

CAHIER DES CHARGES  
Projet flight arena

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

16 janvier 2015

# Sommaire

<b>1</b>	<b>Modification du cahier des charges</b>	<b>2</b>
<b>2</b>	<b>Retard/Avance par rapport au cahier des charges</b>	<b>3</b>
2.1	Retards . . . . .	3
2.2	Avance . . . . .	3
<b>3</b>	<b>Travail par membre</b>	<b>5</b>
3.1	Guillaume Rebut . . . . .	5
3.2	Vincent Rospini-Clerici . . . . .	5
3.3	Arthur Remaud . . . . .	5
3.3.1	Scripts . . . . .	5
3.3.2	Modélisation 3D . . . . .	6
<b>4</b>	<b>Pour la prochaine soutenance</b>	<b>9</b>

## Partie 1

# Modification du cahier des charges

Suite au départ de Nikolaš Miletic en S1#, nous avons dû modifier le cahier des charges pour réajuster le budget et répartir à nouveau les tâches entre les trois membres restant.

Au niveau du budget, en terme de matériels, nous sommes passé de 3800 € à 2100 €, car l'ordinateur de Nikolaš coûte 1700 €. Le budget total est donc passé de 5397,29 € à 3697,29 €. Le budget pour les logiciels n'a pas évolué.

Pour la répartition du travail, le script dont était chargé Nikolaš a été confié à Arthur, Guillaume s'est chargé entièrement du gameplay. Nous avons malgré tout pris du retard pour ce qui est du son et de la gestion des données.

Le son a été pris sur internet ou sur d'autres logiciels existant déjà, alors que nous avions prévu de le faire nous-même. A cause de la baisse de main-d'œuvre, nous n'avons plus le temps nécessaire de le faire, et nous préférons accorder plus de temps au jeu lui-même. La gestion des données dont devaient se charger Arthur et Nikolaš n'a quasiment pas été commencée, car nous avons voulu privilégier le script principal du jeu.

Pour les soutenances à venir, nous avons donc renoncé à faire le son, mais nous devons nous rattraper pour les données afin d'avoir au moins des sauvegardes. Nikolaš devait aussi s'occuper de l'I.A et du site web. C'est donc Guillaume et Arthur qui vont faire l'I.A, et le site web sera construit par toute l'équipe.

## Partie 2

# Retard/Avance par rapport au cahier des charges

Les retards que nous avons par rapport aux prévisions du cahier des charges sont principalement causées par le départ d'un membre. Si nous regardons ce que nous nous avions fixé par personne, nous sommes dans les temps.

### 2.1 Retards

**Son :** Nous avons pris la musique du menu principal du jeu sur internet au lieu de la faire par nos propres moyens : il s'agit de *Captain America March*. Nous ne ferons de toute manière pas la musique du jeu, car cela nous prendrait trop de temps. Pour le son des tirs, nous avons pris un bruitage du jeu Pinball sur Windows XP qui ressemble à un son de laser.

**Gestion des données :** Il n'y a pas pour l'instant de gestion des données, ou du moins pas visible pour le joueur. En effet, il n'y a pas de sauvegardes dans le jeu ou de munitions, et la vie du joueur ne s'affiche pas à l'écran. Le menu option ne comporte pour l'instant que la modification du volume du jeu, qui est cependant conservée même après avoir fermé le jeu.

### 2.2 Avance

**Gameplay :** Le gameplay est plus avancé que ce que nous espérions. Nous avons déjà un vaisseau opérationnel avec des contrôles qui fonctionnent assez bien. Nous avons modélisé plusieurs bâtiments qui offrent un environnement agréable et permet déjà de s'amuser à piloter dans le décor sans trop de difficultés.

**Script :** Nous sommes en avance sur le script du jeu. Unity contient énormément de fonctions prédéfinies qui aident grandement à générer rapidement du code en C#. Nous avons déjà toutes les fonctions pour contrôler le vaisseau du joueur et gérer les tirs et les dégâts générés par ceux-ci. Nous avons aussi bien avancé les menus, que ce soit au niveau du menu principal avec les options, ou au niveau du menu de pause dans le niveau.

