



42 Draconis

Par SpaceY

Premier rapport de soutenance - Promotion 2026

Lucas BURGAUD, Neil SAUVAGE, Etienne SENIGOUT, Bastien GAULIER

Mars 2022



Table des matières

1	Introduction	3
2	L'équipe	3
2.1	Etienne SENIGOUT, chef de projet	3
2.2	Lucas BURGAUD	3
2.3	Bastien GAULIER	3
2.4	Neil SAUVAGE	3
3	Ce qui a été fait	4
3.1	Général	4
3.2	Réseau	4
3.3	Le vaisseau	4
3.3.1	L'intérieur	5
3.3.2	L'extérieur du vaisseau	5
3.4	Les ennemis et l'IA	7
3.5	Ambiance et sons	8
3.6	Génération des planètes	9
3.6.1	Génération d'une sphère	9
3.6.2	Générateurs pour déformer les sphères	10
3.6.3	Génération des planètes	13
3.6.4	Graphisme	13
3.6.5	Optimisation	14
3.7	Génération des systèmes solaires	14
3.8	Communication	14
3.8.1	Site Web	14
3.8.2	Réseaux Sociaux	15
3.9	Inventaire	15
4	Ce qu'il reste à faire	16
4.1	Général	16
4.2	Les ennemis et l'IA	16
4.2.1	Préciser les variables	16
4.2.2	Nouveautés et améliorations	16
4.3	Ambiance et sons	18
4.4	Génération des planètes	18
4.5	Génération du système solaire	18
4.6	Inventaire	18
4.7	Graphisme	18
4.8	Communication	19
5	Organisation et avancée	19
5.1	Organisation	19
5.2	Avance sur le planning	21
5.3	Retard sur le planning	21
6	Conclusion	21
7	Bibliographie	21

1 Introduction

42Draconis est un jeu 3D d'exploration spatiale en coopération développé par l'équipe *SpaceY*. Le jeu est conçu pour se jouer à deux et il se déroule dans un monde généré procéduralement. Les joueurs devront explorer et survivre ensemble pour réussir à finir le jeu en trouvant la fameuse planète *42Draconis*.

Depuis la réalisation du cahier des charges de *42Draconis*, chaque membre de *SpaceY* s'est beaucoup investi dans le développement du jeu. Chacun est motivé et veut toujours aller plus loin dans le projet. Nous avons depuis Janvier réussi à mettre en place tout ce que nous avons prévu pour cette première soutenance, et même plus encore. Dans ce rapport, sont listées nos réalisations ainsi que nos prévisions pour la deuxième soutenance.

2 L'équipe

2.1 Etienne SENIGOUT, chef de projet

La mise en place des idées, l'organisation des tâches ainsi que de la prise en main des différents outils a été un peu longues. Mais une fois cette période passée, chacun a fourni un énorme investissement dans la réalisation des tâches. Nous avons tous de l'avance par rapport à notre planning et c'est très bien, car cela veut dire que nous pouvons faire encore plus de choses jusqu'à la dernière soutenance.

Je suis plutôt fier de l'équipe, il y a un vrai esprit d'entraide, de communication, beaucoup d'investissement et surtout de la motivation. On se tient au courant tous les jours par Discord et en cours ainsi qu'avec des réunions sur le temps du midi ou bien par appel à distance. Cette communication est très importante puisqu'elle permet de créer un climat de confiance et d'entraide. Chacun n'hésite pas à poster ses difficultés en sachant qu'il recevra de l'aide en retour.

Malgré toutes les réalisations, il reste tout de même beaucoup à faire, mais nous ne baisserons pas les bras, nous allons garder cette motivation et cet esprit jusqu'au bout !

2.2 Lucas BURGAUD

Ce projet a avancé à un bon rythme ces dernières semaines. Le début est toujours compliqué, car les premières tâches sont des pas dans le vide et l'inconnu, mais maintenant que la machine est lancée, on arrive plus facilement à savoir où l'on va. La mise en place d'outils de communication et d'organisation a beaucoup aidé à l'avancement rapide du projet.

Pour parler de l'équipe, je suis fier du groupe, j'ai vraiment l'impression que chacun a sa place. Ce qui est intéressant sur un projet comme ça, c'est qu'il y a tellement de tâches différentes à faire que chacun arrive à trouver des tâches qui lui correspondent. Tout le monde est motivé et investi, nous nous poussons vers le haut les uns, les autres et dès que l'on a des questions ou des problèmes, on peut être sûr que quelqu'un sera rapidement disponible pour répondre à nos questions.

Nous avons même pris de l'avance, ce qui met en confiance pour les semaines qui arrivent, cela évite aussi d'avoir de la pression, du stress, et une peur continue. Mais ne prenons pas trop confiance non plus, nous sommes conscients qu'il reste encore beaucoup à faire.

2.3 Bastien GAULIER

En ce qui concerne mon ressenti sur cette première période, le développement concret de notre projet a commencé par l'appropriation de l'outil Unity, qui a pu être assez frustrante au début pour moi. Cependant, j'ai vite surmonté cette difficulté et j'ai donc pu commencer à développer beaucoup d'idées, c'est-à-dire les expérimenter, les corriger, les améliorer et les optimiser.

Durant cette première période de développement, j'ai donc appris de nombreuses choses en expérimentant sur Unity, et j'ai ainsi obtenu des résultats dont je suis fier. De plus, l'avance que nous avons prise nous permet de réaliser plus de choses d'une manière plus complète. Enfin, les outils de communication et d'organisation mis en place assez tôt nous ont permis d'être efficaces et de prendre de l'avance sur notre planning initial. Notre objectif pour la seconde soutenance : maintenir l'avance que nous avons prise !

2.4 Neil SAUVAGE

Durant cette première partie de la création du jeu, j'ai appris de nombreuses choses. L'avancée était rapide au départ jusqu'à ce que des problèmes plus complexes apparaissent, qui m'ont pris énormément

de temps à résoudre.

Je suis cependant fier du travail que j'ai fourni pour le moment, qui est en avance par rapport au planning. De nombreuses choses restent encore à faire, mais le groupe est soudé et nous avançons ensemble. La charge de travail déjà présente m'a poussé à changer mes façons de travailler en m'organisant mieux et en réduisant mon sommeil pour me dégager du temps dédié au jeu.

3 Ce qui a été fait

3.1 Général

Depuis la validation du cahier des charges, chacun a mis en place le squelette de ses parties. Nous ne nous sommes pas encore concentrés sur l'histoire, l'ambiance sonore ainsi que tous les effets graphiques. Notre objectif était d'avoir un réseau fonctionnel, une idée précise de ce que l'on devait faire ainsi qu'un aperçu de chaque partie du jeu.

Voici donc un résumé de tout ce qui a été réalisé par notre équipe jusqu'à présent pour *42Draconis*.

3.2 Réseau

Le premier point mis en place a été le réseau par Neil SAUVAGE, car il est la base du projet jeu vidéo. Il devait pour cette première soutenance être fonctionnel. Il y a donc eu sur cette partie un gros travail pour débiter sur une base solide.

Dans un premier temps, il a fallu réaliser la base du réseau, c'est-à-dire se familiariser avec les outils de Unity comme Mirror, qui permet de gérer les connexions et transferts de données. Les premiers essais étaient seulement les déplacements de deux joueurs puis du vaisseau.

Une fois cette étape terminée, il a nécessité un gros travail pour qu'un des joueurs puisse conduire le vaisseau pendant que l'autre puisse tirer ou bien que les deux joueurs puissent se déplacer librement dans le vaisseau et échanger leurs rôles. Le développement du réseau a cependant été assez rapide une fois le fonctionnement compris, cela a permis de rapidement développer notre projet sur des bases solides.

