

Aegina : rapport de soutenance n° 2

Florian AMSALLEM (amsall_f) , Théo ISSARNI (issarn_t),
Julien MOUNIER (mounie_a), Romain MOUNIER (mounie_r)

AIM²
Aegina



Table des matières

Introduction	3
1 Univers du jeu	3
1.1 Biomes	3
1.1.1 Printemps : forêt	3
1.1.2 Automne	4
1.1.3 Hiver : neige	4
1.1.4 Été : désert	5
1.2 Créatures	5
1.2.1 Sanglier	5
1.2.2 Pampi	6
2 Réalisations	7
2.1 Les améliorations du Gameplay	7
2.1.1 La création d'objets (Romain)	7
2.1.2 Les Créatures (Florian)	8
2.1.3 Le combat et le mode joueur contre joueur (Théo et Romain)	10
2.1.4 Le système de Succès (Julien)	11
2.1.5 Les objets équipables et l'interaction avec l'environnement (Théo et Julien)	13
2.1.6 Amélioration du Cristal (Florian et Romain)	14
2.2 Les grandes améliorations du système de jeu	16
2.2.1 La sauvegarde (Florian)	16
2.2.2 La refonte du Menu (Romain)	17
2.2.3 La refonte de la première scène (Julien)	19
2.2.4 Le tutoriel (Théo et Romain)	20
2.3 Les ajouts mineurs	21
2.3.1 Amélioration des commandes et système d'opérateur (Florian) . . .	21
2.3.2 Refonte Graphique : interface et skybox (Théo, Julien et Romain) . .	23
2.3.3 Amélioration de la mort (Romain et Florian)	24
2.3.4 Ajout de son et d'animations (Théo)	24
2.3.5 Ajout d'île et d'un nouveau biomes (Julien et Théo)	25
2.3.6 Les Stats (Florian et Julien)	26
2.3.7 Bêta (Florian)	26
2.4 Le site web (Théo)	27

2.4.1	la page d'accueil	27
2.4.2	Les actualités	28
2.4.3	Le wiki	28
2.4.4	L'équipe	29
3	Prévisions	30
3.1	Les quatre grands axes	30
3.1.1	Histoire et fin	30
3.1.2	Créatures : plus de créatures et introduction de Boss	30
3.1.3	Un monde infini	30
3.1.4	PVP : refonte du système de combat	31
3.2	Les améliorations	31
3.2.1	Survie : équilibrage	31
3.2.2	Cristaux : influence divine	31
3.2.3	PVP : une vraie fin possible	31
3.2.4	Améliorations en masse : graphisme,animation,son et musique	32
	Conclusion	32

Introduction

Ce rapport contient l'état du développement du jeu Aegina du 20 mars 2016 au 1^{er} mai 2016. Nous détaillons d'abord quelques éléments de l'univers du jeu. Nous expliquons ensuite les différentes améliorations du gameplay et du système du jeu et nous détaillons tous les ajouts faits depuis la première soutenance. Nous présentons également le site web et enfin les dernières modifications du jeu prévues d'ici la fin de son développement.

1 Univers du jeu

Dans le rapport de 1^{ère} soutenance nous avons révélé l'histoire entourant Aegina, Ile, les cristaux et les divers minerais existant dans ce monde. Seulement, le jeu étant encore en début de développement, nous n'avons pas vraiment parlé de l'écosystème existant dans le monde d'Aegina.

Parallèlement au développement du jeu, l'univers d'Aegina s'est enrichi et nous pouvons maintenant parler de ses différents environnements ainsi que des créatures y vivant.

1.1 Biomes

Le climat dans l'univers d'Aegina est très étrange. Il peut changer brusquement d'une zone à une autre mais est stable dans chacune des zones. Cette stabilité a permis à de nombreuses espèces de s'installer et de survivre dans ce monde sans craindre un brusque changement de climat. Les différents climats d'Aegina, au nombre de quatre, sont souvent appelés les quatre saisons d'Aegina pour leur ressemblance aux saisons du climat tempéré de notre monde.

1.1.1 Printemps : forêt

Le printemps d'Aegina est un climat dans lequel les températures sont stables entre 6 ° C et 20 ° C. De plus les pluies ne se font pas rares ce qui permet aux îles du printemps d'Aegina d'accueillir une faune et une flore abondantes et variées.

Ces îles aux apparences tranquilles semblent les zones les plus agréables pour les voyageurs d'Aegina qui cherchent un endroit pour se reposer. Cependant détrompez-vous, les créatures d'Aegina sont très agressives et le printemps d'Aegina est la saison de la « mort » pour les aventuriers qui s'y aventurent.

Les cristaux se trouvant dans le printemps d'Aegina ont très souvent la capacité de rendre la terre plus fertile et d'augmenter le nombre d'animaux et de végétaux dans l'île transfor-

mant ainsi les forêts de ces îles en jungles dangereuses pour quiconque s'y aventurerait. Le printemps d'Aegina serait une véritable mine d'or pour des entrepreneurs cherchant à récolter du bois ou cultiver des plantes mais malheureusement ce type de personne survit rarement longtemps dans Aegina.

1.1.2 Automne

L'automne d'Aegina est un climat où les températures varient entre 2 ° C et 25 ° C. Un fort ensoleillement et d'autres raisons obscures ont permis à une végétation assez particulière de se développer dans ce climat. Cette végétation possède une couleur rouge-orangé donnant l'impression d'un automne constant sur l'île. De plus certaines espèces végétales comme les champignons et les citrouilles sont endémiques de l'automne d'Aegina.

Malgré la présence de créatures plus faibles dans l'automne d'Aegina que dans le printemps d'Aegina, ces zones n'en restent pas moins dangereuses car l'automne d'Aegina échange la densité de créatures contre une plus grande diversité de celles-ci, rendant ainsi la chasse plus difficile même pour des chasseurs aguerris. Ces zones n'en restent pas moins les zones favorites des aventuriers du monde d'Aegina car elles présentent le meilleur rapport « confort de vie »/« danger des créatures ».

Les cristaux se trouvant dans l'automne d'Aegina peuvent avoir n'importe quelle capacité rendant ces zones encore plus variées et uniques qu'elles ne le sont déjà. De plus tous les ans, les aventuriers d'Aegina vont sur ces îles pour y récolter et sculpter des citrouilles en pleurant et en se souvenant de leur monde. Cette pratique vaut à l'automne d'Aegina le nom de « la saison de la mélancolie ».

1.1.3 Hiver : neige

L'hiver d'Aegina est un climat extrêmement froid qui atteint des températures avoisinant les -20 ° C et qui ne dépassent jamais les 0 ° C. Ce Climat rend la faune et la flore sur ces îles peu fréquentes. Etrangement, les îles influencées par l'hiver possèdent une abondance de minerais. Cela serait dû au fait que, en raison de la faible présence d'êtres vivants pouvant absorber les énergies étranges d'Aegina, ces énergies sont absorbées par les roches et transformées en minerais.

Malheureusement la faible présence de créatures dans l'hiver d'Aegina n'en fait pas un lieu de rêve pour les aventuriers car le climat y est tout de même redoutable et la nourriture rare. Seules les rares aventuriers capables de forger y viennent pour récupérer des minerais.

Les cristaux se trouvant dans l'hiver d'Aegina ont de plus fortes chances de posséder des pouvoirs divins capables de briser l'espace et le temps. Ceux-ci peuvent permettre de

se déplacer vers d'autres cristaux se trouvant aussi dans l'hiver d'Aegina. Ce qui est très utile pour les aventurier possédant une base près de ces zones et veulant explorer le monde d'Aegina.

1.1.4 Été : désert

L'été d'Aegina est un climat extrêmement chaud et aride. La température y est toujours au dessus de 30 ° C et atteint régulièrement les 50 ° C. La faune et la flore de l'île sont alors très particulières, elles sont toutes les deux globalement constituées de cactus. Etrangement, c'est aussi l'un des seuls endroits où l'on peut trouver de l'eau car les cactus d'Aegina retiennent une énorme quantité d'eau en leur intérieur.

Le climat de l'été d'Aegina n'est pas vraiment le meilleur pour les aventurier mais la présence de cactus, et surtout de l'eau dont ils regorgent, peut transformer cette zone en un lieu très agréable pour se reposer. Malheureusement quelques créatures rodent dans l'été d'Aegina et celle-ci peuvent profiter d'un moment d'égarement pour se repaître d'aventuriers encore chauds.

Les cristaux se trouvant dans l'été d'Aegina ont la particularité d'être le plus souvent des cristaux de guerre. Ceux-ci renforcent les aventuriers d'Aegina en les rendant capables d'abattre un sanglier d'un seul coup d'épée bien affûtée. Ces aventuriers se retrouvent alors également avec un corps résistant comme l'acier, ce qui leur permet de combattre une armée de monstres sans périr. Malheureusement, l'effet ressenti est bien souvent supérieur à l'effet réel amenant alors les aventurier, surestimant leur force et leur résistance, à une mort prématurée face à une horde de monstres.

