciel Blender et d'importer le moins possible depuis l'assetstore. Nous nous sommes donc recentré sur l'aspect cartoon et en avons profité pour réaliser nos premiers éléments à savoir les arbres et un rocher.



premiers élements

La réalisation des modèles suivants s'est fait et continuera à se faire petit à petit en fonction des demandes engendrées par l'avancement du projet.

2.2 Janvier

2.2.1 Barre de selection (Florian)

Lorsque nous sommes entrain de bouger il est impossible de voir ce que contient notre inventaire et il faut nécessairement l'ouvrir, et donc s'arrêter, pour interagir avec lui. Actuellement ce n'est pas un problème mais quand il sera nécessaire d'utiliser des consommables pour survivre nous voulons que ceci soit facile d'accès et ne nécessite pas de s'arrêter (pour que l'on puisse boire une potion de santé alors que l'on se fait courser par un slime par exemple).

Nous avons donc ajouté une barre de sélection synchronise avec la 4^{eme} ligne de notre inventaire. Le joueur peut sélectionner une case de cette barre en utilisant les touches alphanumériques correspondantes et utiliser l'objet s'y trouvant. Cette barre apparait même quand le joueur n'est pas dans son inventaire puisque son but est justement d'utiliser des objets hors de l'inventaire.

2.2.2 Menus (Romain et Florian)

Un élément essentiel à tous jeu et indéniablement de pouvoir le quitter et le commencer. Cela paraît évident mais il a tout de même fallut créer des interfaces permettant ces actions au début du jeu (pour commencer une partie) et au milieu du jeu (pour arrêter de jouer).

pour cela nous avons créé des menus permettant non-seulement de démarrer une partie ou de quitter le jeu mais aussi de modifié de nombreuses option.

Actuellement le menu permet de modifier la langue et le volume sonore du jeu. Ces menus s'ouvre avec la touche *escape* quand aucune autre interface n'est ouvert et ce ferme avec la touche escape ou avec le bouton *continuer*. Dans le menu le joueur et la camera du joueur ne peuvent pas bouger comme lorsque l'inventaire et ouvert. Lors du lancement du jeu on arrive directement à un menu mais celui ne se ferme pas avec la touche *escape* par contre il est possible d'utiliser le raccourci pour accéder aux options.

L'apparence graphique des menus à longtemps était discuté. Un rectangle simple coloré comme l'inventaire ne plaisait pas donc nous avons essayé de mettre comme fond un parchemin. Cependant le placement des écritures sur ce parchemin ne paraissait pas centré et celui-ci pouvait prendre des formes bizarres lorsque la dimension de la fenêtre était changé. Nous avons finalement décidés de ne pas mettre de fond au menu et seulement ajouté des boutons dans le vide ce qui est finalement très acceptable esthétiquement.

2.2.3 Class *Item* (Romain)

Comme dit précédemment dans notre jeu il existe un inventaire et celui-ci contient des objets. Seulement ces objets n'existé pas directement dans unity et leur comportement non plus. Il a donc fallu les créer grâce à une class que l'on a tous simplement appelé *Item*.

La classe *Item* permet de créer un patron d'objet ne contenant que des constantes immuable. Par exemple le bois un objet qui possède deux noms (un français et un anglais), deux descriptions, un identifiant pour le reconnaître, une quantité maximal, une texture 2D un élément correspondant à l'objet lorsqu'il est lui même dans le monde 3D. Cependant Lorsqu'un objet se trouve dans l'inventaire il peut se trouver en plus grande quantité qu'un exemplaire unique. Les objets présents dans l'inventaire sont donc une nouvelle class appelé *ItemStack* contenant une quantité (inférieur à la quantité max de l'objet) et un pointeur vers l'objet qu'il représente. Ainsi Il y a un grand gain de performance et la modification de la quantité d'un objet dans l'inventaire ne change pas la quantité de tous les objets du même type.

2.2.4 Multi-joueurs (Florian)

Le multi-joueurs est une part importante du projet car celle-ci permettra la coopération des joueurs pour progresser dans le jeu. Nous avons dans un premier temps utilise les fonctionnalités de *Unet* pour pouvoir créer un serveur (ou un host dans notre cas), mais aussi pouvoir rejoindre un serveur grâce a son adresse ip.

Nous avons dans un second temps synchronise la position, la rotation et les actions des joueurs. Pour cela on a d'abord utilise le composant d'unity *NetworkTransform*, mais celui-ci ne permettait pas la synchronisation fluide des personnages. Du coup nous avons codé nous-mêmes un script permettant la synchronisation des personnages. De plus pour rendre les déplacements le plus fluide possible nous utilisons le principe d'interpolation.
Pour finir nous avons synchronise les éléments, les sons et les caractéristiques du monde, que ce soit les iles, les biomes, les bruits de pas des joueurs, l'état de la journée (jour/nuit) ou bien les arbres.

2.2.5 Son (Théo et Romain)

Afin de ne pas avoir un jeu fade, nous avons introduit quelque sons. Ces sons nous permette d'avoir une base que l'on pourras enrichir au fur et à mesure que notre jeu avancera. Nous avons commencer par ajouter un son pour l'ouverture et la fermeture de notre inventaire, mais aussi lorsque nous interagissons avec notre menu. Par la suite nous avons ajouté des bruit de pas afin d'entendre notre personnage marcher ou courir, par manque de réalisme nous avons synchronisé ces sons afin que les autres joueurs jouant avec nous les entendent aussi.