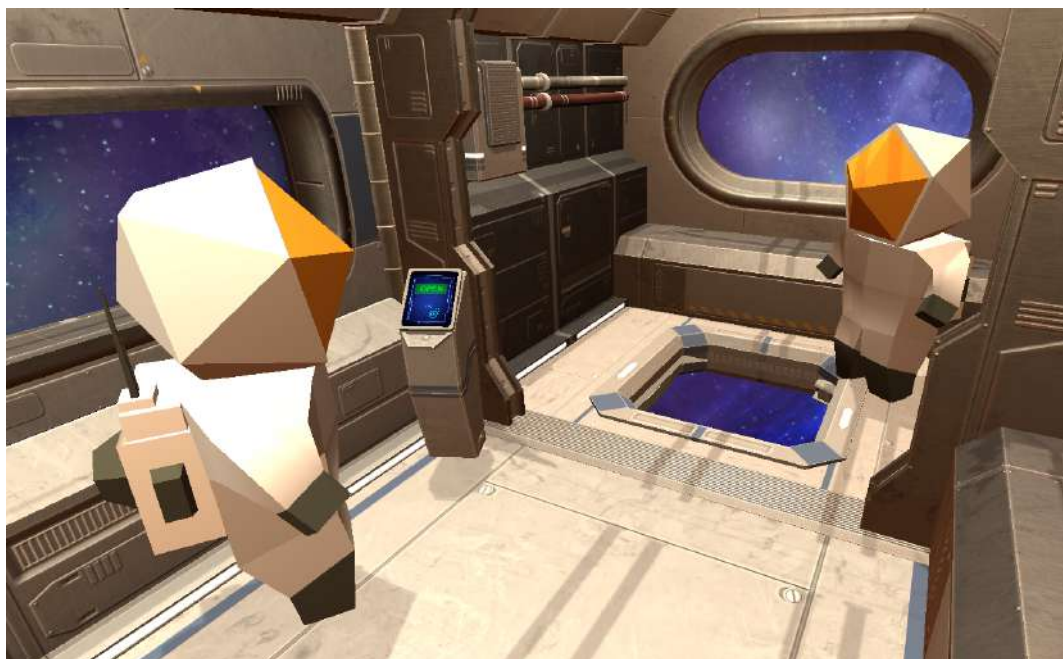


FIGURE 1 – Deux joueurs prêts à sortir du vaisseau

3.3 Le vaisseau

Cette partie a également été réalisée par Neil SAUVAGE.

3.3.1 L'intérieur

L'intérieur du vaisseau contient différentes pièces. Il y a la pièce pour l'apparition du joueur. Que ce soit après sa mort ou à son apparition, cette pièce contient des capsules de cryogénéisation ainsi qu'une console qui nous servira plus tard pour la gestion du vaisseau et de ses améliorations.

Une seconde pièce est le sas du vaisseau où les joueurs pourront s'équiper d'un jetpack qui leur apportera de l'oxygène. Il leur permettra de survivre sur des planètes dangereuses et de se déplacer dans l'espace. Dans cette pièce, il y a aussi une console permettant de dépressuriser le sas pour pouvoir sortir du vaisseau.



FIGURE 2 – Salle d'apparition des joueurs

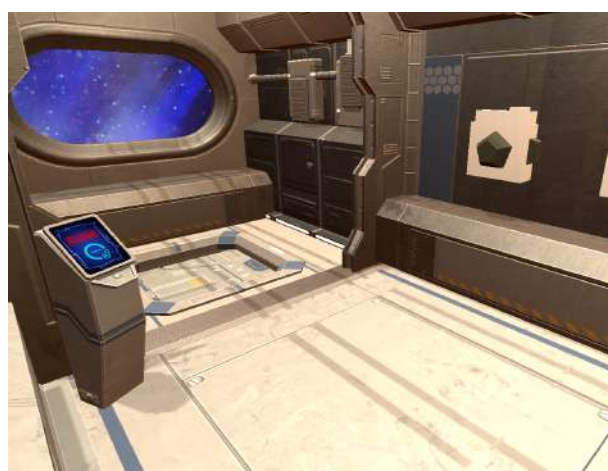


FIGURE 3 – Le sas

Deux dernières pièces permettent respectivement de diriger le vaisseau en accédant au poste de pilotage ou d'accéder au poste de tir en montant une échelle. L'intérieur du vaisseau sera important pour permettre aux joueurs d'échanger leurs rôles, de sortir du vaisseau ou encore de réparer les parties endommagées.

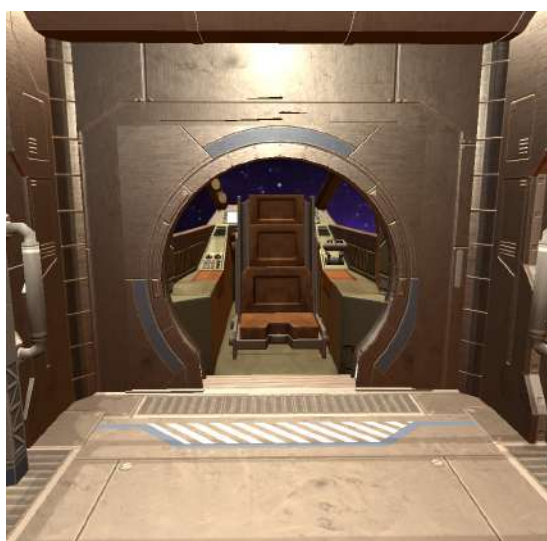


FIGURE 4 – Accès au poste de pilotage

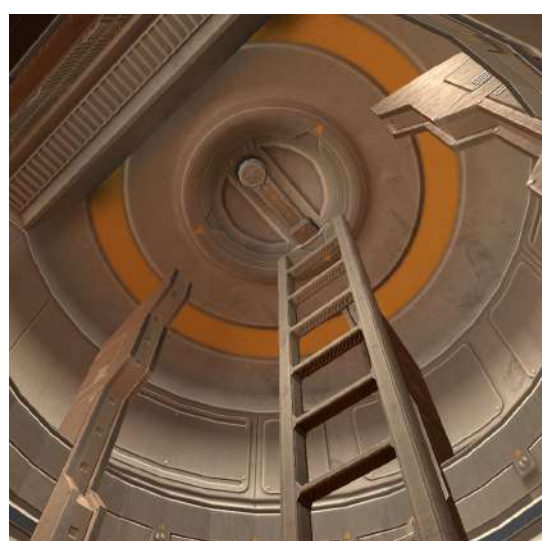


FIGURE 5 – Accès à la tourelle

3.3.2 L'extérieur du vaisseau

Pour l'extérieur du vaisseau, on aura différents accès pour le conducteur et le tireur. Tout d'abord, le conducteur possède une vue extérieure du vaisseau où il a une vision plus large et générale. Cependant,

cette vue manque d'informations. C'est pour cela qu'une vue du poste de pilotage est possible où on peut voir la vitesse et le carburant restant. Il y a aussi deux écrans montrant les zones endommagées à gauche et aussi une caméra facilitant l'atterrissage à droite. En effet, dans cette vue, la caméra pour l'atterrissage permet de voir sa distance par rapport au sol et permet un atterrissage en douceur. Pour rajouter plus de liberté dans cette vue, il est possible de tourner sa tête dans le poste de pilotage afin d'avoir un champ de vu plus grand.



FIGURE 6 – Vue à la troisième personne



FIGURE 7 – Vue dans le poste de pilotage

Un autre élément à l'extérieur est la tourelle. Elle permet au joueur de tirer sur les ennemis et de protéger le vaisseau durant ses déplacements.