1.2 Créatures

Les créatures de l'univers d'Aegina sont toutes des êtres vivants ayant muté à cause de l'influence des cristaux. Cela donne à Aegina un faune unique que ses aventuriers n'apprécient pas forcément.

1.2.1 Sanglier

Le sanglier d'Aegina est une créature présente dans la plupart des biomes. Le seul climat qu'il évite est le froid de l'hiver d'Aegina. Il se porte très bien dans des climats chauds comme l'été d'Aegina où il n'hésite pas à détruire des cactus pour boire.

Contrairement à un sanglier normal, il a plus l'apparence d'un phacochère et se montre très agressif envers les aventuriers d'Aegina. Son régime alimentaire est constitué de tout ce

qu'il peut trouver allant du cactus à la chair humaine et en passant même par les minerais de mithril qu'il affectionne pour des raisons que l'on ignore.

Une particularité que les aventuriers d'Aegina ont trouvée à ces sangliers est qu'ils ont une peur des abysses telle qu'ils n'osent pas s'approcher du bord des îles et n'empruntent les ponts que lorsque leurs ennemis ne se trouvent pas en leur présence de peur que ceux-ci les poussent dans le vide.

Ces sangliers, bien que très dangereux, sont très appréciés des aventuriers d'Aegina car leur chair est un met succulent et peut être facilement cuisinée en de nombreux plats. Ces sangliers sont les meilleurs mets de tout l'univers d'Aegina loin devant les autres créatures qui, parfois, ne peuvent même pas être mangées.

1.2.2 Pampi

Le pampi est un mystère pour tous les biologistes ayant mis les pieds dans le monde d'Aegina. Cette créature vivant dans le désert ressemble à un cactus humanoïde. Personne ne sait si les pampis sont des cactus ayant acquis la capacité de se mouvoir, des créatures prenant l'apparence de cactus pour se camoufler ou des humains ayant été transformés en cactus.

Ces trois hypothèses restent plausibles car le régime alimentaire du pampi reste inconnu et pourrait bien se limiter à un peu d'eau fraîche et de soleil comme les autres plantes. Cependant, les pampis possèdent une bouche et, même si celle-ci ne semble pas servir à manger, elle marque bien l'appartenance du pampi au règne animal.

De plus le comportement du pampi semble confirmer l'hypothèse qu'il a pris l'apparence d'un cactus pour se camoufler car il est d'une nature très peureuse et ne montre ses cotés agressifs qu'en deux occasions : quand il est acculé et n'a plus de possibilité pour s'enfuir et quand on l'appelle pampa. En effet, le pampi est une espèce orgueilleuse qui n'aime pas que l'on déforme son nom.

Enfin la dernière hypothèse vient en partie de ce caractère orgueilleux typiquement humain mais surtout des cris que pousse le pampi, ceux-ci rappelant très fortement des cris humains.

Les aventuriers d'Aegina qui chassent le pampi ont deux règles d'or : ne pas acculer un pampi et ne pas prononcer le mot pampa. Un pampi agressif est l'une des créatures les plus dangereuses d'Aegina, il est capable de propulser ses épines avec la puissance d'une mitrailleuse. Les squelettes que l'on peut trouver dans le désert sont les vestiges d'une espèce détruite par les pampi. Malheureusement pour elle, cette espèce poussait un cri ressemblant à s'y méprendre au mot pampa.

2 Réalisations

2.1 Les améliorations du Gameplay

Les améliorations faites depuis la première soutenance se sont en grande partie concentrées sur le Gameplay. En effet lors de la première soutenance nous avions de nombreuses bases dans notre jeu qui nous permettaient de se déplacer et d'admirer le paysage mais rien d'autre. Il était impossible de réellement jouer et de s'amuser. Les nouvelles améliorations font de notre jeu un jeu à part entière procurant de la distraction et de l'amusement.

2.1.1 La création d'objets (Romain)

Une nécessité en survie, et donc dans tous les jeux de survie, est l'obtention d'outils primitifs pour fabriquer un abri ou chasser. Pour cela, il faut pouvoir créer ces outils. Nous avons donc rajouté un système de création d'objets. Chaque objet peut être créé à partir d'une recette définissant le type et le nombre d'ingrédients nécessaires à sa réalisation. Les objets qu'il est possible de créer se classent en cinq catégories : les établis qui permettant de fabriquer plus d'outils, les ressources de construction qui sont principalement utilisées pour fabriquer des objets plus évolués, les consommables et les armes qui seront présentés plus en détail dans la suite de ce rapport et enfin les armures qui ne sont pas encore implémentées.

La réalisation de chacune de ces recettes nécessite plusieurs conditions. D'abord, il faut posséder les composants nécessaires à la fabrication de l'objet souhaité. Cela paraît naturel, il est impossible de faire une pioche à partir de rien tout comme on ne peut pas faire de ragout sans viande. Il faut aussi posséder une place vide dans l'inventaire. Même si le joueur possède déjà les composants de base de l'objet qu'il veut fabriquer, la transformation d'un bâton en une pioche ne se fait pas de manière directe. Il faut d'abord créer un manche et forger un fer de pioche puis assembler les deux. Cet espace vide dans l'inventaire est nécessaire pour toutes ces actions. Enfin, il faut se trouver à proximité des établis nécessaires à la fabrication des objets souhaités. Aucune indication n'est donnée pour savoir quel établi est nécessaire pour faire un objet mais cela reste évident : il faut par exemple être à proximité d'une forge pour fabriquer des lingots ou un autre objet en fer.

Mais malgré tout cela, il manque une chose pour pouvoir fabriquer facilement des objets : l'expérience. Réaliser une recette avec succès ne se fait pas dès la première tentative, fabriquer du verre ou forger une épée sans expérience demande plus de ressources que nécessaire. Tant que la recette n'est pas maîtrisée, le coût en ressources sera doublé. Pour maîtriser une recette il n'existe actuellement qu'une seule possibilité : La chance.

Enfin certaines recettes sont cachées et il est impossible de savoir qu'elles existent sans

remplir certaines conditions ou rechercher sur le wiki.

En jeu, il est possible d'accéder au système de craft en même temps que l'inventaire. Sur la partie gauche de l'écran cinq boutons permettent de choisir quelle catégorie d'objets on veut créer (entourée en bleue sur la figure 1). Après avoir cliqué sur ce bouton, une liste d'éléments pouvant être créés apparaît (entourée en rouge sur la figure 1). Les flèches en haut et en bas permettent de faire défiler les éléments. En sélectionnant sur l'un de ces éléments sa recette s'affiche sous la forme d'une liste de composants (entourée en vert sur la figure 1). Dans cette liste, la dernière case est cochée en vert si l'objet peut être créé, ou occupée par une croix rouge dans le cas contraire. La création d'un objet se fait en cliquant sur la case cochée en vert.



FIGURE 1 – Interface de la création d'objet

Pour permettre au joueur d'améliorer les capacités de son personnage, nous lui donnons la possibilité de réaliser des potions, qui sont bien sûr des consommables.

2.1.2 Les Créatures (Florian)

Chaque type de créature est défini par un ensemble de caractéristiques permettant de les générer dans les biomes adaptés, de récupérer des ressources lors de leur mort, ... Voici l'ensemble de ces caractéristiques :

- Une probabilité d'apparition ;

- Des dégâts ;
- Une vision d’agro ;
- Une vision de fuite ;
- Une vitesse de déplacement en marche ;
- Une vitesse de déplacement en course ;
- Une vitesse d’attaque ;
- Une liste de biomes sur lesquels la créature peut apparaître ;
- Un ensemble d’objets que la créature peut produire à sa mort ;

L’arrivée des créatures dans le monde d’Aegina a imposé la création d’intelligence artificielle, leur comportement devant être automatiquement géré par le jeu.

Pour gérer les déplacements des créatures, nous avons besoin de connaître les chemins possibles et donc la topographie des îles. Nous avons commencé par la génération de graphes lors du chargement des chunks. Chaque chunk possède son graphe qui contient un certain nombre de noeuds comportant tous plusieurs informations telles que la présence d’éléments, par exemple un arbre ou un rocher, et les liens entre ces éléments (voir figure 2).

