

Table des matières

1	<u>Introduction</u>	2
2	<u>Reprise du cahier des charges</u>	2
3	<u>Avancées</u>	3
3.1	Gestion du joueur (Mathieu)	3
3.2	Gestion IA (Dorian)	3
3.3	Caméra (Mathieu)	4
3.4	Réseau (Mathieu)	4
3.5	Collisions (Victor)	5
3.6	Modèles 3D (Victor)	5
3.7	Ambiance Sonore/Musiques (Dorian)	6
3.8	Menus (Dorian/Victor)	9
3.9	Environnements (Dorian/Mathieu)	9
3.10	Animations 3D (Victor)	10
4	<u>Avances et Retards</u>	11
5	<u>Prévisions d'avancement</u>	13
6	<u>Ressenti du projet</u>	15
7	<u>Conclusion</u>	17
8	<u>Annexes</u>	18

1 Introduction

La prise en main d'Unity, de ses outils et de son environnement 3D a constitué une bonne part de cette première phase de développement de notre jeu, Synth Patrol. Elle nous a permis de mettre en perspective et de concrétiser certaines de nos idées décrites dans le cahier des charges, de trouver de nouvelles fonctionnalités à ajouter mais aussi de nous rendre compte de la difficulté d'implémentation de certaines. Ainsi, les fonctionnalités implémentées à ce jour permettent un fonctionnement simpliste du jeu, ouvrant la possibilité à certaines améliorations mais surtout permettant une aisance de développement futur, posant des bases solides sur lesquelles le jeu pourra s'étoffer. Pour l'instant, aucune des implémentations effectuées n'est problématique, même si elles méritent quelques améliorations, mais nous n'excluons pas la possibilité de supprimer ou de modifier certaines fonctionnalités si elles nous posent problème par la suite.

2 Reprise du cahier des charges

Depuis le cahier des charges, nous avons effectué quelques menues modifications notamment au niveau de la répartition des tâches : c'est dorénavant Dorian qui se chargera de la création des environnements, toujours suppléé par Victor et Mathieu qui suppléeront les animations 3D. Ces changements sont dus à une charge de travail plus importante que prévue sur le réseau et la gestion des joueurs. Dorian ayant alors pris l'initiative de faire des recherches sur les moyens de création de certains des environnements, mais ayant aussi, se chargeant de la musique, une meilleure vision de ce que pourrait être le design des niveaux, nous avons décidé d'un commun accord de le laisser se charger de la création des environnements 3D. Le changement de suppléance des animations 3D est alors un moyen de rééquilibrer la charge de travail entre les différents membres du groupe.

Pour ce qui est des changements relatifs aux fonctionnalités du jeu, nous avons décidé de rester sur les exigences que nous nous sommes fixées dans le cahier des charges, d'une part car nous avons réussi à implémenter une bonne partie des fonctionnalités que nous avons imaginé pour notre jeu mais aussi car nous ne voyons pour l'instant aucune contrainte majeure rendant impossible l'implémentation des futures fonctionnalités fixées.

3 Avancées

3.1 Gestion du joueur (Mathieu)

Je me suis tout d'abord attelé au déplacements du joueur car c'est l'élément primordial du jeu. La première difficulté a été de comprendre comment appliquer des forces sur les coordonnées locales du joueur. Pour cela j'ai utilisé tout d'abord `AddLocalForce` de `Rigidbody` mais je me suis rendu compte que les collisions ainsi que le système physique des `rigidbody` n'était pas adapté pour ce que nous voulions faire. J'ai donc choisi d'utiliser plutôt un `CharacterController` qui permet un meilleur déplacement du joueur ainsi qu'une facilité de rotation directement sur le transform, ce que ne permet pas `Rigidbody`. Mais ceci a créé d'autres problèmes comme la boîte de collision du vaisseau, qui alors n'est pas adaptée au modèle 3D du vaisseau. Dans la mesure où la plupart des collisions se feront au centre du vaisseau, cela ne pose pas de problème majeur mais reste une correction à amener.

Pour les rotations du vaisseau, nous avons choisi de les activer par des triggers placés sur le trajet du joueur car cela sera plus facile pour synchroniser les virages du joueur avec la musique. La difficulté avec les rotations a été la compréhension du comportement des quaternions et de ses méthodes pour faire exactement ce que l'on voulait sur le jeu.