2.3 Février

2.3.1 Génération aléatoire (Florian et Julien)

Comme nous l'avions spécifié dans notre cahier des charges, le monde du jeu est généré de manière procédurale. Pour cette première soutenance, nous avons implémenté la génération aléatoire d'un chunk, et la liaison des chunks entre eux. Nous avons donc au lancement d'une partie un monde composé de quatre chunks chacun relies par des ponts.
Nous avons créé 2 formats de chunk différents possédants chacun une ile. Sur ces iles sont posé des ancres qui correspondent a des emplacements d'objets. Cela permet lors de la génération du monde de choisir un biome aléatoirement parmi forêt, désert ou neige, puis de créer sur les ancres un objet cohérent avec le biome. Par exemple dans un biome désert nous trouverons des cactus et des rochers alors que dans un biome foret nous trouverons des arbres et des fleurs.
Nous avons donc tous les éléments pour créer des chunks, il ne restera plus qu'a faire apparaitre les chunks de manière procédurale avec un agencement correct. Puis il sera aussi nécessaire de sauvegarder le monde avec les modifications que les joueurs y ont apportées.

2.3.2 Drop d'Item (Florain et Julien)

Le drop d'un item correspond a son passage de l'inventaire au monde 3D. Pour cela le joueur a la possibilité depuis son inventaire de glisser un tas d'objet a cote de celui-ci. Il peut aussi grâce a une touche, lancer l'objet sélectionné dans sa barre d'outil. Ces actions vont instancier l'objet dans le monde et puis le lancer a quelques mètres du personnage. Dans cet état n'importe quel joueur peut alors ramasser l'objet, du moment qu'il se trouve assez proche de celui-ci. De plus si deux objets identiques sont assez proches alors ils vont se rassembler. Cela permet d'afficher moins d'objet et donc d'améliorer les performances du jeu. D'autre part les objets dans cet état depuis plus d'une minute disparaissent, cela permet encore une fois d'améliorer les performances.

2.3.3 Bibliothèque (Florian, Romain et Théo)

2.3.4 Histoire (Romain et Théo)

2.4 Mars

2.4.1 Intéraction avec l'environnement (Florian et Julien)

2.4.2 Cristal (Romain et Julien)

2.4.3 Chat (Florian)

Comme déjà préciser la coopération est un élément important du jeu, pour cela la communication est primordiale. Nous avons donc ajouté un chat qui permet aux joueurs de s'envoyer des messages sans passer par un logiciel tiers. Le chat ne permet pas seulement de communiquer, il permet aussi de rentrer certaines commandes (communément appelé *cheatcode*), en voici l'énumération :

/**help :** Enumere les commandes avec leurs arguments ;

/**time** <**value**> **:** Met l'heure de la journee a une certaine valeur ;

/**give** <**id**> <**quantity**> **:** Donne une certaine quantite de l'objet correspondant a cet identifiant ;

/**nick** <**name**> **:** Permet de changer son nom ;

/**msg** <**player**> <**message**> **:** Permet d'envoyer un message prive au player indique ;

Certaines de ces commandes sont surchargées, c'est a dire que nous pouvons mettre plusieurs types d'informations en argument, par exemple la commande */time day* est accepte et va mettre le soleil au zénith. D'autres commandes ont des alias par exemple la commande */msg* peut aussi être écrit */m*.

2.4.4 Musique (Théo)

Afin de compléter l'univers que nous essayons de créer à travers notre jeu, nous avons ajouté des musiques de fond. Nous avons ajouté pour l'instant quatre musique à notre jeu, la première est une musique qui se lance dès que l'on arrive sur le menu principal de notre jeu, il s'agit du seul endroit où vous pourrez entendre cette musique qui est consacré à notre menu. Trois autres musique on part la suite étaient ajouté, avec trois thèmes différents qui sont : la foret, le désert et la neige. Pour le moment le choix de la musique écouté se fait aléatoirement.

2.4.5 Survie (Théo)

2.4.6 Alpha (Florian)

La dernière étape avant la soutenance est de partager notre première version, *v1.0-alpha*. Ce partage nous a permis d'avoir des retours de bugs et quelques avis. Les bugs importants rapporte étaient :

- Le respawn du joueur qui parfois ne fonctionnait pas ;
- Un bug graphique au niveau de l'élément proche lorsque la camera était a l'intérieur ;
- La musique qui en multi-joueurs pouvait se superposer avec la musique des autres joueurs alentour.

Par la suite nous avons corrige la totalité des bugs rapportes pour former la version *v1.1-alpha*, la version présenté lors de la première soutenance.

3 Prévision

3.1 Les quatre grand axe

3.1.1 Creation d'objet

3.1.2 Ajout des créatures

3.1.3 Ajout des succès

3.1.4 PVP et Conquête

3.2 Les améliorations

3.2.1 Survie

3.2.2 Génération aléatoire et sauvegarde

3.2.3 Histoire

3.2.4 Environnement sonore

3.2.5 Personnage

Conclusion