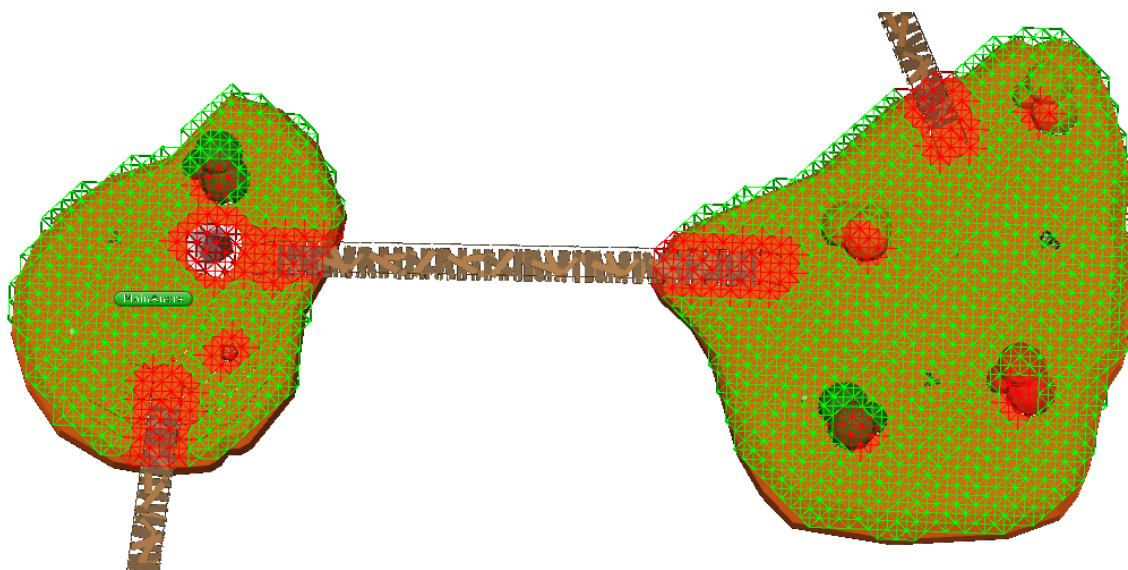


FIGURE 2 – Graphe

Une fois le graphe généré pour chaque chunk, nous avons pu coder un algorithme de *path finding*, le A^* . Ceci nous permet de trouver le meilleur chemin pour aller d’un point A à un point B du graphe. De plus les îles séparées par du vide ou un pont ont des graphes disjoints, cela permet entre autre que les créatures restent sur leur île.

Une fois le déplacement implémenté, nous nous sommes occupés de l’appliquer aux créatures. Nous avons la possibilité de déplacer nos créatures intelligemment en tenant compte de leurs

caractéristiques. Pour déterminer le comportement d'une créature, poursuite du joueur, fuite ou attaque, nous tenons uniquement compte de la distance entre la créature et le joueur le plus proche pour le moment. Tant qu'une créature est loin de tout personnage, elle va se promener en choisissant aléatoirement un point de destination sur le graphe (si le point de destination choisi n'est pas accessible, un nouveau choix aléatoire est réalisé). Dès qu'un joueur est trop proche alors la créature va fuir (le point de destination sera éloigné du joueur). Cependant si le joueur est très proche ou s'il l'attaque, la créature va elle aussi chercher à l'attaquer (ou se défendre). Elle poursuit alors le joueur en évitant tous les obstacles. Une fois à portée de celui-ci, elle va l'attaquer. On peut noter que cette implémentation va permettre de varier les comportements en fonction des créatures en utilisant un seul script d'IA. Par exemple, les sangliers vont pouvoir attaquer sans fuir dès qu'un joueur approche alors que le pampi fuira et n'attaquera le joueur que s'il est agressé. Nous pouvons aussi imaginer des créatures futures qui ne feront que fuir sans jamais attaquer le joueur.

2.1.3 Le combat et le mode joueur contre joueur (Théo et Romain)

Avec l'ajout des créatures, il nous faut aussi le moyen de nous défendre. C'est pour cela que nous avons donné à Ille le moyen d'attaquer. En étant immobile ou en se déplaçant le joueur peut, en utilisant un clic gauche, déclencher l'attaque de Ille. Celui-ci va alors porter un rapide coup en avant avec l'objet dans ses mains ou à main nue si il ne possède pas d'objet. Cependant il faut faire attention, si Ille saute, déclenche un menu ou utilise un objet pendant les deux premiers dixièmes de seconde après le début de son attaque, celle-ci sera annulée. De plus, Ille n'est pas un barbare et cela ne sert à rien de cliquer le plus vite possible avec la souris car Ille ne peut déclencher une attaque que tous les huit dixièmes de seconde même si la précédente attaque a été annulée.

Avec l'ajout de la capacité d'attaque deux nouvelles valeurs sont mises en jeu : l'attaque et l'armure. Ille possède une attaque prédéfinie qui est influencée par l'arme qu'il possède grâce à la formule :

$$DommageDeL'attaque = AttaqueDuJoueur \times CoefficientD'attaqueDeL'arme$$

L'arme du joueur a donc un grand impact sur les dégâts qu'il inflige.

Cependant les dommages d'une attaque ne sont pas directement de la vie perdue par la créature ou le joueur attaqué car celui-ci peut utiliser son armure pour se défendre. La vie perdue suit alors la formule :

$$ViePerdue = DommageDeL'attaque \times \frac{1}{1 + Armure * CoefficientD'armure}$$

Lorsqu'il s'agit d'une créature, son *CoefficientD'attaqueDel'arme* est égal à 1 et son *Armure* est égale à 0. Pour rendre des créatures plus résistantes que d'autres, nous augmentons leurs points de vie.

En même temps que le combat, un mode joueur contre joueur a été ajouté. Lors du lancement d'une partie il est possible de choisir de jouer en mode « joueur contre joueur ». Les joueurs qui se connectent sur le serveur vont alors être séparés en plusieurs équipes. Ces joueurs auront alors des points de réapparition différents et pourront attaquer les joueurs qui n'appartiennent pas à leur équipe. De plus ils pourront attaquer les cristaux des autres joueurs pour les leur voler. Ainsi ils profiteront des avantages de ces cristaux et les joueurs adverses seront affaiblis.

2.1.4 Le système de Succès (Julien)

Les succès représentent les exploits réalisés par les joueurs, ils sont propres à une partie et sont donc partagés entre les joueurs qui peuvent coopérer pour réaliser un succès. Notre jeu laissant beaucoup de liberté au joueur, il est possible que celui-ci ne sache plus quoi faire. C'est là que les succès interviennent, car ils proposent différents défis auxquels le joueur peut se confronter. Ces exploits peuvent porter sur différents aspects de la survie comme le nombre de créatures tuées ou encore la quantité de potions consommées.

D'un point de vue technique, tous les différents succès sont stockés comme attribut static d'une classe particulière. Mais ces succès étant accomplis par les joueurs à n'importe quel moment, il fallait trouver un moyen de mettre à jour ces attributs à chaque instant tout en y passant le moins de temps possible. Les mises à jour doivent donc répondre à quelques contraintes :

- ne pas tester les succès déjà accomplis ;
- ne pas tester les succès dépendant de succès non accomplis (l'objectif *Tuer 10 monstres* ne peut pas être réussi avant l'objectif *Tuer 1 monstre*) ;
- indiquer au joueur le moment où il réussit un succès.

Pour régler ces premiers problèmes, nous avons opté pour l'utilisation de deux structures. Tout d'abord, un nouvel objet appelé *Succes* qui contient : un booléen indiquant si le *Succes* est réussi, la liste de toutes les conditions nécessaires à l'accomplissement de celui-ci, son nombre de parents et une liste de *files*. Ces deux derniers éléments permettent de créer une hiérarchie entre les différents succès et de faciliter l'affichage.

La deuxième structure est une liste de succès à mettre à jour, présente uniquement sur le serveur. A intervalles de temps réguliers, le serveur teste si les succès présents dans la liste sont accomplis. Si c'est le cas, alors il est retiré de la liste à tester et tous ses fils y sont ajoutés

s'ils n'étaient pas déjà présents. De plus, l'affichage d'un pop-up en haut à gauche de l'écran est demandé sur chaque client attestant de l'accomplissement du succès (voir figure 3). La demande d'affichage est mise dans une file pour que les pop-up ne se superposent pas.

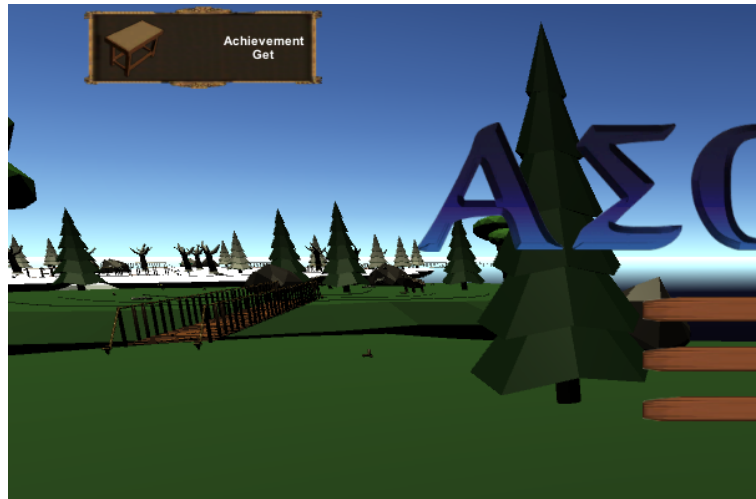


FIGURE 3 – Affichage du Pop-up



FIGURE 4 – Interface des succès

Une interface est également présente pour permettre aux joueurs de contempler les succès qu'ils ont accomplis (voir figure 4). Dans cette interface, chaque succès est représenté par une image et leur hiérarchie est visible grâce à des lignes. A terme, une si petite fenêtre

ne sera pas suffisante pour afficher tous les succès, c'est pourquoi des barres de défilement seront présentes à droite et en dessous de la zone d'affichage (celle de droite est déjà visible).

