

RAPPORT DE LA DEUXIÈME SOUTENANCE
Projet flight arena

mr cube :
Vincent ROSPINI-CLERICI,
Guillaume REBUT
chef de projet : Arthur REMAUD

7 mai 2015

Sommaire

1	Retard/Avance par rapport au cahier des charges	2
1.1	Prévisions	2
1.2	Retard	2
1.2.1	Option changer de touches de commande	2
1.3	Avance	2
2	Travail par membre	3
2.1	Guillaume Rebut	3
2.1.1	Ce que Guillaume doit faire pour la soutenance finale	3
2.2	Vincent Rospini-Clerici	3
2.2.1	Création de nouveaux vaisseaux	3
2.2.2	Création du site internet	3
2.2.3	Ce que Vincent doit faire pour la soutenance finale	4
2.3	Arthur Remaud	4
2.3.1	Multijoueur en réseau	4
2.3.2	Quaternions	4
2.3.3	Intelligence artificielle	5
2.3.4	Menus	6
2.3.5	Ce qu'Arthur doit faire pour la soutenance finale	6
3	Pour la prochaine soutenance	7

Partie 1

Retard/Avance par rapport au cahier des charges

1.1 Prévisions

1.2 Retard

1.2.1 Option changer de touches de commande

Nous voulions rajouter dans le menu des options la possibilité de choisir ses propres touches sur le clavier pour piloter le vaisseau et faire en sorte que le jeu le retienne. Cependant, nous ignorons totalement comment le faire et aucun site internet ne parle de cela.

Comme Unity, de base, permet au démarrage d'assigner ses touches, et que cela n'est qu'un bonus pas nécessaire, nous avons donc abandonné notre idée de base.

1.3 Avance

Partie 2

Travail par membre

Nous allons vous décrire ce que chaque membre de l'équipe mr cube a fait pendant cette deuxième période, avec leurs difficultés rencontrées et les techniques utilisées.

2.1 Guillaume Rebut

2.1.1 Ce que Guillaume doit faire pour la soutenance finale

2.2 Vincent Rospini-Clerici

2.2.1 Création de nouveaux vaisseaux

Dans le soucis d'ajouter du contenu, deux nouveaux vaisseaux ont été créés sur Blender et textures sur Photoshop. Il s'agit tout d'abord d'un vaisseau qui ferait penser à un avion militaire tout droit venu du futur. L'autre est inspiré du P40, un avion militaire américain utilisé durant la seconde guerre mondiale et auquel Vincent a rajouté des nacelles de missiles et des réacteurs pour lui donner un air à la fois futuriste et ancien. Deux textures ont été créées pour ce dernier vaisseau mais nous nous servirons pour le moment d'une seule d'entre elle pour la première soutenance.

2.2.2 Ajout des particules des vaisseaux

Les flammes sortant des réacteurs des vaisseaux ont été rajoutées par le biais de la création de particules qu'offre Unity. Tout d'abord, Vincent a rajouté des éléments de particules pour créer juste des flammes sortant des réacteurs, mais le groupe s'est mis d'accord pour que les réacteurs laissent une trainée après le passage du vaisseau. Cependant, les particules étant contenu dans le *gameobject* du vaisseau, les flammes tournaient lorsque le vaisseau tournait, au

lieu de simplement suivre sa trajectoire. Nous avons après quelques recherches sur internet nous avons trouve le parametre permettant de le faire. Il s'agissait de definir l'espace de simulation au monde et non à l'objet.

2.2.3 Création du site internet

Vincent s'est occupé de la création du site internet en vue de la seconde soutenance. Il a décidé de créer le site en langage *HTML* par le biais des logiciels d'adobe, Photoshop et DreamWeaver. Ce dernier est un logiciel permettant d'à la fois modifier directement le code du site et de s'occuper de son coté graphique plus aisément. Il a voulu créer un site avec un design sombre qui rappellerait un ciel étoilé avec de nombreuses images des vaisseaux créés afin de le rendre attirant. Les pages du site ont été désignées sur Photoshop avec des images créées spécialement pour le fond d'écran et pour embellir les pages du site. Le site est prévu pour avoir toutes ses pages traduites en anglais et en français en cliquant simplement sur le drapeau correspondant à la langue voulue. Le site contiendra une page d'accueil, une page pour présenter le jeu, une pour présenter l'équipe et une pour que les visiteurs puissent avoir accès au cahier des charges, à l'exécutable du jeu, et peut-être à d'autres contenus à venir.

2.2.4 Ce que Vincent doit faire pour la soutenance finale

2.3 Arthur Remaud

2.3.1 Multijoueur en réseau

Pendant cette période, Arthur a surtout travaillé sur le multijoueur. Il a d'abord essayé de faire un protocole UDP pour relier en LAN (Local Area Network) en s'inspirant du TP que nous avons fait dans les cours habituels lorsque nous avons travaillé sur le protocole TCP, avant de se rendre compte qu'il existait la classe *Network* sur Unity qui simplifie grandement l'élaboration d'un mode multijoueur sur un jeu. Plusieurs tutoriels existent à ce sujet sur internet, Arthur s'en est donc inspiré pour faire un réseau.