3.2 Gestion IA (Dorian)

Le joueur peut lancer un laser devant lui afin de tuer les ennemis qui apparaissent devant lui. Les ennemis apparaissent devant le joueur à des moments clés de la musique du niveau afin d'immerger le joueur dans l'univers. Nous avons ajouté un temps minimum entre deux tirs afin que quand le joueur appuie très vite ou laisse le bouton de tir appuyé, le vaisseau ne tire pas des lasers à l'infini. Nous avons envisagé que le joueur puisse accéder à plusieurs styles de lasers ayant différentes propriétés tels que la puissance de tirs ou la dispersivité du tir. Ces différents lasers pourraient être débloqués à la fin de chaque niveau et seraient accessibles par la suite par le joueur. Ils auraient tous leur utilité face aux différentes manières dont les ennemis peuvent apparaître. Le style de tir que le joueur sélectionne serait affiché à l'écran par la couleur de l'effet d'électricité située sur les ailes du vaisseau que le joueur contrôle. Cette idée sera juste un bonus que nous ajouterons si nous avons le temps.

Les ennemis peuvent aussi tirer des lasers (de couleur différente afin d'évi-

ter la confusion pour le joueur). Ils sont placés à certains moments du niveau, le joueur doit les éliminer ou éviter les tirs jusqu'à ce que le joueur les ait devancées (les ennemis disparaissent au bout d'un certain temps, ils s'arrêtent juste d'avancer dans la même direction que le joueur et ils sont donc dépassés par le joueur qui continue d'avancer dans le niveau). Les ennemis tirent simplement devant eux sur l'axe où le joueur avance. Les ennemis n'ont pas encore été complètement implémentés. Pour l'instant, ils sont fixes et tirent juste dans une direction sans s'arrêter.

La position des tirs et leurs directions m'as causé quelques problèmes. Je suis d'abord parti sur l'idée que les tirs devaient être directement liés au vaisseau du joueur et donc être en fils dans Unity mais ce choix pose quelques problèmes car les tirs possèdent les mêmes coordonnées en X et Y que le joueur. Si le joueur tire puis se déplace, nous voyons que les tirs se déplacent avec le vaisseau du joueur. Je prévois donc de modifier la manière dont ils sont générés dans la prochaine soutenance.

3.3 Caméra (Mathieu)

Quand il a fallu s'atteler à la caméra, plusieurs choix s'offraient à nous : faire une caméra fixe qui ne suivrait le joueur que lorsqu'il sortirait de son champ de vision, ou alors fixer la caméra sur le joueur. Nous avons finalement opté pour la seconde solution, à la fois plus simple à mettre en place, car il suffit d'un script qui replace la caméra à chaque fois qu'elle a besoin de bouger sur un point fixe derrière le joueur en suivant les mêmes rotations que lui, mais aussi plus lisible pour le joueur, qui voit alors exactement où il tire. Mais placer la caméra juste derrière le joueur n'était pas très lisible, c'est pourquoi j'ai décidé de placer la caméra de sorte à voir le joueur de haut tout en le gardant au centre de l'écran. Cette rotation de la caméra a alors posé problème pour l'utilisation du script car la rotation de la caméra était alors faussée. Pour résoudre ce problème, j'ai alors utilisé la rotation locale de la caméra : en récupérant les rotations en angles d'Euler, il me suffit d'appliquer une rotation sur x à ma valeur voulue et de laisser les deux autres angles comme récupérés.

3.4 Réseau (Mathieu)

Pour pouvoir implémenter le réseau, j'ai créé une seconde scène indépendante de celle du jeu solo. L'infrastructure réseau implémentée est celle fournie par Unity. Elle permet aux joueurs de se connecter depuis une IP locale

sur la scène. Lorsque le joueur lance une partie, l'interface réseau se charge de faire apparaître le joueur aléatoirement sur l'un des points de spawn définis. Une scène offline servant de lobby pour la connection des joueurs est aussi prête à l'emploi.

3.5 Collisions (Victor)

Les collisions sont plutôt simples à gérer, elles sont presque automatiquement gérées par Unity. J'ai donc ajouté un tag pour certains objet, nommé `crashTrigger`, qui, lorsque l'objet touchera un obstacle portant ce tag, stoppera l'Update et le `fixedUpdate` du script `ControlPlayer` et affichera un écran de Game Over. Lorsque cet écran sera affiché, il y a la possibilité de reload la scene en appuyant sur R.