La distinction entre les succès accomplis, les succès présents dans la liste (pouvant être accomplis) et ceux ne pouvant pas l'être se fait par l'utilisation de trois *Shader*. Comme expliqué dans le premier rapport de soutenance, un *Shader* indique à la carte graphique comment calculer la couleur de chaque pixel de l'écran.

Le premier *Shader* affiche l'image du succès sans la modifier (*succès accompli*), le deuxième ne tient compte que de la transparence de l'image et affiche un gris là où l'image n'est pas transparente (*succès pouvant être accompli*) et le dernier n'affiche pas du tout l'image (*succès impossible à accomplir pour le moment*).

2.1.5 Les objets équipables et l'interaction avec l'environnement (Théo et Julien)

Dans cette partie nous expliquons toutes les méthodes que nous avons mises en place pour permettre au joueur de visualiser les conséquences de ses actions sur l'environnement.

Pour commencer nous avons fait la distinction entre deux catégories d'objets, les premiers qui ne sont que des ressources et les seconds qui peuvent être utilisés et ont un effet particulier soit sur l'environnement soit sur le joueur lui même. Ces derniers objets peuvent être équipés, c'est-à-dire mis dans la main du joueur. Pour se faire, il suffit que l'objet soit placé dans la barre d'outils et que le sélecteur soit sur la case du dit objet. Ainsi, l'objet apparaîtra dans la main du personnage et tous les joueurs pourront le voir.

Une fois un objet équipé, il suffit de faire un clic droit pour l'utiliser. Il existe plusieurs types d'interaction :

- **Les consommables** : qui lors d'un clic droit sont consommés et produisent un effet sur le joueur ;
- **Les outils** : qui permettent d'interagir avec certains éléments de l'environnement ;
- **Les objets pouvant être posés** : qui une fois équipés, apparaissent dans le monde et peuvent être posés avec le clic droit.

Commençons par les consommables. Leurs effets sont très variés, ils peuvent influencer sur les caractéristiques d'un personnage, allant de sa vitesse jusqu'à sa santé en passant par sa hauteur de saut. Ce sont les consommables qui permettent entre autre de se nourrir et de boire.

Passons ensuite aux outils. Jusqu'à présent, il n'était possible d'interagir qu'avec les cristaux et à condition d'être suffisamment proche d'eux. Maintenant, il est possible d'interagir avec n'importe quel élément de l'environnement à condition d'être équipé d'un outil appro-

prié. Il est en effet impossible de couper un arbre à la main, il faut une hache pour cela. Si l'élément est trop éloigné, le joueur peut garder son clic droit enfoncé, son personnage va alors s'approcher des ressources l'entourant et les récolter jusqu'à ce qu'il n'y ait plus de ressources dans les environs ou que son outil équipé ne soit pas approprié à la récolte.

Enfin, pour ce qui est de la pose des objets, lorsqu'un objet pouvant être posé est équipé, celui-ci apparaît en transparence dans le monde. Cet objet transparent est situé à une unité de distance devant le personnage et est tourné vers lui. Mais ce n'est pas l'objet réel, c'est une prévisualisation de l'objet qui permet au joueur de choisir où il veut réellement le poser. Un indicateur de couleur permet d'informer le joueur sur la validité de la position actuelle de l'objet (voir figure 5). Il n'est pas possible de poser une table à travers un arbre ou sur un rocher. Le joueur peut confirmer la pose de l'objet par un simple clic droit.

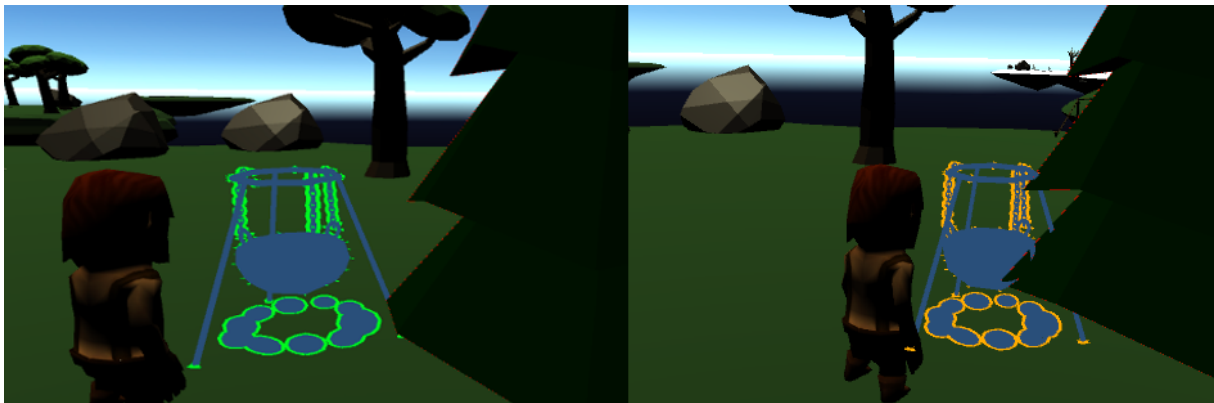


FIGURE 5 – La prévisualisation d'un objet

2.1.6 Amélioration du Cristal (Florian et Romain)

Par rapport à la première soutenance, les capacités du cristal ont été énormément améliorées. Celui-ci a maintenant une place importante dans le jeu.

La première amélioration que l'on remarque sur le cristal est son interface qui est devenue bien plus garnie.

Dans le cadre bleu de la figure 6 se trouvent les informations sur le type du cristal et ses pouvoirs. Les pouvoirs du cristal se distinguent en trois parties :

- le pouvoir de guerre qui renforce la puissance du joueur. Si un joueur se trouve sur l'île, il bénéficie d'un bonus en attaque et en armure significatif. En effet si le pouvoir de guerre d'un cristal est maximisé, les joueurs sur l'île verront leur attaque multipliée par deux par rapport à un joueur sans cristal. De plus, ils réduiront également de moitié tous les dégâts qu'il recevront ;

- le pouvoir de la récolte qui renforce la production du joueur. Si un joueur sur l'île tente de fabriquer un objet, il aura plus de chance de maîtriser une recette. De plus les ressources autour de ce genre de cristal auront plus de chance de réapparaître ;
- le pouvoir divin qui a une influence plus tactique sur les joueur. Actuellement ce pouvoir permet juste de désigner un cristal comme un point de réapparition possible pour le joueur. Plus tard, il permettra également de se téléporter vers les autres cristaux possédant un pouvoir divin.

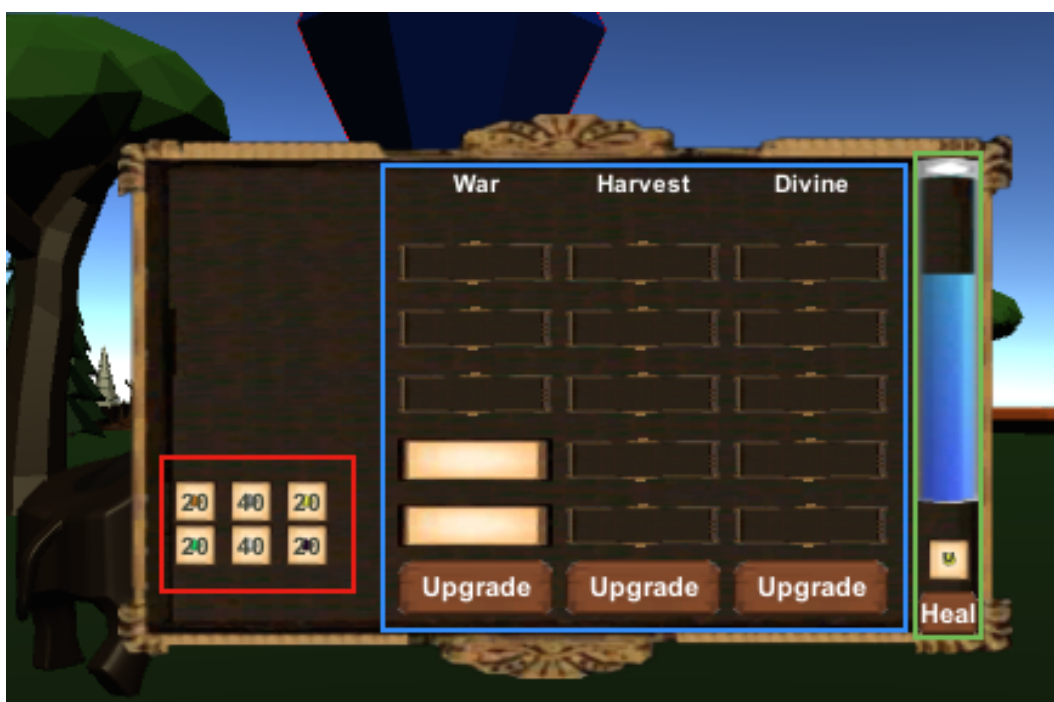


FIGURE 6 – Interface de l'interaction avec le cristal

Il est possible d'augmenter n'importe quel pouvoir du cristal en appuyant sur le bouton « Upgrade » de la colonne du pouvoir voulu et en consommant les ressources présentent dans le cadre rouge sur la figure 6. Cependant un cristal après son activation ne peut être amélioré que trois fois et chaque amélioration est plus chère que la précédente.