Maintenant, le joueur peut choisir d'héberger une partie en LAN ou de rejoindre une partie déjà existante. S'il crée sa propre partie, il s'affiche alors dans un coin son adresse IP locale pour qu'il puisse la donner aux joueurs souhaitant le rejoindre. Ces derniers doivent d'abord saisir l'adresse IP pour rejoindre une partie, et ensuite leur vaisseau apparait dans la partie pour commencer à jouer. Cette technique existait pour certains jeux comme *Age of Empire I* pour pouvoir jouer en LAN.

Les règles du jeu sont les mêmes que dans le mode un joueur ou écran séparé.

Le principal problème fut qu'au départ, les joueurs contrôlaient le vaisseau de l'autre joueur et donc devait regarder sur l'autre écran pour jouer. De plus,

à trois joueurs, les deux premiers connectés voyaient un seul et même vaisseau qu'il contrôlaient tous les deux pendant que le troisième joueur en pilotait un autre. Le dernier vaisseau quant à lui n'était contrôlé par personne et restait immobile. Au final, ce problème venait de l'assignation des caméras et des scripts aux joueurs lorsqu'ils instancient un nouveau vaisseau en arrivant.

2.3.2 Quaternions

Il nous avait été demandé lors de la dernière soutenance de modifier les déplacements des vaisseaux en rajoutant des quaternions pour faire les rotations des vaisseaux. Comme c'était Arthur qui s'était chargé des mouvements à la base, c'est lui qui a modifié le code pour mettre à la place des quaternions. Maintenant, les mouvements sont plus réalistes grâce à l'inertie mais cela rend le jeu un peu plus difficile.

Nous n'avions pas utilisé cette technique plus tôt car elle laissait les forces de collisions sur le vaisseau qui le faisaient déplacer après avoir touché un immeuble. En effet le *rigidbody* du vaisseau conservait les collisions subies et donc le rebondissement du vaisseau sur les objets. Dès que le joueur touchait un immeuble, le vaisseau rebondissait et cette force ne s'annulait que très lentement. Nous avons remédié à ce problème en mettant à zéro les déplacements du vaisseau dans la fonction *Update* si le joueur ne bouge pas.

Parfois, lorsque le vaisseau tournaient longtemps sur lui-même, il se mettait soudainement à partir dans l'autre sens avant de repartir de la rotation voulue. En effet le quaternion dépassait les 180° et donc la rotation s'inversait. En ajoutant un maximum à l'accélération, on a pu régler facilement ce petit imprévu qui rendait le pilotage imprévisible.

2.3.3 Intelligence artificielle

Arthur a commencé à faire une intelligence artificielle pour pouvoir jouer contre des vaisseaux contrôlés par l'ordinateur. Pour le début nous voulions faire un algorithme de *pathfinding*. C'est un algorithme qui permet de calculer la trajectoire la plus courte d'un point A à un point B en évitant les obstacles, et donc permet à une intelligence artificielle de se déplacer facilement. Cependant la 3D nous a posé problème, car on ne peut représenter facilement le terrain sous forme de tableau. Il faudrait utiliser un graphe, mais non seulement nous ne savons pas comment le faire, mais en plus il faudrait en faire un différent pour chaque carte, et nous n'avons pas eu le temps.

Nous avons donc opté pour une autre technique : devant le vaisseau, nous avons rajouté des cylindres invisibles qui détectent des collisions avec les murs. Lorsque cela se produit, le vaisseau ralentit puis tourne en conséquence pour ne

pas rentrer dans l'obstacle. Il y a quatre cylindre : un pour la gauche, un pour la droite, un pour le haut et un pour le bas. Ils sont placés de manière conique pour mieux détecter les immeubles. Le vaisseau circule donc entre les bâtiments qu'il détecte et évite. Nous avons remarqué que cela ne fonctionnait pas pour les objets qui avait pour *collider* un *Mesh Collider*. Nous avons donc rajouté des *Box Collider* à tous les bâtiments et aux limites invisibles du jeu pour qu'il ne fonce pas dedans, tout en veillant à ce qu'il ne crée pas de collision avec les autres vaisseaux.

Cette intelligence artificielle n'est pas encore optimale : elle se prend toujours des bâtiments et surtout elle ne cherche pas le joueur adverse pour le battre. Elle se contente de tirer en continue droit devant elle pendant qu'elle se déplace.

2.3.4 Menus

Le menu a été complété par Arthur. Il a rajouté la sélection de la qualité d'image dans le menu option pour pouvoir choisir en fonction de la performance de son ordinateur les meilleurs graphismes possibles.

Il a aussi fait en sorte que le joueur puisse choisir le vaisseau qu'il veut piloter parmi les différents proposés avant de commencer la partie, ainsi que la sélection de la carte en mode un joueur.

Au départ, nous voulions faire en sorte que le joueur puisse choisir la répartition des touches de son clavier s'il voulait changer ses commandes pour être plus à l'aise. Cependant, nous ignorons totalement comment le faire et Arthur n'a rien trouvé sur internet pour le faire, même dans la documentation de Unity. Nous avons donc abandonné cette idée, comme nous l'avons dit dans la partie de retard.

2.3.5 Ce qu'Arthur doit faire pour la soutenance finale

Partie 3

Pour la prochaine soutenance

bla