FIGURE 8 – Tourelle de défense

3.4 Les ennemis et l'IA

En ce qui concerne les ennemis de notre jeu, notre objectif pour cette soutenance était d'avoir mis en place les bases les concernant. Nous nous sommes penchés sur le développement d'entités sous la forme de vaisseaux, censés pourchasser une cible à travers l'espace, ainsi que l'attaquer à l'aide de projectiles. Le développeur principal de cette partie du projet est Bastien GAULIER.

Il a tout d'abord fallu réfléchir à leurs apparences, les rapports qu'ils auront avec les joueurs, leurs modes de déplacements ainsi que leur génération, c'est-à-dire comment, où et quand ils doivent apparaître.

Ensuite, une fois tout cela posé, le développement a pu commencer. Il a tout d'abord été mis en place un squelette de vaisseau, en partie grâce à des aides trouvées sur Internet. Ce "vaisseau" pouvait reconnaître un objet particulier possédant un certain tag (un mot) dans un certain rayon autour de lui, ainsi que pivoter de manière linéaire vers elle. Si plusieurs cibles se trouvaient dans son rayon de détection, le vaisseau choisissait la plus proche. Il pouvait aussi faire apparaître des projectiles qui se dirigeaient vers cette cible, et disparaissaient au bout d'un certain temps ou s'ils la touchaient. Ces projectiles n'étaient tirés que si l'angle entre l'ennemi et l'objet ciblé était assez faible.

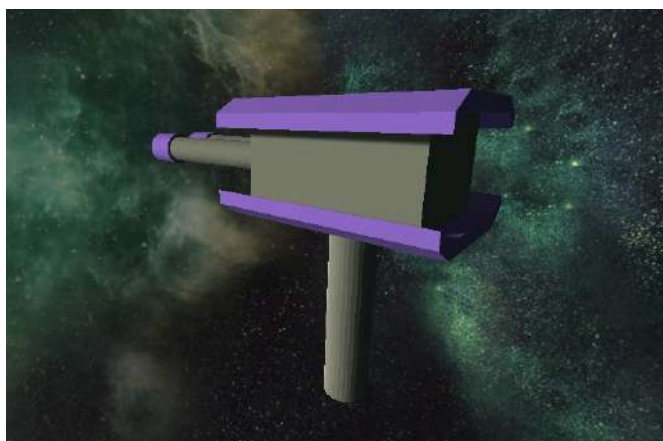


FIGURE 9 – Ancien concept d'ennemi "tourelle".

Le système de mouvement a été rajouté peu après, et il a été sujet à de nombreuses évolutions. Tout d'abord, l'ennemi avançait de manière linéaire en direction de sa cible s'il lui faisait face. Ensuite, de même que pour les projectiles, il ne bougeait que si sa direction (son "regard") formait un angle

assez faible avec l'objet traqué. Peu après, le mouvement a été revu, le vaisseau se déplace maintenant avec l'aide de forces, et non de variation de position. De plus, si le vaisseau était trop près de sa cible, il lui est désormais appliqué une force de recul pour le faire s'éloigner d'elle. Enfin, si un vaisseau ne traquait rien, mais qu'il avait toujours une vitesse, il peut maintenant ralentir grâce à une force de frottement. Cependant, pour rester un peu réalistes, nous avons décidé de ne pas le faire s'arrêter totalement.

L'étape d'après a été de rajouter un peu d'intelligence dans ces vaisseaux. C'est pourquoi un système de traque a été incorporé. Désormais, le vaisseau vérifie que rien ne se trouve entre lui et sa cible avant d'essayer de pivoter ou de se diriger vers elle. Si quelque chose est présent, l'ennemi ne va rien faire. Durant cette étape, les différents scripts gérant ces vaisseaux ont été revus, simplifiés et optimisés, ainsi que modifiés à l'aide de notions vues en cours.

Il a été ensuite temps d'ajouter un peu de graphismes et de formes concrètes à ces vaisseaux. Jusqu'à présent, ils s'agissaient de simples polygones ou plusieurs objets primitifs mis ensemble. Cependant, nous avons trouvé sur Internet de magnifiques modèles de vaisseaux libres de droit et gratuits dont les graphismes nous plaisaient et convenaient avec notre vision de départ. De plus, nous nous sommes rendus compte qu'ils étaient modulaires, c'est-à-dire que l'on peut créer nos propres modèles à l'aide de plusieurs petites parties de vaisseaux ! Nous avons donc décidé pour l'instant que nos ennemis auraient les modèles de ce package, nommé StarSparrow Modular Spaceships.

Poursuivons le long du développement de nos ennemis, et entrons dans la phase des relations plus concrètes avec l'extérieur. Ce qu'il faut entendre par là, c'est un système de vie, de dégâts et de collisions. Les vaisseaux disposent désormais d'une vie définie par un nombre, et ce nombre baisse si l'ennemi est touché par un projectile disposant d'un paramètre de dégâts. Sa santé baisse aussi lors d'une collision avec autre chose de physique, et la vie perdue est déterminée par la différence entre la vitesse de ces deux corps. Si la santé d'un ennemi passe en dessous de 0, celui-ci disparaît dans une explosion de particules. D'ailleurs, des particules différentes sont aussi émises du vaisseau quand la vie de celui-ci est en dessous de 60%, 40% puis 20%. Ces particules, comme bien d'autres choses, sont personnalisables.



FIGURE 10 – Vaisseau ennemi détruisant une cible.

Enfin, il a bien fallu se pencher un minimum sur les détails et les composants divers du jeu, car ce sont eux qui lui ajoutent le plus de vie. La première chose qui a été ajoutée est le fait que les moteurs situés à l'arrière des vaisseaux s'allument lorsqu'ils accélèrent ou reculent, cela de manière progressive. L'intensité de la lumière émise n'est pas gênante pour le joueur, mais est tout de même visible.

Note : tous les fonds, modèles, effets montrés ici sont temporaires et seront donc éventuellement amenés à changer.

3.5 Ambiance et sons

Concernant l'aspect son et musique du jeu, l'objectif pour cette soutenance était assez léger, le but était seulement de faire des recherches et des essais. Cette partie du projet est réalisée par Lucas BUR-

GAUD et Bastien GAULIER.

Nous avons ainsi fait des recherches sur le site *Lasonothèque* qui est une banque de son totalement gratuite et libre de droits qui rassemble plus de 2000 bruitages destinés justement à des projets d’audiovisuel. Nous avons pu rassembler une quarantaine de sons qui nous seront utiles dans le projet, et si, lors de l’application de ces bruitages, l’un d’eux est manquant, nous pourrons aller piocher dans cette banque de sons.

Nous avons aussi trouvé le site *Uppbeat* qui rassemble des musiques de milliers d’artistes, la version gratuite permettant de télécharger jusqu’à 10 musiques libres de droit par jour.

Nous avons aussi commencé la création de nos propres musiques d’ambiance à l’aide d’*Audacity* et d’un piano. Cependant, concernant celles-ci, nous avons juste mis les outils en place et discuté le style que nous voulons leur donner. Nous hésitons encore entre faire la musique nous-même ou se servir dans une banque de son.

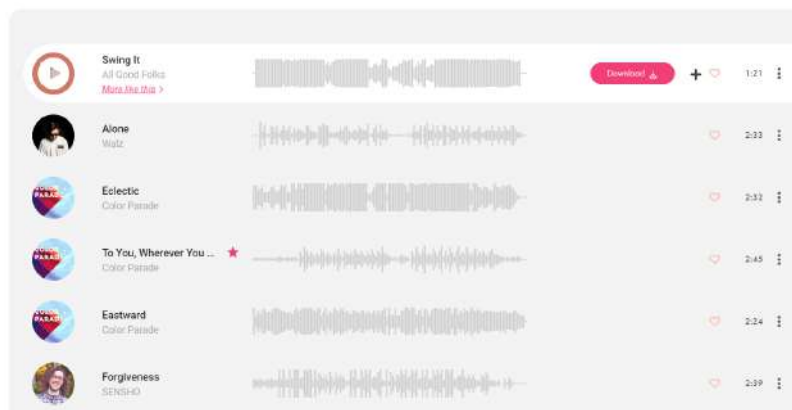


FIGURE 11 – Page d’accueil du site Uppbeat.io.