En mode « joueur contre joueur », un cristal est associé à l'équipe du joueur qui l'a activé. Ses pouvoirs ne s'appliquent qu'aux joueurs de cette équipe. Les autres joueurs peuvent attaquer le cristal pour faire descendre la vie de celui-ci. Cette vie est représentée par la jauge bleue dans le cadre vert de la figure 6. Lorsque la vie d'un cristal tombe à 0, il prend les couleurs de l'équipe qui l'a détruit et perd un niveau aléatoirement dans l'un de ses trois pouvoirs. Il est possible pour les joueur alliés au cristal de soigner celui-ci en consommant une dizaine de ressources identiques et en appuyant sur le bouton « Heal » en dessous de la

vie du cristal. Plus le cristal est d'un niveau élevé plus la ressource nécessaire pour le soigner est rare.

2.2 Les grandes améliorations du système de jeux

En plus d'améliorer le gameplay, nous avons également fait de grandes améliorations du système de jeux en général comme une refonte du menu principal et l'ajout d'un système de sauvegarde.

2.2.1 La sauvegarde (Florian)

La sauvegarde est une des grandes implémentations pour cette deuxième soutenance, celle-ci permet aux joueurs d'arrêter une partie et de la reprendre quand ils veulent. Le joueur qui crée une partie en est en quelque sorte le « propriétaire ». La partie est hébergée sur son ordinateur qui devient le serveur pour cette partie. C'est sur ce même ordinateur que les sauvegardes seront effectuées.

La première chose que nous avons dû modifier est la sauvegarde du monde. Pour cela nous avons utilisé un système de *seed*. Le principe est le suivant, lorsque le joueur crée un monde alors un nombre va être choisi aléatoirement. Ce nombre est appelé la *seed* du monde. À partir de celui-ci nous pouvons en déduire toutes les informations sur le monde, c'est-à-dire, les îles et leur position mais aussi leur biome et leur environnement. Ce système rend la sauvegarde légère en mémoire et donc rapide. Nous avons dû compléter le système de génération du monde et plus précisément d'un chunk.

Chaque chunk va calculer son propre *seed* à partir du *seed* du monde et de sa position, à partir de ce nouveau *seed* nous allons pouvoir connaître ses îles et son biome. Puis pour chacune des ancres nous savons l'élément qui doit être à cet emplacement.

Nous avons uniquement besoin de sauvegarder la *seed* pour sauvegarder notre monde. Cependant les modifications du joueur ne peuvent pas être déterminées grâce à la *seed*. Nous avons donc besoin de sauvegarder des informations supplémentaires telles que l'heure de la journée, les éléments détruits et posés, mais aussi les caractéristiques des cristaux. Enfin l'arrivée des succès et des statistiques de jeu doit être elles aussi sauvegardées. Pour finir chaque joueur a des informations à sauvegarder en voici la liste :

- la position ;
- le contenu de son inventaire ;
- sa vie, sa faim et sa soif ;
- les effets (régénération, poison ...) et leur temps restant ;
- son équipe en mode *PvP* .

Enfin la sauvegarde s'effectue automatiquement lors de l'arrêt du serveur, mais cela ne suffit pas. En cas de crash ou si le « propriétaire » ne ferme pas correctement l'application, la sauvegarde ne peut pas s'effectuer. Il y a donc une sauvegarde automatique toutes les minutes. Celle-ci ne provoque aucun ralentissement du jeu car elle est très légère.

2.2.2 La refonte du Menu (Romain)

Si nous avons la possibilité de sauvegarder un monde il faut aussi avoir la possibilité de choisir avec quel monde nous voulons jouer. Pour cela nous avons dû refaire le menu principal.

Dans ce menu principal nous avons rajouté une petite interface donnant accès à la liste des mondes disponibles. De même, on peut enregistrer des adresses de serveurs qui nous permettront de ne pas devoir ré-entrer les adresses IP des serveur auxquels on veut avoir accès en prenant le risque de se tromper.



FIGURE 7 – Menu Principal

Sur la figure 7, la partie encadrée en rouge n'apparaît que lorsque l'on appuie sur le bouton « Join » et montre la liste des serveurs que l'on a enregistrés. De même, une interface similaire apparaît lorsque appuie sur le bouton « Play » mais cette fois avec une liste de mondes. Cette liste de mondes est générée automatiquement en parcourant les fichiers de la sauvegarde.

Dans cette interface, il est possible de sélectionner un serveur ou un monde en cliquant dessus. On ne le voit pas vraiment mais une fine ligne blanche forme alors le contour du monde sélectionné. Il est alors possible de supprimer le monde ou le serveur de la liste en appuyant sur le bouton « Delete » ou de lancer le jeu en appuyant sur le bouton « Start ». Lorsque la liste possède plus de 3 éléments l'écran n'en affiche que trois mais il est possible

de parcourir toute la liste en appuyant sur les flèches se trouvant dans la partie droite du rectangle rouge.

Enfin il est possible d'accéder à une interface de création de monde ou de serveur en appuyant sur le bouton « Create ».

Dans l'interface de création de monde (voir figure 8) Il faut donner un nom à son monde, choisir le type de jeu que l'on veut faire en appuyant sur le bouton « Coop/PVP » pour changer de type et si l'on veut on peut préciser la *seed* de la map en base 36 (*seed* composée d'au maximum six chiffres et lettres en minuscules). Cette *seed* est propre à un monde particulier et permet donc de régénérer le même monde (sans prendre en compte les modifications faites pas le joueur). Il y a donc 2 176 782 336 mondes différents qui peuvent être générés. Il suffit alors d'appuyer sur le bouton « Create » pour créer un nouveau monde et commencer une partie. Pour des raisons de comptabilité des sauvegarde il est impossible de créer plusieurs sauvegardes du même nom.



FIGURE 8 – Création de monde

Dans l'interface d'enregistrement de serveur (voir figure 9) il nous est seulement demandé de choisir une adresse IP et un nom sous laquelle l'enregistrer puis d'appuyer sur le bouton « Create » pour enregistrer un nouveau serveur. Contrairement à la création de monde on peut créer plusieurs serveurs de même nom ou de même adresse IP sans créer de conflit.

Notre génération de monde commençant à être longue avec la génération des îles et de leurs graphes Nous avons rajouté un écran de chargement sous la forme d'une vidéo format MP4 tournant en boucle.

Enfin en jeu, nous avons rajouté des options permettant d'influencer la caméra. La première permet de modifier la sensibilité de la souris sur les mouvements de la caméra et la

deuxième permet de modifier la distance maximum de vision de la caméra. Cette dernière option à été rajoutée car avec l'agrandissement de notre monde les plus vieux ordinateurs n'étaient pas capable de faire tourner le jeu sans laguer.

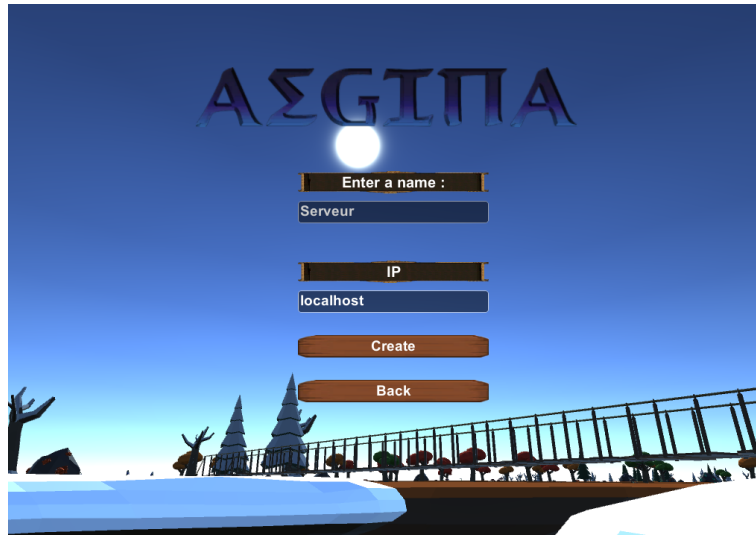


FIGURE 9 – Enregistrement d'un serveur

2.2.3 La refonte de la première scène (Julien)

La scène du menu, aussi appelée première scène, a une valeur très particulière à nos yeux. Même si cette scène ne sert que de paysage à la création de partie, elle nous permet de mettre en avant tout l'univers graphique de notre jeu et de montrer au joueur les différents éléments présents dans le jeu.