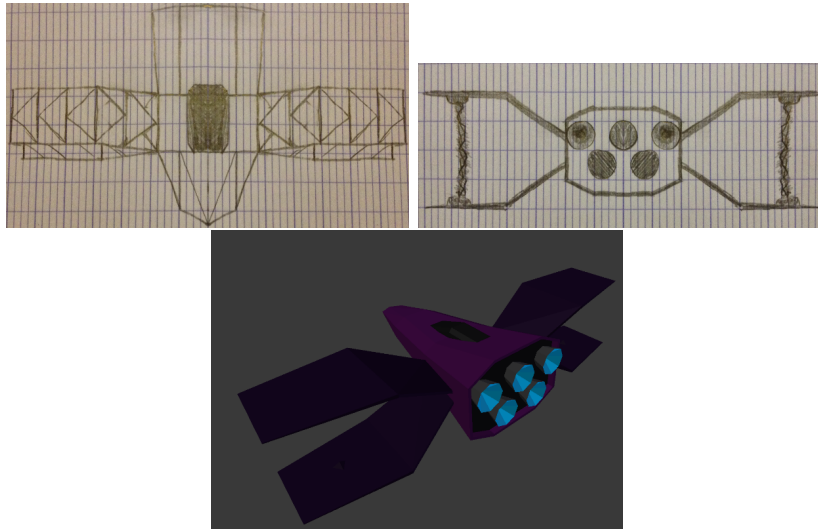
Le principal problème ici fût le reload de la scene. En effet, après cette action, la lumière du niveau était tout simplement supprimée. Après plusieurs tentatives de résolutions du problème et quelques recherches, il s'est avéré que ce bug est dû à l'éditeur Unity, et que la désactivation du calcul automatique des lumières corrige ceci. Cependant il sera nécessaire de recalculer les lumières lors de chaque ajouts ou modification des lumières, mais cela est faisable en 2 clics, et est corrigé après un build, rendant le problème insignifiant.

3.6 Modèles 3D (Victor)

Mon objectif pour la première soutenance a été fixé à 10%. Nous devions avoir au moins un vaisseau de modélisé. J'ai donc imaginé et dessin un vaisseau en s'inspirant notamment de vaisseaux spatiaux tirés de Star Wars, Star Fox, ainsi que F-Zero, tout en gardant un esprit dit low-poly (nous pouvons notamment citer les boosters se situant à l'arrière du vaisseau, qui comporte une base octogonale).

Des couleurs sombres proches du violet et du bleu ont été choisies afin de rester dans l'esthétique globale du jeu. Lors de l'ajout du modèle sur Unity, les matériaux ont pris une couleur beaucoup plus sombre qu'initialement prévue. Des couleurs plus claires sont donc à prévoir.

Pour créer le vaisseau, J'ai modélisé la moitié du vaisseau puis j'ai ajouté un composant miroir au modèle afin de faire l'autre moitié. Le modèle vu de derrière respecte le dessin original, mais pour la vue de dessus, le dessin original ne convenait pas à nos attentes, Je suis donc arrivé au modèle final par succession de changements mineurs, allant de l'allongement du nez du



vaisseau au repositionnement et à l'agrandissement des ailes.

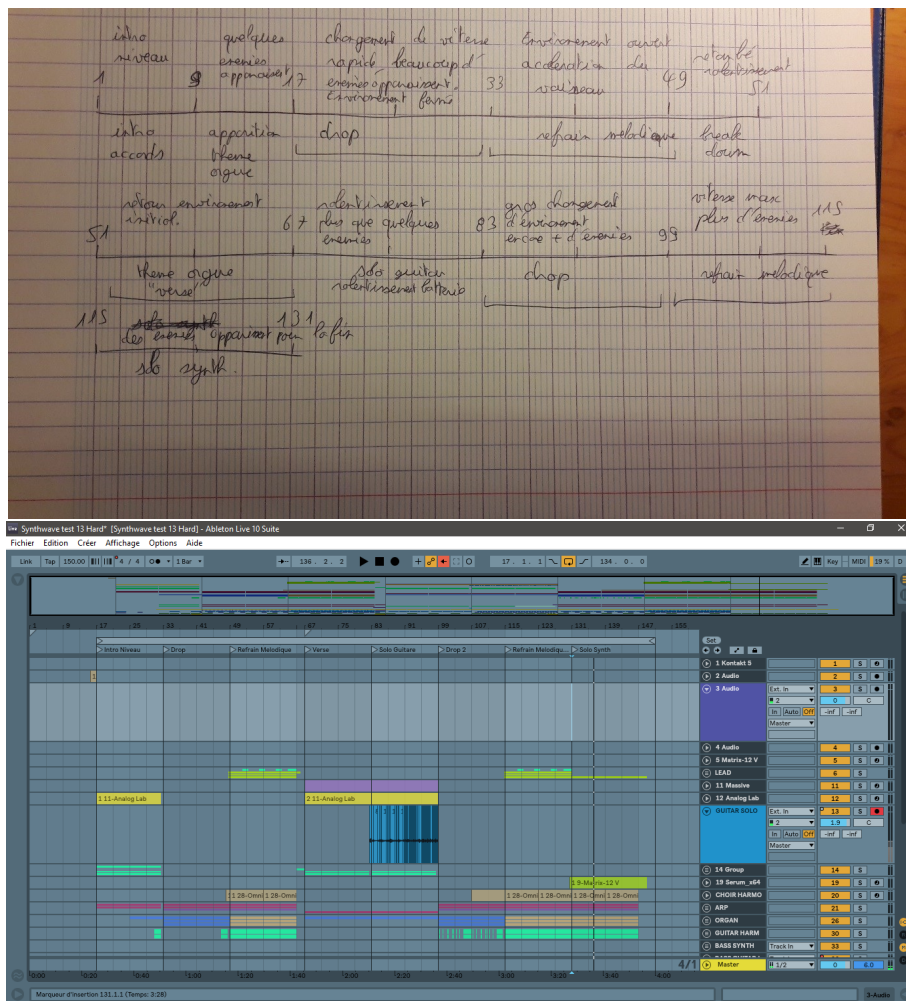
Un problème rencontré lors de l'importation sur Unity fut que certaines faces n'avaient pas l'air d'être rendu. Il s'avérait en fait que ces faces étaient à l'envers sur le modelé, un simple A – N en mode Edit réglait le tout. Un autre problème fut que les scripts de déplacements créés par Mathieu utilisaient les coordonnées locales, le vaisseau se retrouvait donc à avancer devant lui tout en pointant vers le haut. Une rotation sur Unity ne pouvait pas corriger ce problème, mais une solution simple était possible par le biais de Blender, en mettant un scale de -1, le vaisseau se retournait sans rotation et les axes locaux étaient donc placés correctement.