3.6 Génération des planètes

La génération des planètes s’est découpée en plusieurs parties. Tout d’abord, il a fallu avoir une forme sphérique pour commencer à travailler. Ensuite, nous avons mis en place différents générateurs pour déformer cette sphère. Une fois que nous avons eu suffisamment de générateurs différents, nous avons équilibré leurs paramètres et créé des combinaisons particulières afin d’obtenir les différents biomes voulus. La touche finale a été de mettre en place des graphismes propres à chaque biome. Toutes les recherches et le code pour cette partie ont été mis en place par le développeur Etienne SENIGOUT. La validation concernant la forme et l’apparence des biomes a été réalisée par toute l’équipe.

3.6.1 Génération d’une sphère

Nous avons besoin de générer une sphère avec une résolution personnalisable. C’est-à-dire avec un nombre de vertex contrôlable. Un vertex est, en imagerie 3D, un point d’une forme géométrique quelconque. Trois vertex reliés ensemble forment un triangle qui est la figure primitive dessinée par un ordinateur. C’est en connectant des milliers, et même des millions de triangles que l’on obtient n’importe quel objet.

Il est donc évident que plus une figure possède de vertex, plus elle possède de détail. Nos planètes ont besoin d’un très grand nombre de détails. Or, les sphères proposées par Unity n’offraient que très peu de détail et la répartition des vertex n’était pas uniforme. Il a donc fallu trouver un moyen de générer des sphères.

Notre toute première approche a été de créer un cube en générant une à une les six faces avant de les assembler. Une fois notre cube fraîchement obtenu, on va le transformer en une sphère parfaite. Pour ce faire, on va normaliser chaque point de toutes ses faces. La normalisation va permettre de garder le sens et l’orientation d’un vecteur tout en raccourcissant sa distance à une unité. Concrètement, les côtés du cube vont se tasser jusqu’à former une sphère (Cf. figure 12).

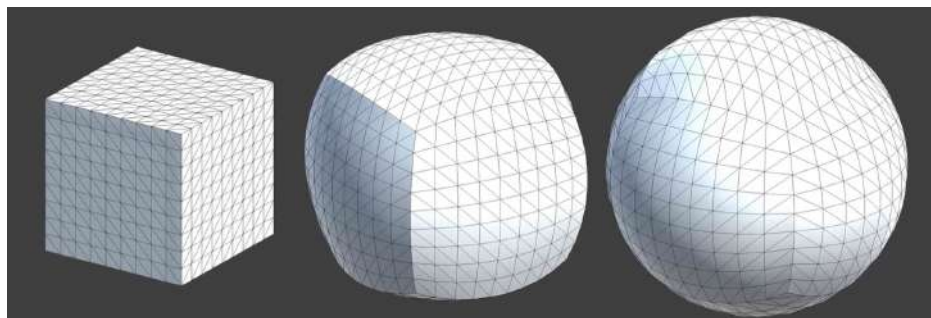


FIGURE 12 – À gauche notre cube, au milieu une demi normalisation et tout à droite, notre sphère.

L'avantage de cette technique est que la résolution est facilement contrôlable et l'algorithme très facile à mettre en place. Seulement, la répartition des points n'est pas tout à fait uniforme. En effet, lorsque notre cube est transformé en sphère, la fréquence des points est bien plus importante au niveau des extrémités des faces qu'au niveau de leur centre (Cf. figure 13).

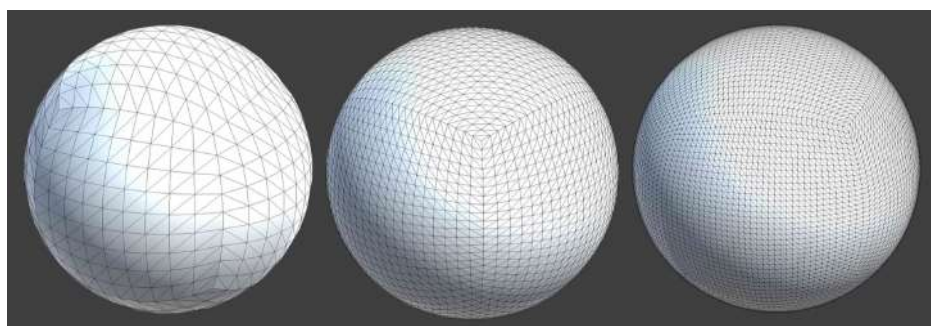


FIGURE 13 – Différents niveaux de résolution et forte concentration de point au niveau des angles.

Pour cette raison ainsi qu'un autre problème par rapport à la lumière, nous avons cherché d'autres techniques. Après plusieurs approches différentes, nous nous sommes finalement arrêté sur ce que l'on appelle une icosphère. Cette méthode part d'un icosaèdre, un polygone à 20 faces triangulaires. On découpe ensuite chaque face en un nombre contrôlable de sous-triangles, ce qui va transformer progressivement notre polygone en une sphère. L'algorithme, bien que plus complexe à mettre en place que celui précédemment expliqué, offre une répartition parfaitement uniforme des points et ne possède aucun problème pour la lumière (Cf. figure 14).

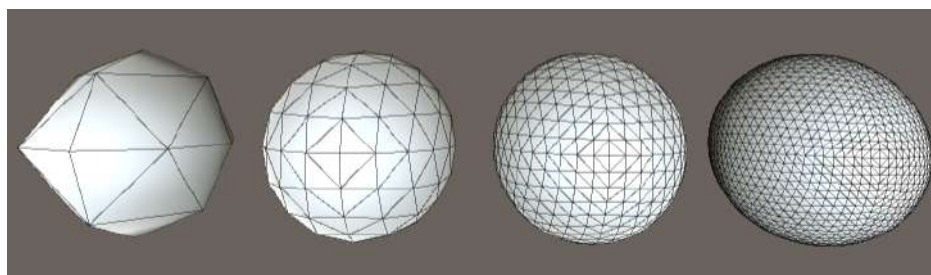


FIGURE 14 – À gauche l'icosaèdre et à droite, différents niveaux de résolution.

3.6.2 Générateurs pour déformer les sphères

Maintenant que nous avons nos sphères, il est temps de les déformer. Pour transformer nos sphères en planètes, on attribue à chaque vertex une hauteur. Cette hauteur va être générée par différents algorithmes que nous avons mis en place.

Le premier type de générateur de hauteur se base sur du bruit. En génération procédurale, le bruit consiste à générer un ensemble de nombres pseudo-aléatoires. Dans notre cas, nous utilisons des bruits

de type gradient et de type cellulaire.

Les bruits gradients sont appelés de cette façon puisqu'il y a une certaine cohérence dans la génération des nombres aléatoires. Il est l'opposé du bruit blanc qui lui n'est qu'un ensemble de nombres sans aucune cohérence (Cf figure 15).