Lors du dernier rapport nous vous avons présenté une île unique avec une caméra orbitant autour. Aujourd'hui, en plus d'évoluer dans un vrai archipel, la caméra suit un mouvement ne pouvant plus être réduit à une simple fonction mathématique. Nous avons décidé que la caméra suivrait un parcours bien plus complexe (voir pointillés verts sur la figure 10) et bien plus intéressant. Plutôt que de se limiter à une orbite circulaire, la caméra va serpenter entre les îles, s'en rapprocher, s'en éloigner, voire passer en dessous.

Chaque point sur la figure 10 est en réalité une balise. La première balise connaît la deuxième balise qui connaît la troisième balise et ainsi de suite jusqu'à la dernière balise qui elle connaît la première balise. Nous obtenons ainsi un circuit fermé et il ne reste plus qu'à faire avancer la caméra le long de ce circuit. Cette technique permet de résumer des mouvements très complexes en un enchaînement de petits déplacements linéaires ce qui est

bien plus simple à réaliser. Plus il y aura de balises le long du circuit, plus le déplacement paraîtra fluide et naturel.

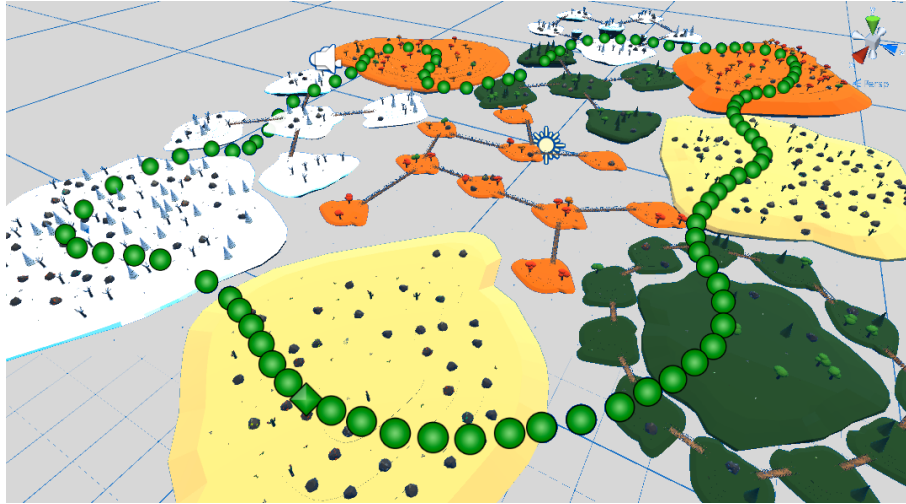


FIGURE 10 – La Première Scène

2.2.4 Le tutoriel (Théo et Romain)

Un défaut majeur de la première version de notre jeu était que celui-ci permettait de nombreuses actions (comme ouvrir un inventaire, se déplacer, jeter des objets) mais aucune information ne disait comment les réaliser. Nous avons donc décidé de créer un tutoriel expliquant au joueur comment effectuer les différentes actions dans le jeu. Nous profitons également de ce tutoriel pour introduire l'histoire du jeu puisque celle-ci s'enchaînera directement à la suite du tutoriel. Seulement contrairement au tutoriel, qui est propre au joueur puisque chaque joueur doit pouvoir apprendre à jouer, l'histoire sera propre au monde et ne se répétera pas pour chaque joueur.

Au lancement d'une partie tout nouveau joueur voit une fenêtre apparaître lui demandant si oui ou non il veut faire le tutoriel. Ne pas faire le tutoriel n'aura aucune influence sur le reste du jeu car celui-ci sera alors considéré comme réussi.

Dans le jeu, le tutoriel est visuellement présent grâce à deux zones de texte. La première zone de texte dans le rectangle rouge de la figure 11 raconte le développement de l'histoire du tutoriel. C'est dans cette partie que les raisons des actions de Ille sont expliquées, ce qui donne une logique à l'existence du tutoriel. Ce texte est temporaire et ne reste que peu de temps à l'écran car celui-ci prend beaucoup de place et risquerait d'encombrer la vision du joueur. Plus tard dans le développement du jeu cette partie sera également lue oralement par un narrateur.



FIGURE 11 – présence du tutoriel en jeu

La seconde zone de texte dans le rectangle vert de la figure 11 est directement adressée au joueur. Ce texte explique concrètement quels sont les objectifs du joueur et comment les accomplir. Il donne par exemple des indications sur quelle touche appuyer pour ouvrir l’inventaire ou sauter. Tant que les objectifs ne sont pas tous accomplis cette zone de texte ne disparaît pas. En appuyant sur la croix se trouvant en bas à droite de cette zone une fenêtre apparaît nous demandant si on souhaite terminer automatiquement le tutoriel. Cette option a été rajoutée pour les joueurs qui auraient malencontreusement lancé le tutoriel ou pour les joueurs voulant sauter la dernière étape du tutoriel car celle-ci demande d’utiliser tout ce qui a été appris dans le tutoriel et est donc une partie assez longue.

Si un joueur quitte le jeu au milieu du tutoriel sa progression sera sauvegardée et il pourra continuer le tutoriel en relançant sa partie sans tout recommencer depuis le début.

2.3 Les ajouts mineurs

2.3.1 Amélioration des commandes et système d’opérateur (Florian)

Une amélioration notable est celle du chat et des commandes. Désormais le « propriétaire » de la partie est un opérateur, cela lui permet de faire des commandes de triche telles que /GIVE. Cependant l’ensemble des joueurs possèdent des commandes utilitaires et de

communication. De plus une série de commandes ont été ajoutées avec les nouveautés du jeu. Voici un récapitulatif des anciennes et nouvelles commandes.

Disponible pour tous les joueurs :

- /help {page (int)}** : énumère les commandes avec leurs arguments si la page n'est pas spécifiée alors affiche la première page ;
- /msg <player (string)> <message (string)>** : envoie un message privé au joueur dont le nom du personnage est passé en paramètre ;
- /seed** : affiche dans le chat le seed du monde actuel ;
- /music <clip (int)>** : lance la musique correspondant au clip ;
- /team <message (string)>** : envoie un message uniquement lisible par les coéquipiers du joueur.

Disponible uniquement pour les opérateurs :

- /time <value (int)>** : détermine l'heure de la journée ;
- /give {player (string)} <id (int) ou nom (string)> {quantity (int)}** : ajoute à l'inventaire d'un joueur (ou nous-même s'il n'est pas spécifié) la quantité indiquée (ou un seul s'il n'est pas spécifié) de l'objet dont l'identifiant correspond au paramètre ;
- /tp <player (string)>** : nous téléporte vers le personnage souhaité ;
- /kick <player (string)>** : fait sortir du serveur le joueur dont le personnage est passé en paramètre ;
- /save** : force la sauvegarde du monde ;
- /kill <player (string)>** : tue le personnage passé en paramètre ;
- /effect <player (string)> <id (int)> {power (int)}** : affecte le joueur passé en paramètre de l'effet correspondant à l'identifiant avec une puissance choisie (la puissance est d'un si elle n'est pas spécifiée) ;
- /op <player (string)>** : donne les droits d'opérateur au joueur passé en paramètre ;
- /deop <player (string)>** : supprime les droits d'opérateur au joueur passé en paramètre ;
- /choseteam <id (int) ou nom (string)> {player (string)}** : attribue à une équipe choisie le joueur passé en paramètre (nous s'il n'est pas spécifié) .

Nous pouvons noter que la plupart des commandes prennent en paramètre le nom d'un joueur, il est donc utile de connaître la liste des joueurs connectés. Nous avons donc donné la possibilité aux joueurs d'afficher la liste des joueurs connectés en appuyant sur la touche *tabulation* de leur clavier, elle apparaît en haut de l'écran (voir la figure 12).



FIGURE 12 – Liste des joueurs

2.3.2 Refonte Graphique : interface et skybox (Théo, Julien et Romain)

Après la première soutenance nous avons refait la plupart de nos sprites. Ceux-ci avaient été fait à la main et n'étaient objectivement pas beaux. De ce fait de nombreux membres du groupe s'en plaignaient et ne les avaient gardés pour la première soutenance que par défaut de ne rien avoir de mieux.

Pour créer les nouveaux sprites nous nous sommes inspirés des sprites de *Don't Starve*, l'un des jeux ayant inspiré le projet, dans lequel nous avons modifié les sprites afin qu'il conviennent à notre jeu (voir figure 13). Nous avons également sélectionné des sprites ressemblant à des pancartes en bois pour faire certains de nos boutons.