Deux petites pyramides ont été placées de chaque côté du vaisseau qui permettront un ajout futur d'électricité liant les deux ailes, à la manière d'un podracer de Star Wars I, par le biais de particules.

3.7 Ambiance Sonore/Musiques (Dorian)

J'étais aussi responsable des ambiances sonores du jeu. Comme nous l'avons dit dans le cahier des charges, nous nous inscrivons dans une esthétique années 80/Synthwave. La musique que je réalise se place donc dans ce style. A l'arrivée de la première soutenance, j'ai réalisé trois musiques pouvant être utilisées. Elles ne sont pas totalement finalisées mais leurs structures et

leurs réalisations sont bien défini. Deux musiques pourront être utilisé comme base pour des niveaux et la dernière pourrait convenir au menu principal. Les deux premières musiques ont été réalisé en pensant à une structure possible de niveau. J'ai retranscrit les passages d'un morceau en évènements pouvant intervenir en jeu. Les musiques possèdent des changements de dynamique (des changements de volume général et d'idée musicale) afin de retranscrire la diversité du niveau et que le joueur ne s'ennuie pas en écoutant la musique en jeu. Les musiques pouvant servir aux niveau ont un style assez agressif



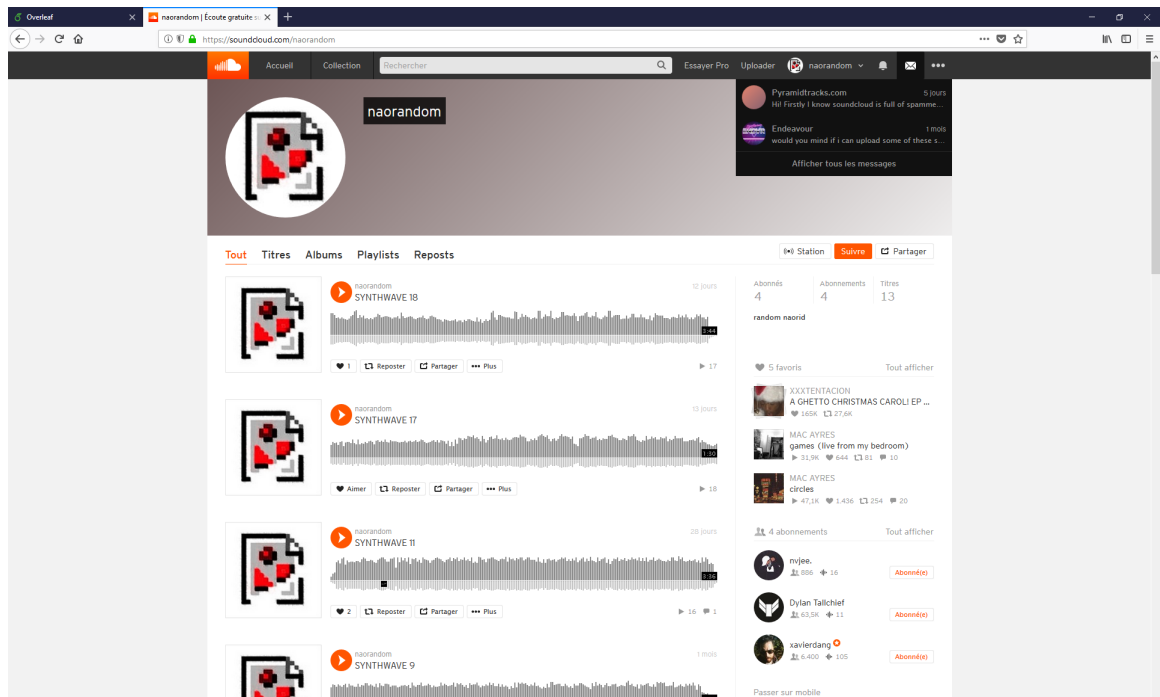
car j'avais en tête les niveaux de fin de jeu qui sont les plus difficile et les plus marquant. L'intérêt de lier la musique et le jeu aussi étroitement est d'inviter le joueur dans une ambiance particulière et une expérience intéressante.