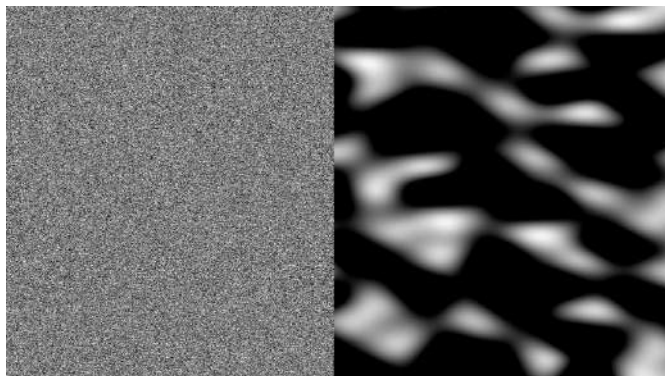


FIGURE 15 – À gauche du bruit blanc et à droite un bruit gradient.

Comme vous avez peut-être pu le reconnaître sur l'image, nous utilisons le *Bruit Simplex*. Nous avons préféré utiliser celui-ci à la place du très connu *Bruit de Perlin* puisque le *Bruit Simplex* est bien plus rapide à générer en 3D que son prédécesseur. Avec ce bruit, nous allons pouvoir générer la forme globale des planètes ou bien des détails sur celle-ci (Cf. figure 16).



FIGURE 16 – À gauche le bruit sous forme d'image, au centre la forme globale et à droite l'ajout de détail.

Nous avons aussi mis en place des variantes du *Bruit Simplex* pour créer des montagnes ou bien des rivières (Cf. figure 17).

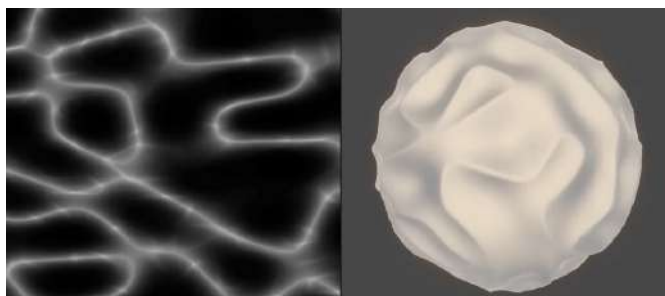


FIGURE 17 – À gauche le bruit sous forme d'image et à droite sa représentation.

Le deuxième type de bruit que nous utilisons est le bruit cellulaire. Il est appelé ainsi puisqu'il consiste à découper notre espace en cellules. Le plus connu de cette catégorie est le *Bruit de Worley* qui nous permet de générer des crevasses ou bien des cristaux (Cf. figure 18).

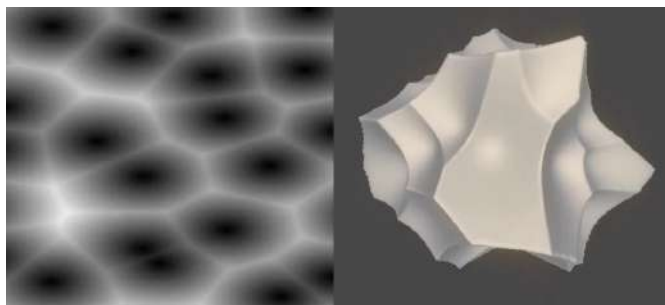


FIGURE 18 – À gauche le bruit sous forme d'image et à droite sa représentation.

Nous utilisons un dérivé du *Bruit de Worley* pour avoir une forme de terrain craquelé, asséché ou bien pour avoir des plaques tectoniques (Cf. figure 19).

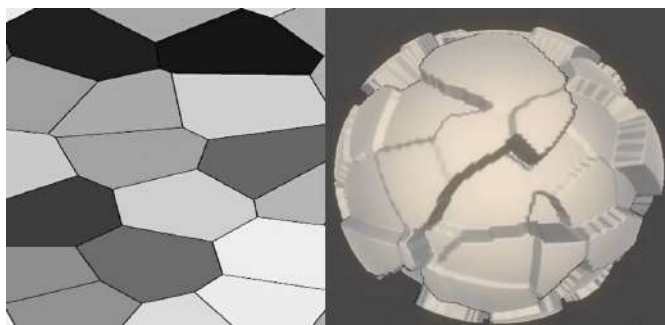


FIGURE 19 – À gauche le bruit sous forme d'image et à droite sa représentation.

Le deuxième type de générateur que nous avons mis en place se base sur des fonctions mathématiques. En effet, on peut tronquer une parabole en haut et en bas afin de créer des cratères (Cf. figure 20).

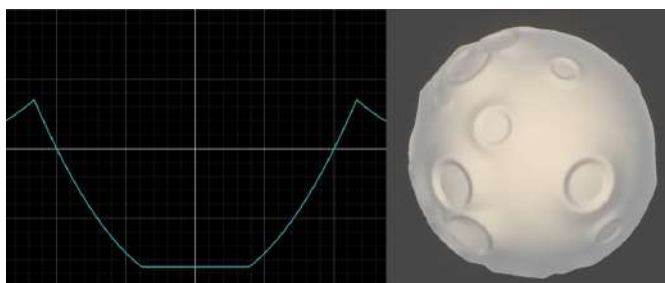


FIGURE 20 – À gauche la parabole tronquée et à droite des cratères générées.

On peut également modifier une courbe de Gauss pour créer des volcans (Cf. figure 21).

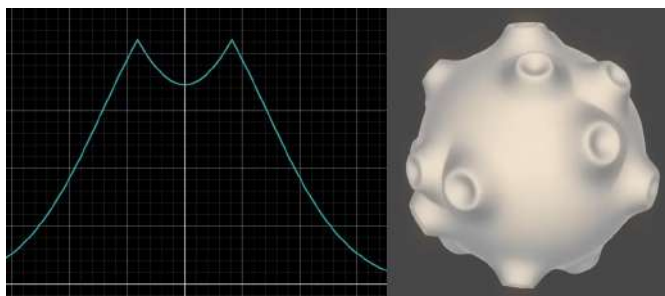


FIGURE 21 – À gauche la courbe de Gauss tronquée et à droite plusieurs volcans.

Ces différents générateurs ont pris beaucoup de temps à être codés, mais après un mois de travail nous avons pu nous lancer dans la réalisation des planètes.

3.6.3 Génération des planètes

Même si tous nos générateurs ont été codés, il a fallu passer du temps pour trouver les bonnes combinaisons et les bons paramètres jusqu'à obtenir le résultat souhaité.

Le premier biome de planète à avoir été mis en place est le biome calciné. Nous avons déjà beaucoup d'avance alors nous avons passé du temps à mettre en place des variantes de ce biome. Ces variantes sont la variante océan de lave qui peut ou non avoir des volcans (Cf. figure 22), la variante rivière de lave (Cf. figure 23) et la variante craquelée (Cf. figure 24). Pour ce biome, nous avons voulu que le joueur comprenne immédiatement que la température de la planète est très élevée et inhabitable.



FIGURE 22 – Océan de lave

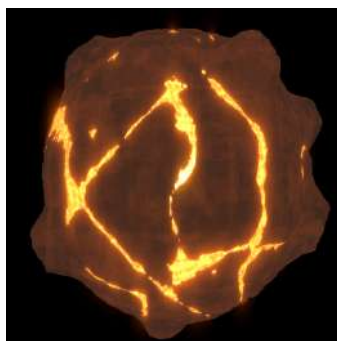


FIGURE 23 – Rivière de lave



FIGURE 24 – Craquelée

Après le biome calciné, nous nous sommes lancés dans la réalisation du biome mort. Nous avons essayé que les planètes de ce biome paraissent avoir subi beaucoup de dégâts par des astéroïdes. Nous lui avons également attribué une palette de couleur proche de celle de notre lune (Cf. figure 25).



FIGURE 25 – Trois générations différentes du biome mort.