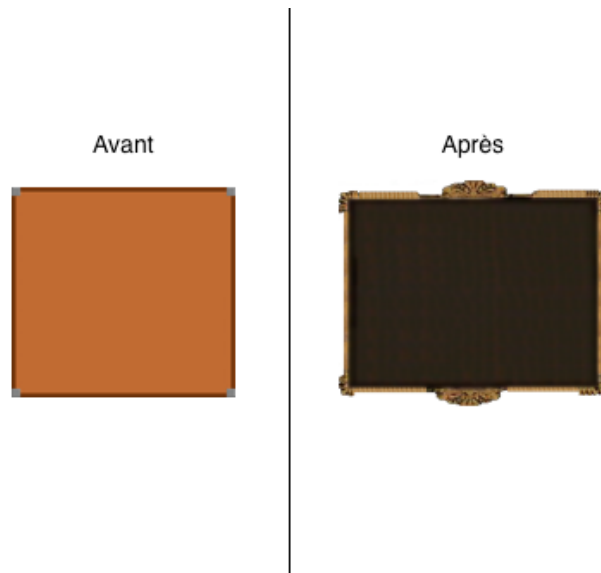


FIGURE 13 – Exemple de refonte graphique

Dans cette refonte seule la petite fenêtre apparaissant lorsque l'on passe sa souris sur un objet et qui nous donne des information sur celui-ci n'a pas été modifiée.

En plus d'avoir refait les sprites, nous avons également modifié la skybox de notre monde. Pour rappel, la skybox correspond à l'apparence du ciel et du vide se trouvant au dessus et en dessous de notre monde. Sur cette skybox, la principale modification a été de rendre le bas de notre monde beaucoup plus sombre car, étant sensé représenter des abysses sans fond, nous devions lui donner une apparence bien plus sombre qu'un simple gris clair. Maintenant, quand le joueur tombera dans les abysses il aura vraiment l'impression de tomber dans un vide sans fond.

2.3.3 Amélioration de la mort (Romain et Florian)

En tant que développeur d'un jeu de survie nous prenons tout aussi soin de la survie de nos joueurs que de leur mort. Il faut que la mort soit assez terrifiante pour que le joueur évite de mourir intentionnellement dans le seul but de revenir plus rapidement à son point de réapparition. Lors de la première soutenance, lorsque le joueur mourait il revenait à son point de réapparition sans aucune contrepartie et récupérait sa fin, sa soif et sa vie.

Maintenant, lorsque le joueur meurt son personnage disparaît et une fenêtre apparaît devant l'écran du joueur lui demandant si il veut réapparaître ou si il veut quitter. Si il décide de continuer la partie il sera transporté sur un lieu de réapparition mais tous les objets de son inventaire seront jetés au sol sur le lieu de sa mort. Si il décide de quitter, la prochaine fois qu'il lancera le jeu, il se retrouvera sur un point de réapparition mais tous les objets de son inventaire auront été jetés sur son lieu de mort avant de quitter la partie. Comme il n'existe pas d'autre moyen de transporter des objets que dans son inventaire, la mort revient presque à recommencer une partie à zéro. Heureusement, si vous avez un allié avec vous, celui-ci pourra récupérer tous vos objets mais si un ennemi se trouve dans les parages celui-ci pourra décider de vous tuer pour récupérer vos équipements.

Les points de réapparition sont les îles possédant un cristal de la même équipe que le joueur et la première île.

2.3.4 Ajout de son et d'animations (Théo)

Lors de notre première soutenance nous avons présenté un environnement sonore (musique et bruitage) assez restreint qui se limitait à quatre musiques et quelques bruitages.

Nous avons trouvé ce contenu insuffisant, enrichir ce dernier est alors devenu essentiel afin de rendre l'expérience de jeu plus agréable.

La création d'un nouveau biome, le biome automne, nous a permis d'introduire dans le jeu une nouvelle musique.

Nous avons alors quatre musiques de jeu, une pour chaque biomes. Cependant, ces

musiques se jouaient de manière aléatoire sans aucune cohérence avec le biome dans lequel nous nous trouvions. Nous trouvions cette incohérence fort dommage, nous avons alors fait en sorte que les musiques se lancent en fonction du biome sur lequel le joueur se trouve.

L'ajout de divers nouveaux bruitages et animations, se sont faits en même temps que l'ajout de gameplay au jeu. Il a fallu rendre cohérent les actions du personnage car ce dernier ne pouvait pas obtenir du bois sans couper un arbre, il nous a alors fallu créer des animations et bruitages pour les diverses actions indispensables à notre jeu comme couper du bois, ramasser des objets...

Ces multiples ajouts de bruitages et musiques ont nettement amélioré l'environnement sonore du jeu. Au final cet enrichissement sonore et ces nouvelles animations permettent une assez bonne cohérence entre ce que l'on souhaite réaliser et ce que l'on peut observer dans le jeu.

2.3.5 Ajout d'île et d'un nouveau biomes (Julien et Théo)

Lorsque nous lançons une partie lors de l'alpha, chaque univers créé était différent des autres, cependant, ils se ressemblaient toujours plus ou moins.

Cette ressemblance nous déplaisait. Afin de la minimiser, nous avons créé quatre nouveaux chunks. Cette augmentation du nombre de chunks disponibles a permis d'augmenter la diversité des îles qui apparaissent lors de la génération de l'univers. Ces chunks, contrairement aux anciens chunks, sont composés de plusieurs îles ce qui permet de diversifier le jeu. Le joueur n'aura pas l'impression de se trouver dans un monde où toutes les îles se ressemblent car certaines fois il se retrouvera sur des archipels ou dans des sortes d'arènes où la seule façon de sortir sera de sauter d'île en île.

En plus de ces ajouts de chunks, nous avons décidé de rajouter un quatrième biome, le biome automne, cet ajout nous a permis, non pas d'augmenter notre nombre d'îles, mais de diversifier les éléments que pouvaient contenir ces dernières.

Cet ajout de biome ne s'est pas juste fait en changeant la couleur du sol. Un réel ajout de contenu a été réalisé. Vous ne trouverez les différentes espèces de champignons et citrouilles existantes dans Aegina que dans un biome automne. De plus avec sa musique qui lui est propre cet ajout amène vraiment un nouvel environnement dans Aegina.

Les améliorations ne se sont pas arrêtées au nouveau biome car nous avons modifié certains biomes déjà existants. Par exemple, nous avons rajouté des carcasses d'animaux dans le biome été.

La création de ce biome automne, en plus de diversifier le contenu de Aegina, nous a permis de compléter le jeu, car avec l'ajout de ce biome, l'univers d'Aegina est dorénavant

composé de quatre biomes, les quatre saisons.

En plus des ajouts liés aux biomes, nous avons rajouté de nombreux objets que le joueur peut utiliser comme des haches, des épées, des nouveaux plats, des matériaux de construction, des torches et des établis.

2.3.6 Les Stats (Florian et Julien)

Nous avons décidé de créer un objet regroupant des d'informations sur l'évolution d'une partie. Parmi ces informations, nous trouvons le nombre de créatures et de joueurs tués par un joueur ainsi le nombre de joueurs morts, la quantité d'objets fabriqués, etc.

Comme vous pouvez vous en douter, ce sont sur ces statistiques que les succès s'appuient. Elle sont donc, comme les succès, propres à la partie et non au joueur.

2.3.7 Bêta (Florian)

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

- L'animation d'attaque par le joueur ;
- L'interaction avec des petits éléments ;
- Un fort déséquilibre du jeu, celui-ci était beaucoup trop dur.

Nous avons corrigé la totalité des bugs rapportés, puis grâce aux avis, nous avons ajouté et modifié quelques détails artistiques et de gameplay. Ces modifications ont abouti à la version *v.1.1-beta* qui est la version présentée lors de la seconde soutenance.

2.4 Le site web (Théo)

Le site web que nous avons créé (accessible à l'adresse <http://jmounier.github.io/Aegina-Website/>) avait pour notre équipe plusieurs objectifs à remplir :

- nous devons respecter l'obligation qui était de construire un site web ;
- créer un lieu où nos testeurs et nos, nous espérons, futurs joueurs puissent trouver toutes les informations nécessaires afin de rendre leur expérience en jeu agréable ;
- permettre à ces personnes de télécharger notre jeu mais aussi de nous transmettre leurs expériences.

Tout ceci est réalisable depuis les différentes pages présentes sur le site, que nous allons maintenant vous présenter.

2.4.1 la page d'accueil

Sur cette page (voir figure 14) vous pourrez apercevoir, au premier plan plusieurs éléments :

- un bref descriptif du jeu ;
- des accès pour télécharger la version bêta de Aegina en fonction de votre système d'exploitation, Windows, Mac et GNU/Linux ; -
- la dernière actualité ajoutée sur le site.

Vous pouvez aussi voir sur le côté un menu, il permet de naviguer entre les différentes pages du site web, il est positionné au même endroit sur toutes les pages.