La musique pouvant servir pour le menu principal a quand a-t-elle été

conçu comme une boucle pouvant être écoutée sans interruptions brutales. Je prévois de faire une musique introduisant le jeu (quand les logos apparaissent) menant ensuite sur cette boucle du menu.

Les musiques ne sont pas réalisées de façon linéaire, il est donc difficile d'estimer l'avancement de cette tâche car il reste pour toutes les musiques le Mastering du son afin d'améliorer la cohérence entre les différentes musiques et niveaux du jeu.

Avant d'arriver à ce résultat, j'ai effectué beaucoup de tests. Certains de ces tests pourront être réutilisés par la suite. Tous les morceaux de test et des versions Beta des dernières musiques sont disponibles sur mon Soundcloud réserve pour le projet : soundcloud.com/naorandom N'hésitez d'ailleurs pas à vous abonner et jeter un coup d'œil, plus de contenu arrivera par la suite !



L'ambiance sonore comprend aussi les différents effets sonores que l'on pourra entendre dans le jeu. Je prévois donc d'intégrer des sons de tirs de laser pour le joueur et les ennemis ainsi qu'un son du bruit vaisseau et les sons de sélections dans les menus. Nous envisageons que la hauteur du son du moteur soit liée à la vitesse du vaisseau afin d'ajouter un feedback montrant une accélération ou une décélération. J'ai réalisé pour l'instant des sons de

tirs de laser s'inscrivant eux aussi dans l'esthétique générale du jeu.

3.8 Menus (Dorian/Victor)

Après avoir commencé l'implémentation des collisions, nous avons décidé de commencer le travail sur les menus en créant un menu de Game Over apparaissant quand le joueur percute un élément mortel. Le menu est constitué d'un texte indiquant la fin de partie et un texte montrant que le joueur peut appuyer sur la touche "r" afin de relancer une partie. Ce menu est susceptible d'évoluer au cours du développement du jeu car il serait utile que le joueur puisse quitter le niveau en cours et revenir au menu principal. Nous avons choisi des polices d'écriture évoquant de l'esthétique Synthwave pour garder le joueur impliqué dans le style du jeu.



3.9 Environnements (Dorian/Mathieu)

Les environnements sont pour l'instant très sommaires puisque nous n'avons jugé utile que de faire des environnements de tests pour les déplacements. Ainsi, j'ai créé des tubes vides qui font office de guide pour le vaisseau et des virages comprenant les triggers entraînant les rotations du vaisseau. Nous avons aussi choisi de définir l'environnement de base avec un sol inatteignable par le joueur et l'horizon afin de déterminer l'ambiance finale du jeu et nous aider à mieux visualiser ce que sera le reste du jeu. J'ai utilisé un shader que j'ai modifié afin de pouvoir changer très facilement la couleur des lumières pavant le sol. Il sera donc possible de les changer à chaque niveau et même de modifier les couleurs en direct afin que l'ambiance évolue au cours

du niveau.

3.10 Animations 3D (Victor)

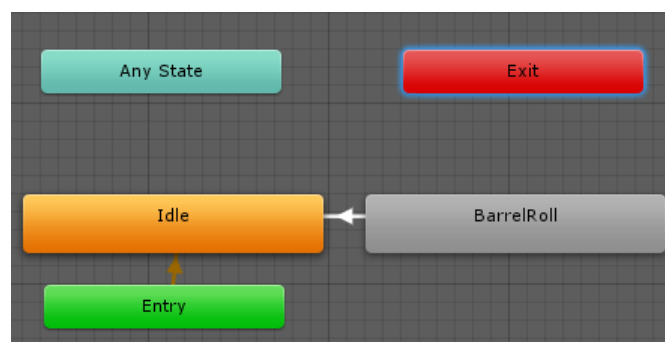
Cette partie a connu un départ difficile : en effet, la caméra était initialement liée au vaisseau pour faciliter son déplacement, mais cela compliqua beaucoup les choses pour les animations. Lors d'une rotation du vaisseau dans le cas d'une animation, la caméra entraînait elle aussi en rotation.

Un premier fix m'est venu en tête : celui de faire une rotation de la caméra mais opposé. Seulement, la caméra se situant légèrement au-dessus du vaisseau afin d'avoir un angle de plongé de 10 degrés, la caméra ne faisait bel et bien plus de rotation, mais faisait un petit déplacement circulaire autour de l'axe Z du vaisseau.

Il fallait donc fixer la caméra en la détachant du vaisseau, chose que Mathieu a réussi à faire peu après la rencontre de ces problèmes, utilisant un box invisible (plus de détails dans la partie consacrée à ce sujet). Maintenant la première animation, un barrel roll, fonctionne parfaitement.