Finalement, le dernier biome que nous avons mis en place pour cette première soutenance est le biome irradié. Nous avons un peu modifié ses caractéristiques par rapport à ce qui a été écrit dans le cahier des charges. En effet, nous avons rajouté des lacs d'acide verts pour que le joueur comprenne que cette planète est toxique. La palette de couleur à également été revue pour être différentiable du biome mort (Cf. figure 26).

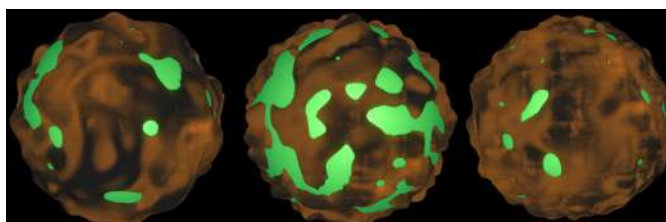


FIGURE 26 – Trois générations différentes du biome irradié.

3.6.4 Graphisme

Ayant de l'avance sur notre planning, nous avons mis en place des palettes de couleur propre à chaque biome ainsi que de nombreuses autres propriétés pour que chaque biome soit unique visuellement. Bien

évidemment, ces propriétés sont générées procéduralement, c'est-à-dire que chaque planète aura un aspect différent aussi bien au niveau de sa forme qu'au niveau de sa couleur et de son aspect global.

Nous avons aussi commencé à mettre en place des effets visuels, comme pour la lave qui semble irradiée de lumière.

3.6.5 Optimisation

Nos planètes ayant beaucoup de détails, il peut être très lourd pour un ordinateur peu puissant de les afficher. Nous avons donc mis en place un système de niveau de détail. En fonction de la distance à laquelle le joueur se trouve de la planète, la résolution de celle-ci va diminuer.

3.7 Génération des systèmes solaires

Une fois les biomes calcinés, morts et irradiés mis en place, nous avons créé un soleil et mis en mouvement nos astres autour de celui-ci. Les planètes tournent autour d'un soleil avec une vitesse constante et elles tournent également sur elles-mêmes.

3.8 Communication

Cette partie du projet a été réalisée par Lucas BURGAUD.

3.8.1 Site Web

Avant de créer un site web, il faut déjà se poser la question du moyen par lequel nous allons héberger le site. Grâce à notre statut d'étudiant, nous avons accès à l'offre étudiante de Github qui nous permet d'utiliser la version premium de ce dernier. Github est un site web qui permet aux développeurs de stocker et partager leur code en utilisant Git. Nous pouvons donc l'héberger pour stocker le code de notre site. Cependant sans payer, nous n'avons qu'accès à l'URL : <https://aziiborn.github.io>. Ce n'est pas très professionnel. Grâce à l'offre de Github, nous avons aussi accès à un URL gratuit parmi ceux proposés sur name.com. Nous nous sommes donc mis d'accord sur <http://42draconis.studio>. Il a suffi d'un fichier sur Github qui permet de faire renseigner l'URL, et de quelques manipulations de coordination entre Name et Github et l'URL était liée à GitHub.

Créer un site de A à Z peut être très long et très compliqué si on part de connaissances en HTML et CSS fragiles. La meilleure solution est donc d'utiliser un constructeur de site web, qui permettrait de le convertir en code HTML ensuite. Nous avons donc utilisé NicePage qui permet d'ajouter des blocs plus ou moins pré-désignés, des textes, boutons, images, etc.²⁷

Nous pouvons donc ensuite le convertir en code HTML et CSS et le mettre sur GitHub pour qu'il soit en ligne. Une première a pu être mise en ligne, ça a permis d'avoir un premier jet d'idée, et nous avons organisé une réunion pour se mettre d'accord sur éléments à afficher, et leur disposition sur les différentes pages. Cela a donné naissance à la version 2 du site.

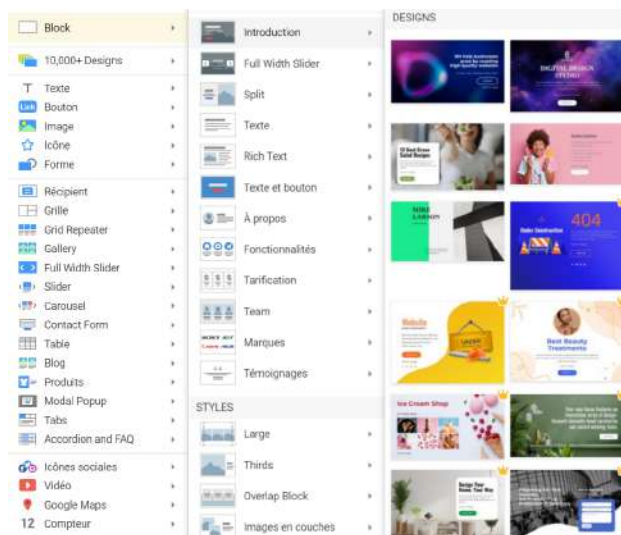


FIGURE 27 – Exemple de designs proposés sur NicePage.



FIGURE 28 – Exemple de la page téléchargement du site web.

3.8.2 Réseaux Sociaux

Pour faire connaître le projet à Epita Rennes, notre promotion, ou même au grand public, nous avons besoin de réseaux. Nous avons donc choisi de nous concentrer sur 3 réseaux : *Instagram*, *Youtube* et *Discord*

Pour *Instagram* et *Youtube* nous avons tout simplement créé une adresse mail et un compte Youtube *42 Draconis* et un compte Instagram *42.draconis*, l'objectif est de régulièrement mettre à jour les avancées du jeu sur ces réseaux. Pour *Discord*, nous avons créé un serveur Discord pouvant accueillir des dizaines de personnes, avec un système de vérification anti-bot, et un système de rôles.

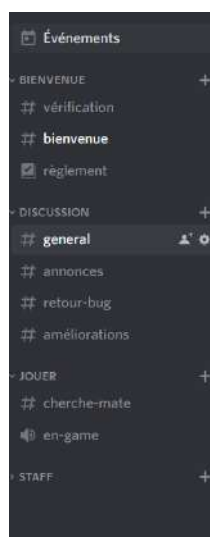


FIGURE 29 – Les salons textuels et vocaux.

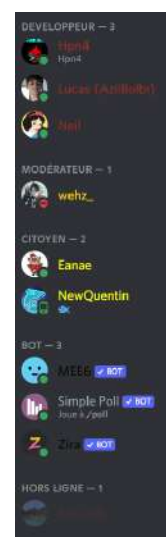


FIGURE 30 – Les rôles

3.9 Inventaire

Pour l'inventaire, l'objectif est de créer un inventaire pour chaque joueur et un inventaire pour le vaisseau, cette partie est développée principalement par Lucas BURGAUD. Cette partie a avancée, mais pas au niveau qui était attendu par le cahier des charges. Nous avons fait le choix de ralentir cette partie du projet pour plus travailler sur la partie site web, communication et musique. Il y a tout de même

une partie du côté front-end (ce qui va être vu par le joueur). Nous avons donc désigné les rendus des ressources qui seront visibles depuis l'inventaire.



FIGURE 31 – Les items désignés pour l'instant.

Pour la partie back-end, il n'y a pas de travaux concrets qui ont été faits, nous savons quels outils nous allons utiliser, et le tutoriel d'aide a été trouvé.

4 Ce qu'il reste à faire

4.1 Général

Pour la prochaine soutenance, nous allons regrouper tous les développements des différentes parties en un seul projet central. Cela sera réalisé à l'aide de l'outil Unity Collaborate, disponible directement dans Unity.

Il sera aussi bien sûr nécessaire de travailler sur les différentes parties déjà commencées, pour améliorer leur contenu et leur ajouter d'autres fonctionnalités.