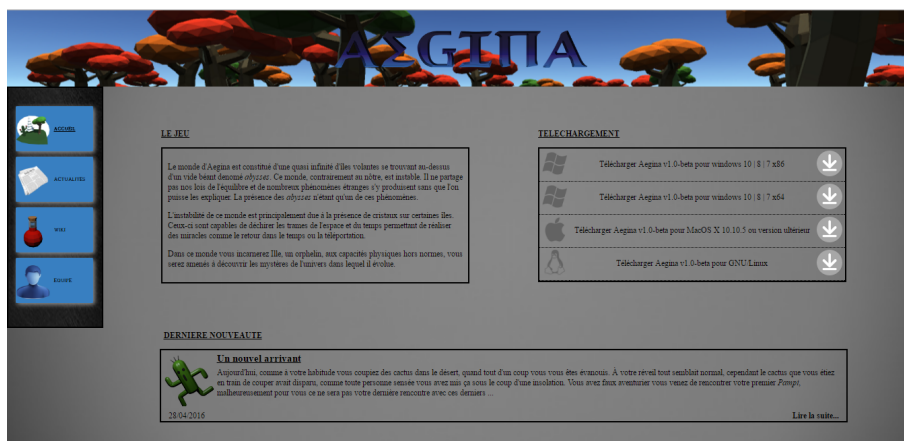


FIGURE 14 – Page d'accueil du site web

2.4.2 Les actualités

Sur la pages des actualités (voir figure 15), que nous tiendrons à jour vous pourrez trouver les différents ajouts qu'il y aura dans le jeu.

Les actualités pourront concerner l'ajout d'un nouveau biome, de monstres ou encore le lancement d'une nouvelle version disponible au grand public.

Nous espérons que les différentes annonces qui seront faites sur cette page rempliront les attentes que vous avez envers Aegina.



FIGURE 15 – Actualités du site web

2.4.3 Le wiki

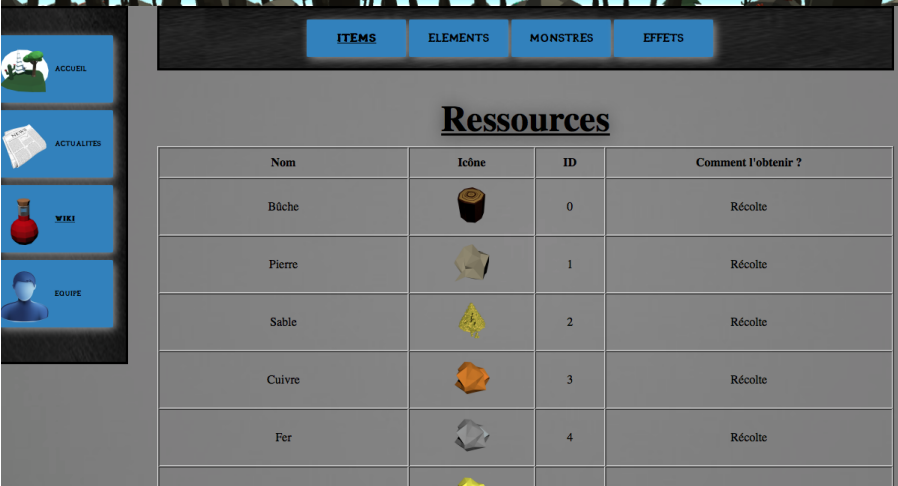
La page wiki (voir figure 16) est pour notre équipe une page très importante, nous connaissons Aegina et donc nous connaissons la difficulté de rester en vie dans ce monde hostile même pour des personnes aguerries, c'est pourquoi nous avons décidé de mettre en place cette page.

Dans ce wiki vous pourrez trouver toutes sortes d'informations, toutes plus utiles les unes que les autres pour survivre.

Vous y trouverez les différents objets que vous pouvez posséder dans le jeu ainsi que leurs possibles effets, les éléments et monstres qui existent dans Aegina.

Le wiki est actuellement en quatre parties : la liste des objets, la liste des éléments que l'on trouve sur les îles, la liste des créatures et la liste des effets. On peut passer d'une partie à l'autre en cliquant sur le nom de la partie en haut de la page

Il faut cependant savoir que ce wiki est rempli au fur et à mesure de notre développement, il peut donc manquer certaine informations.









Nom	Icône	ID	Comment l'obtenir ?
Bûche		0	Récolte
Pierre		1	Récolte
Sable		2	Récolte
Cuivre		3	Récolte
Fer		4	Récolte
			

FIGURE 16 – Wiki du site web

2.4.4 L'équipe

Sur la dernière page de notre site web vous pourrez trouver diverses informations sur les membres de l'équipe AIM², ces informations présenterons les membres de l'équipe et leur rôle dans la création du jeux Aegina.

3 Prévisions

Dans cette partie nous allons vous présenter les différents éléments que nous allons ajouter à notre jeu pour notre troisième soutenance et qui nous permettrons de terminer celui-ci.

3.1 Les quatre grands axes

En plus des améliorations de tous les éléments présents il y aura également des ajouts majeurs dans notre jeu qui permettront de faire d'Aegina un jeu complet et fini.

3.1.1 Histoire et fin

Actuellement l'histoire d'Aegina est encore pauvre mais dans la version finale du jeu un vrai scénario existera et permettra, en le complétant, d'arriver à la fin du jeu. Cette histoire sera globalement linéaire sauf sur la fin où deux fins différentes pourront-être débloquées, par le joueur ou automatiquement.

3.1.2 Créatures : plus de créatures et introduction de Boss

Notre univers ne compte actuellement que deux créatures différentes mais c'est bien trop peu pour la diversité du monde d'Aegina donc nous allons en rajouter afin de rendre le jeu plus vivant, imprévisible et surprenant.

De plus avec l'avancement dans le jeu, celui-ci pourrait commencer à devenir trop simple. C'est pour cela que nous allons également rajouter des Boss. Ils pourront être liés à l'histoire du jeu ou tout simplement être rencontrés par chance mais nous n'allons pas vous en dire plus et vous laisser la surprise pour la troisième soutenance.

3.1.3 Un monde infini

Un des éléments propre au monde d'Aegina était que celui-ci devait être infini. Jusqu'à présent nous n'avons généré qu'un certain nombre d'îles mais nous allons instaurer un système de chargement et déchargement qui permettra au joueur de se déplacer dans un monde quasiment infini. Pour être exact notre système de génération fonctionnant à partir d'entiers, si le joueur parcourt un très grand nombre d'îles (deux fois la valeur maximale des entiers) celui-ci reviendra sur des îles identiques à celles qu'il a déjà visitées. Mais cette limite est limite théorique car il est inimaginable qu'un joueur tente de faire ce genre d'exploit.

3.1.4 PVP : refonte du système de combat

Actuellement le système de combat est très pauvre et peu varié. Pour donner aux joueurs une meilleure expérience de jeu dans le mode PVP et en combattant les créatures d'Aegina nous avons décidé de faire une refonte de ce système pour la troisième soutenance. Le joueur aura alors plusieurs choix pour attaquer autres que le simple fait de marteler son clic gauche en visant son adversaire.

3.2 Les améliorations

La 3ème soutenance signifie aussi la fin du projet. Lors de celle-ci le jeu doit être considéré comme un jeu complet et fini. Nous allons donc apporter des finitions aux éléments existants pour qu'ils puissent être de vrai éléments de jeu fini.

3.2.1 Survie : équilibrage

Il est évident que notre système de survie ne peut pas être équilibré dès le début. Nous allons donc, pour la troisième soutenance, faire des équilibrages par rapport au rendu que nous aurons des joueurs de la bêta. Ces modifications ne seront pas forcément visibles en jeu mais seront très appréciées de nos testeurs en général.

3.2.2 Cristaux : influence divine

Actuellement les cristaux, même s'ils ont un certain impact dans le monde, ne sont pas encore assez puissants. Les cristaux sont censés être dotés de pouvoirs divins qui leur permettent de faire trembler le monde. Cette puissance des cristaux ne se fait pas encore sentir dans le jeu. Nous allons modifier cela afin que le contrôle des cristaux soit d'une importance primordiale pour la réussite du jeu Aegina.

3.2.3 PVP : une vraie fin possible

En plus de rajouter une fin au mode coopération nous allons également permettre aux parties joueur contre joueur de se terminer avec un gagnant. Après tout si quelqu'un joue une partie joueur contre joueur c'est avant tout pour détruire ses ennemis et montrer qu'il est le plus fort. Si il est impossible de finir une partie on ne peut plus prouver sa supériorité et ce mode de jeu perd tout son sens. C'est pourquoi nous allons régler ce problème.

3.2.4 Améliorations en masse : graphisme, animation, son et musique

Finalement nous allons également faire tout un tas d'autres améliorations, par ajout ou modification, sur les graphismes, les animations, les sons et les musiques.

Conclusion

Pour la deuxième soutenance d'Aegina nous avons encore une fois réussi à atteindre les objectifs fixés par le cahier des charges. Cette fois-ci nous n'avons pas pris énormément d'avance sur la prochaine soutenance mais nous avons posé de bonnes bases pour les ajouts et améliorations prévus. Nous avons déjà fait des brainstorming et nous avons donc des idées de comment implémenter ces divers éléments. Notre beta a assez bien marché et les testeurs ont enfin commencé à s'amuser.