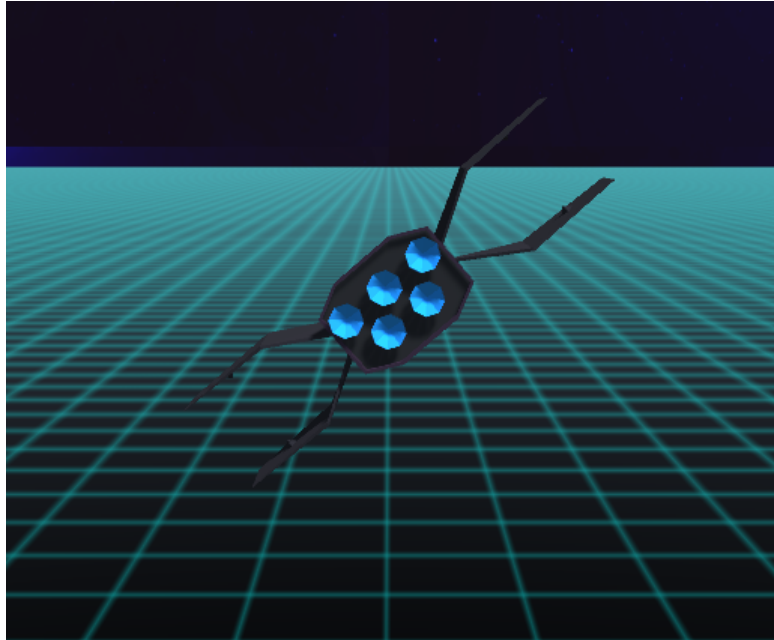
Malgré tout cela, un dernier problème était présent. L'animation ne pouvait s'effectuer que lorsque le Root Motion était désactivé, mais désactiver ceci reviendrait à empêcher de faire la rotation du vaisseau dans les virages. Un fix simple fut trouvé.

L'animation est déclenchée dans un simple script qui détecte l'appui de la touche B (temporaire), et lance ensuite l'animation, qui revient immédiatement à l'état initial après une exécution de l'animation.



L'animation utilise donc tous ces fixes, en appliquant l'animation inverse

sur la box invisible.



4 Avances et Retards

Nous sommes relativement satisfait de l'avancement de notre projet, qui concorde à peu près à ce que nous nous étions fixé lors de l'élaboration du planning. Néanmoins, certains domaines sont plus avancés que d'autres tandis que du retard a été pris sur d'autres.

— Gestion du Joueur :

Les avancements prévus sont respectés. En effet, comme le déplacement du joueur dans l'environnement 3D est l'élément principal de gameplay de notre jeu, nous avons prévu que les déplacements de base du joueur soient implémentés dès le début du développement. Ces déplacements basiques, correspondant à 80% de la charge de travail, sont donc terminés.

— Gestion IA :

L'avancement de cette partie n'est pas aussi bonne que ce que nous l'espérions. En effet, l'implémentation des tirs ayant pris plus de temps que prévu, nous avons choisi de laisser de côté l'implémentation des ennemis, avec leur spawn et leurs déplacements, entraînant un retard de 10% sur ce que nous avions prévu. Les ennemis ont été en partie implémentés mais nous n'avons pas encore fait de scripts faisant apparaître les ennemis dans des patterns particuliers. Ce retard devrait être rattrapé rapidement grâce à la meilleure compréhension du système de spawn acquise avec l'implémentation des tirs ainsi qu'à l'implémentation des déplacements du joueur.

— Gestion Caméra :

La gestion de la caméra a avancé plus rapidement que ce que nous avions prévu, dans la mesure où actuellement elle répond à tous les besoins que nous pourrions avoir sur le jeu, nous faisant penser que cette partie est achevée, prenant alors une avance de 20% sur ce que nous pensions effectuer pour cette première soutenance.

— Réseau :

L'implémentation du réseau est moins avancée que prévu car elle ne nous semblait pas essentielle pour l'instant et qu'elle nécessitait que la caméra soit implémentée ainsi que les tirs. Son implémentation nécessite aussi la création d'un script particulier au déplacement des joueurs, qui a pris du retard suite aux difficultés rencontrées lors de l'implémentation des déplacements du joueur dans la partie solo du jeu. Ainsi, nous avons pris un retard de 10% sur les estimations, qui devrait être rapidement comblé maintenant que les fonctionnalités nécessaires sont prêtes.

— Collisions :

L'essentiel des collisions élémentaires sont prêtes à l'emploi dans le jeu, suivant alors approximativement notre estimation de 80% des collisions implémentées pour cette première soutenance.