Enfin, il faudra commencer à se pencher sur le système de sauvegarde du jeu dans tous ses aspects, que ce soit la capacité du jeu à enregistrer la scène en cours ou bien la possibilité de charger une scène à partir de fichiers.

4.2 Les ennemis et l'IA

4.2.1 Préciser les variables

Commençons par rappeler que, en ce qui concerne les ennemis, beaucoup de paramètres restent assez vagues. En effet, comme évoqué plus haut dans la partie *Ce qui a été fait*, les tests ont été réalisés dans un univers de développement, de test. Ainsi, pour l'instant, les rapports d'un ennemi avec ses semblables et le joueur sont purement spéculatifs. C'est pourquoi, pour le prochain rendu, il faudra se pencher sur cette question et établir une échelle pour les différents paramètres customisables des vaisseaux (l'accélération, la vitesse de rotation, les dégâts des projectiles, etc...).

Il faudra également préciser les systèmes graphiques comme choisir quels modèles de vaisseaux auront telles caractéristiques. Un tel est lent, mais tire des missiles, un autre est très rapide mais fragile... Les projectiles eux-mêmes demanderont un minimum d'attention. Nous espérons mettre en place au moins deux modèles de projectiles simples pour le prochain rendu, tels qu'un projectile normal style laser ou néon, ainsi qu'un gros missile ou bien encore une mine. Enfin, il faudra préciser la nature, la forme et la couleur des particules émises par un vaisseau endommagé. En effet, pour l'instant, celles-ci ne sont encore que des particules de test. Des systèmes définitifs devront être choisis.

Nous pensons ainsi réguler les différents paramètres de nos vaisseaux pour le prochain rendu, mais nous nous autoriserons une marge afin de tout finaliser pour le dernier rendu.

4.2.2 Nouveautés et améliorations

Parlons maintenant des choses déjà présentes, mais qui nécessitent une amélioration ou une révision. Nous ne pensons pas que les fonctionnalités actuelles des ennemis nous demanderont beaucoup plus de



FIGURE 32 – Un modèle de vaisseau.

travail. En effet, les bases sont posées, et à part de l'optimisation, tout est déjà assez solide.

Mais nous nous servirons de ces bases pour développer des comportements nouveaux, à commencer par la mise en place de plusieurs types de vaisseaux, comme nous l'avions imaginé et déjà mentionné auparavant. Nous envisageons pour le prochain rendu de développer aux alentours de 3 à 4 ennemis aux différents graphismes et caractéristiques.

- Le premier vaisseau serait un ennemi classique, équilibré, qui n'a pas beaucoup de vie mais qui fait quelques dégâts avec des projectiles relativement rapides tirés à une distance moyenne du joueur.
- Le second pourrait être une sorte de poids lourd : lent mais fort, il tirerait à une fréquence faible de gros missiles qui traqueraient le joueur à une certaine vitesse, qui permettrait tout de même à ceux-ci de l'éliminer avant le contact, qui serait dévastateur.
- Une troisième idée pourrait être un vaisseau lui-même projectile : son but serait de s'écraser sur le vaisseau des joueurs à grande vitesse en lui infligeant beaucoup de dégâts.

Ce ne sont que les premières idées de petits vaisseaux, mais d'autres pourront être ajoutées au fur et à mesure. Nous espérons donc réaliser ces trois différents vaisseaux pour la deuxième date de rendu, avec éventuellement un ou deux autres en plus.

Il faudra aussi se pencher sur les relations entre les vaisseaux : c'est pourquoi nous mettrons en place un système d'escadrille que nous avons déjà précédemment évoqué. Cela consiste à ce que plusieurs vaisseaux du même type assez proches se reconnaissent et aillent tous dans la même direction. Cela est intéressant au sens de limiter les collisions entre les ennemis lorsqu'ils poursuivent une même cible. De même, cela permettrait une éventuelle amélioration du système de traque. Si la cible traquée par un vaisseau est masquée par un objet, ce vaisseau pourrait communiquer avec ses "camarades" afin de



FIGURE 33 – Une idée de modèle pour un missile.

continuer à traquer la cible sans la "voir".

Cela reste bien sûr de la spéculation. L'objectif principal pour la prochaine date de rendu est de consolider les bases déjà présentes, mais surtout de lier ces ennemis aux joueurs et de les adapter à cette situation.

4.3 Ambiance et sons

Pour le prochain rendu, nous espérons avoir commencé à incorporer des effets sonores aux différentes parties de notre jeu, et éventuellement d'avoir inventé un système capable de jouer et enchaîner plusieurs musiques, inventées ou libres de droit.

4.4 Génération des planètes

Pour la prochaine soutenance, il nous faudra terminer tous les biomes. C'est-à-dire créer le biome gelé et verdoyant ainsi que perfectionner ceux déjà existants.

Une fois cela fait, nous nous lancerons dans la génération de détail et des ressources sur les planètes. Ces détails seront des petits cailloux, des cristaux ou pour la planète verdoyante, des arbres et des plantes.

Une dernière chose sera finalement d'ajouter des ceintures d'astéroïdes autour des planètes et potentiellement des lunes.

4.5 Génération du système solaire

Notre objectif est de pouvoir générer un système solaire entier en utilisant tous les biomes mis en place. Que chaque planète soit suffisamment espacée les unes des autres et qu'elles puissent tourner sans aucun problème.

4.6 Inventaire

Le but est que pour la prochaine soutenance, l'inventaire soit fonctionnel, pour rattraper le retard (même si on peut plus parler de choix que de retard), et avoir aussi le système de fabrication ("craft") fonctionnel. Le mieux serait d'avoir réussi à implémenter ces systèmes là dans la version commune.

De plus, pour être en accord avec le cahier des charges, nous voulons avoir fini la partie *craft*, c'est à dire la transformation d'objets en d'autres.

4.7 Graphisme

Pour la seconde soutenance, nous prévoyons d'avoir un logo du jeu, ainsi qu'une animation en boucle, qui peut être utilisée dans le système de chargement. Concernant les animations, nous voulons avoir une

animation du logo de *SpaceY* qui apparaîtra au chargement au lancement du jeu.

Nous comptons également revoir le graphisme de chaque partie, c'est-à-dire le vaisseau, les ennemis, les planètes et potentiellement les objets de l'inventaire. L'objectif est d'harmoniser les différents styles graphiques.

Nous essayerons également de mettre en place des effets graphiques en fonction des conditions. Par exemple un écran qui se givre lorsque le joueur rentre dans les capsules de cryogénéisation ou s'il reste trop de temps sur les planètes gelé. Ou encore un écran déformé par la chaleur sur les planètes calciné.

4.8 Communication

Pour la seconde soutenance, l'objectif est d'avoir pu mettre à jour régulièrement le site web, *Instagram* et *Youtube* avec les différentes avancées du jeu. Si nous avons réussi à créer une version téléchargeable (dans le meilleur des cas donc), nous ouvrirons le Discord Communauté au grand public.

5 Organisation et avancée

5.1 Organisation

Au début de cette période de développement, nous avons mis en place différents outils de communication et d'organisation, principalement Discord. Nous utilisons un système de sous-conversation qui permet de créer un espace de discussion pour chaque problème/sujet du moment pour pouvoir s'y retrouver.