— Modèles 3D :

L'avancement sur cette partie est conforme à nos attentes. En effet,

seul le modèle 3D du joueur était nécessaire à ce niveau du développement du jeu : à la fois pour comprendre les contraintes de boîtes de collision d'un modèle 3D mais aussi pour mieux aborder l'ambiance que nous voulions créer pour notre jeu.

— Ambiance Sonore :

La partie amiance sonore et musique a beaucoup plus avancé que ce que nous avions prévus, Dorian ayant travaillé et finalisé des morceaux pour le jeu plus rapidement que ce qu'il avait lui même prévu, nous avons une avance de 20% sur ce que nous avions prévu sur le planning avec un total de trois morceaux et quelques bruitages.

— Menus :

Les menus étant pour l'instant une partie mineure à ce stade de développement, nous avons décidé d'y accorder moins de temps. Néanmoins, nous avons complété les 10% fixés avec la création du menu de Game Over et de redémarrage de partie.

— Environnement :

Afin de définir l'identité visuelle de notre jeu et à des fin de test, nous avons besoin d'un minimum d'environnement dans lequel le joueur puisse se déplacer, correspondant environ à 10% du total des environnements à créer. Cet avancement a été atteint avec la création du sol, de la Skybox et des quelques morceaux de niveaux créés.

— Animation 3D :

Nous pensons qu'à ce stade de développement, les animations 3D ne sont pas primordiales, c'est pour cela que nous avons fixé un avancement de 10%. Un avancement atteint avec l'implémentation du barrel-roll.

5 Prévisions d'avancement

Les bases du jeu étant désormais posées, le développement d'un niveau jouable pour la deuxième soutenance est envisageable mais nécessite l'implémentation de diverses fonctionnalités.

— Gestion du Joueur :

Implémentation du script de déplacement pour le multijoueur. L'apparition de bug liés à l'implémentation des animations est à prévoir, entraînant un travail supplémentaire de débogage.

— Gestion IA :

Implémentation du déplacement des ennemis et de leur spawn. Cela comprends un script permettant l'instauration d'un pattern pour les ennemis, d'un script qui génère des ennemis depuis un point donné dans l'espace ainsi que d'un trigger pour déclencher ce script.

— Réseau :

Implémentation d'un lobby de connection au serveur ainsi que des interactions entre les joueurs et de leur déplacements. Dans la mesure du possible, nous souhaiterions aussi personnaliser l'interface de connection au serveur de jeu pour coller mieux à l'esthétique de notre jeu et permettre une meilleure immersion du joueur.

— Collisions :

Au niveau des collisions, il reste à implémenter les collisions entre les tirs des ennemis et le joueur et celles entre les tirs du joueur et les ennemis ainsi que le système de points de vie. Pour cela, il faut encore déterminer les conditions exactes du Game Over qui est pour l'instant activé seulement lors d'une collision avec un mur plein.

— Modèles 3D :

Pour le bon déroulement d'un niveau de jeu, il sera nécessaire de modéliser les ennemis, de quelques éléments de décors qui permettront de mettre du relief sur la map de jeu tel que des bâtiments ainsi que de différents obstacles que le joueur pourra rencontrer sur sa route pour le gêner dans sa course et l'obliger à se déplacer.

— Ambiance Sonore :

La musique du premier niveau étant déjà prête, il nous restera à composer les bruitages pour les lasers ainsi que finaliser la musique du menu de selection des niveaux. Potentiellement, une musique pour un second ou un troisième niveau pourrait être créée, en fonction de

l'avancement des autres fonctionnalités dont Dorian à la charge.

— Menus :

Implémentation d'un menu principal de sélection entre les deux modes de jeux ainsi que d'un menu de sélection des niveaux. Il sera proche du design final même s'il restera une ébauche vouée à être finalisée pour la troisième soutenance.

— Environnement :

Création d'un niveau avec un environnement visible et un couloir fait de murs invisibles pour guider le joueur sur un rail. Toutes les actions du niveau seront synchronisées sur la musique selon un pattern précis.

— Animation 3D :

Animation des éclairs présent entre les deux ailes du vaisseau ainsi que l'ajout de particules permettant une meilleure immersion du joueur en lui donnant une plus grande impression de vitesse. Une amélioration de l'animation déjà présente pourrait être possible en décalant le vaisseau vers la gauche, afin de donner l'utilité initiale au barrel roll. La même animation sera créée mais avec une rotation vers la droite. D'autres animations tel qu'une animation pour les virages, ou une animation de demi-tour pour le multi-joueur

— Site Web :

Le site web sera créé par Victor. Ce site sera très sobre et inclura une brève présentation de la Team Gozzkra et du Projet Synth Patrol, un lien de download et éventuellement un dev-blog.

6 Ressenti du projet

Mathieu :

Le démarrage du projet a été plutôt lent au début : nous tâtonnions à la fois sur comment bien commencer le projet mais aussi sur comment aborder Unity. Pour cela, les tutoriaux intégrés dans Unity ont été d'une aide précieuse. Ils m'ont permis de bien comprendre les bases du moteur physique et ainsi de démarrer sereinement le projet. La documentation d'Unity

a aussi été d'une aide précieuse pour comprendre les différences par exemple entre RigidBody et CharacterController ou encore comment appliquer une force sur un axe local. Je pense que l'ambiance au sein du groupe est plutôt bon enfant, malgré quelques divergences lorsqu'un choix entraîne des difficultés pour l'implémentation d'autres fonctionnalités. Cette ambiance est par contre à double tranchant, une ambiance « cool » entraîne un relâchement au niveau du travail ce qui a impacté surtout la motivation au début : nous avions plein d'idée mais le sentiment que le temps ne nous manquerait pas. Cela nous a obligé à enchaîner quelques jours de travail intensif à l'approche de la soutenance. Personnellement, c'est une ambiance de travail qui me déplaît car elle induit un stress inutile, dans la mesure où le temps donné est plus que correct, mais aussi car elle entraîne une baisse de la productivité. Par exemple, nous nous sommes rendu compte après ce rush que l'implémentation de nos lasers fonctionnait visuellement mais que les lasers, étant fils du joueur, suivent tous ces mouvements. C'est pour ces raisons que je compte prendre des mesures pour éviter ce genre de rush pour la prochaine soutenance, en arrêtant par exemple de modifier les scènes UNITY une semaine avant la soutenance nous laissant alors un temps largement suffisant pour nous concentrer sur le rapport de soutenance et la préparation de l'oral.

Dorian :

J'ai personnellement pris un peu de temps à vraiment commencer à démarrer le projet car je me suis d'abord concentré sur les musiques du jeu. Cet élément m'avait beaucoup motivé au début du projet et du coup j'ai mis un peu de temps à vraiment travailler sur Unity. La collaboration s'est plutôt bien passée avec Mathieu et Victor mais nous avons dû finaliser une partie du projet à la fin de la période de rendu. L'ambiance était toujours agréable mais le fait de travailler peu de temps avant la soutenance a créé une tension plutôt problématique. Dès que quelque chose ne fonctionnait pas, nous avions tendance à nous reprocher le fonctionnement d'un autre élément. Après, quand nous avons commencé à travailler sur les assets, nous regagnions espoir sur la réussite du projet et cela améliorait l'ambiance générale.

Victor :

Malgré le démarrage du projet relativement lent, nous avons réussi à atteindre nos objectifs fixés pour cette première soutenance. De plus, cette première partie m'a énormément intéressé, particulièrement les modèles 3D et les animations 3D. Je n'avais jamais utilisé Blender auparavant, mais j'ai appris un soir en 2 heures et j'ai modélisé le vaisseau aussitôt. Cette partie

me plaît beaucoup et je compte faire tous les modèles 3D qui constitueront le jeu. La partie Animation 3D, bien que pouvant créer de nombreux problèmes, est aussi une partie très amusante, et très satisfaisante lorsque l'animation désiré fonctionne. Les collisions sont un peu moins intéressantes, dû au fait de la prise en charge automatique par Unity, mais néanmoins indispensables.

7 Conclusion

Le projet SynthPatrol a connu à ce stade du développement un démarrage plutôt lent mais nous avons su, pour la plupart des éléments intégrés dans le jeu, ce que nous voulions. Cette première partie du projet fût pour nous une première expérience de collaboration, et la majorité du temps dans une très bonne ambiance. Malgré un handicap présent dès le début (notre groupe n'est constitué que de 3 personnes, par chance nous savions cela depuis le tout début, nous permettant ainsi de gérer notre organisation plus facilement). L'ambiance générale du groupe était plutôt bonne même si la fin fut assez précipitée. Nous sommes satisfaits du résultat pour le moment car nous avons posé les bases du projet et nous sommes de plus en plus sur de la vision globale du jeu. Nous avons hâte de travailler sur la suite du projet afin de continuer la conception de ce jeu qui nous intéresse énormément.

8 Annexes

Tâche \ Personne			
	Dorian	Victor	Mathieu
Gestion Joueur	S		P
Gestion IA	P	S	
Gestion Caméra	S		P
Gestion Réseau		S	P
Gestion Collisions	S	P	
Modèles 3D		P	S
Ambiance Sonore	P		S
Menus	P	S	
Environnement	<i>P</i>	S	
Animation 3D		P	<i>S</i>
<i>P: principal S: suppléant</i>			

Tâche \ Soutenance			
	1	2	3
Gestion Joueur	80	100	100
Gestion IA	70	100	100
Gestion Caméra	100	100	100
Gestion Réseau	23	66	100
Gestion Collisions	80	100	100
Modèles 3D	10	50	100
Ambiance Sonore	50	70	100
Menus	10	30	100
Environnement	10	30	100
Animation 3D	10	50	100
<i>Valeurs en %</i>			