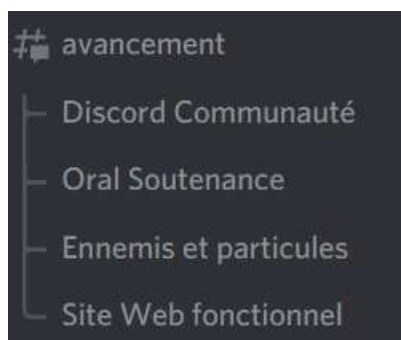
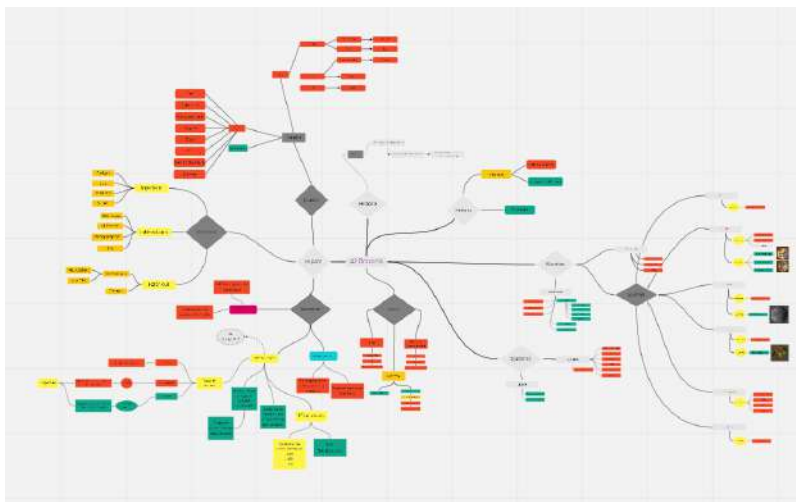


FIGURE 34 – Exemple des discussions du moment.

Mais *Discord* seul n'était pas suffisant pour obtenir une bonne organisation, d'où l'utilisation d'autres outils.

Premièrement, pour y voir plus clair sur le contenu de notre jeu, nous avons mis en place une carte mentale à l'aide de l'outil *Miro*, sur laquelle nous avons regroupé tout le contenu de notre jeu. Nous pouvons y rajouter des éléments quand des idées sont validées par le groupe. Pour chaque élément, il est possible de voir ce qui est fait et ce qui ne l'est pas grâce à un code couleur simple et visuel, rouge pour une tâche qui n'a pas du tout été travaillée, orange pour une tâche en cours, et vert pour une tâche finie.

FIGURE 35 – La carte mentale actuelle sur *Miro*.

Miro nous permet aussi de créer une note que l'on utilise pour proposer des tâches à faire prochainement, et où chacun peut piocher ce dont il se sent capable de faire prochainement. L'objectif est de ne pas oublier certaines tâches à faire, mais aussi d'optimiser le temps de travail. En effet, nous sommes des fois peu ou pas motivés pour travailler sur une certaine tâche, il est donc alors possible d'aller piocher sur une autre des tâches à faire et commencer des recherches dessus. Cela évite aussi le travail en doublon, pour ne pas perdre de temps inutilement.

Tâches à piocher :

(Cocher la case si vous décidez de prendre la touche et indiquer votre nom à côté)

Général :

- Conditions des planètes : barre de résistance de la combinaison (mort après trop de temps)
- Gestion des dégâts et de la mort (résurrection)
- Equipements du joueur : sonde, sac à doc...
- Système de sauvegarde du jeu. ETIENNE
- Gestion des loots (des ennemis, ressources sur les planètes, structures). Ressources pur, transformé, draconis, plan de construction ou histoire

Graphique :

- Effet graphiques pour les conditions de planètes : écran gelé, écran brouillé (chaleur), écran flou (asphyxie). ETIENNE
- Ecran de chargement (début du jeu et entre système solaire)
- Cinématique de début de jeu et de fin de jeu
- Barre de vie et d'oxygène

UI :

- Menu du début du jeu. ETIENNE
- Menu pour les options : graphiques et touches.

Autres :

- Trouver des assets pour les structures abandonnées BASTIEN
- Trouver assets de débris (pour les planètes et les vaisseaux ennemis) BASTIEN
- Discord Communauté. LUCAS
- Version 2 du site LUCAS

FIGURE 36 – Les tâches à piocher sur *Miro* en ce moment.

Ensuite, nous avons aussi décidé d'exploiter Excel pour établir un planning. Le but est que chacun puisse voir visuellement quelles vont être les tâches qu'il a à faire dans la semaine mais aussi de voir celles des autres membres de l'équipe. C'est une aide mentale pour visualiser le travail futur et nous rassurer sur le temps qu'il nous reste, mais aussi un support mental pour éviter d'être surchargés mentalement à penser à toutes les tâches qu'il faut faire.

Semaine du 01/02	Semaine du 07/02	Semaine du 14/02	Semaine du 21/02 (vacances)	Semaine du 28/02 (midterms)	Semaine du 07/03 (soutenance 1)
Générateurs de craquezture (Worley noise)		Générateurs de cristaux		Générateurs de cristaux	
calcinés	Variantes planètes calcinés	Planètes mortes	Planètes mortes	Planètes irradiés	Planètes irradiés
Shader + biome pour planètes calcinés + Bloom pour lave		Mouvement des planètes (rotation autour soleil + autour d'elle-même)		Shader planètes mortes	
		Système de LOD	Amélioration Mesh collider		
				Résoudre les bugs	
Creation de l'intérieur	Fusion Intérieur/Extérieur		Renouvellement de l'intérieur		
Modèle de joueur			Mouvement vaisseau qui bouge		
Animation des portes			Jetpack		
Ajout interface du joueur	Ajout base UI dans le vaisseau	Ajout console SAS			
					Mise en place
	Recherches logiciels/possibilités site	Creation du site web	Poster Site web	MAJ Site Web	logo 42draconis / animation logo
nouveau logo spaceY			Creation de la boutique de son		
Amélioration des scripts		Optimisation des scripts	Damage / Health / Collision System		
	Ajout d'un raycast	Meilleur système de tir	Moving System avec Rigidbody	Premier vaisseau - Normal	Liens avec le joueur et multijoueur
	Début de travail sur vaisseau		Modèles de Star-Sparrow importés		Meilleurs projectiles & meilleur comportement
					Optimisations diverses

FIGURE 37 – Une partie du planning Excel.

5.2 Avance sur le planning

Comme expliqué un peu plus haut, nous avons pris un peu d'avance sur notre planning posé initialement. Cela est dû aux efforts soutenus de chaque membre de l'équipe, qui ont su surmonter les nombreuses difficultés de développement qui se sont trouvées sur leur chemin. En mettant à jour régulièrement notre planning et en communiquant bien entre nous, nous comptons garder cette avance et cette motivation tout au long de ce projet et ainsi réaliser le plus de progrès possible !

5.3 Retard sur le planning

Aucun d'entre nous n'a eu de retard sur le planning à la date de cette soutenance. Il y a quelquefois eu des retards ou plutôt une mauvaise estimation du temps nécessaire à la réalisation d'une tâche, mais ils ont pu être rapidement comblés.

6 Conclusion

De manière générale, nous sommes plutôt fiers du chemin parcouru jusqu'à présent. Faire un jeu de A à Z en autodidacte est loin d'être une tâche facile, mais nous nous en sortons et nous sommes toujours motivés à produire plus. Satisfaits de notre production durant cette première période de développement, nous sommes plus que jamais déterminés à poursuivre notre travail sérieux pour la prochaine soutenance, et ainsi nous diriger vers un jeu dont nous serons fiers.

7 Bibliographie

<https://www.overleaf.com/project/621bd95d5b6a481d6bf54000> UppBeat : <https://uppbeat.io>

Lasonothèque : <https://lasonotheque.org/>

Miro : <https://miro.com>

Discord : <https://discord.com>

Nicepage : <https://nicepage.com>

Github : <https://github.com>

Name : <https://name.com>

Modèles et textures des ennemis : Unity Asset Store / Star Sparrow Modular Spaceships