## Partie 3

# Travail par membre

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

### 3.1 Guillaume Rebut

### 3.2 Vincent Rospini-Clerici

### 3.3 Arthur Remaud

Pendant cette première période, Arthur a fait tous les scripts en C# du jeu et quelques modélisations 3D rapides avec blender.

#### 3.3.1 Scripts

Tous les scripts en C# du jeu ont été fait par Arthur grâce à des tutoriels sur internet. La fonction *Input.GetKey()* de *unity* permet de lire tout ce que l'utilisateur entre sur le clavier. Il a donc enregistré les différentes saisies possibles du joueur, qui provoquent sur le vaisseau des translations avec la fonction *transform.Translate()* lorsque le joueur veut avancer ou des rotations avec la fonction *transform.Rotate()* lorsque le joueur veut tourner. Chacun peut donc aisément piloter le vaisseau pour éviter les obstacles. L'inertie du vaisseau est géré par une variable de déplacement qui s'augmente à force d'appuyer sur la touche d'accélération, et diminue dans le cas contraire.

Il a aussi mis une touche pour tirer avec le vaisseau. Dans ce cas *Unity* crée une instance de la balle (fonction *Instantiate()*) qui part droit devant. Elle disparaît automatiquement au bout de quelques secondes grâce à la fonction *DestroyObject()*, le temps d'au moins traverser le terrain, pour éviter d'avoir

trop d'objets à la fois qui partent vers l'infini, ce qui provoquerait un ralentissement du jeu.

La balle contient aussi un trigger qui s'active lors d'une collision. Si le joueur entre en collision avec une balle, alors celle-ci est détruite et le joueur perd de la vie. Lorsque le joueur n'a plus de vie, le vaisseau est détruit. Pour l'instant il ne fait que disparaître mais par la suite, il y aura une animation de mort.

La principale difficulté rencontrée fut la gestion de l'inertie lorsque l'objet est touché par le décor. En effet, il peut rapidement partir en vrille et n'est plus maniable. Après des recherches sur internet, il est apparu que le problème venait du rigidbody et c'est la vitesse de celui-ci qu'il faut remettre à zéro à chaque fois.

Il a aussi fait les menus, en utilisant les propriétés GUI (Graphical User Interface) de *unity*. Il a pris une police avec un effet de science-fiction nommée *airstrike*, et c'est lui qui a choisi la musique *Captain America March* pour le menu. Jouant de la trompette, il l'avait interprété dans un orchestre l'année dernière, mais pour plus de confort, on a mis la version officielle de la musique. Nous aimerions par la suite conserver ce genre de musique pour le jeu, avec des grands orchestres et des cuivres sonnant, comme on peut en avoir dans *Star Wars*. On peut aller dans un menu option qui, pour l'instant, ne permet que de changer le volume, mais ce volume est conservé lorsque l'on redémarre le jeu.

Il a aussi intégré un menu de pause pour pouvoir quitter le jeu.

Il n'a pas réussi à intégrer les animations d'explosions faites dans *unity*, car l'exportation depuis *blender* échoue. Il existe pourtant des fonctionnalités sur *blender* pour faire exploser un objet en quelques clics, mais

### 3.3.2 Modélisation 3D

Arthur a fait quelques objets rapidement sur *blender* pour le projet. Il a ainsi créé six vaisseaux et quatre bâtiments rapidement. Ils sont cependant assez laids car réalisés rapidement pour pouvoir faire les essais de physiques rapidement et ne seront donc utilisés que pour le décor ou en personnage non joué. Le vaisseau principal est celui fait par Vincent.

Il a utilisé les mêmes techniques que Vincent pour la modélisation, à savoir l'outil *mirror* pour avoir des vaisseaux symétriques par rapport à un axe, et l'UV-mapping pour faire les textures à partir du patron de l'objet.

Il a fait ses textures lui-même avec *photoshop* plutôt que de les prendre en ligne. Elles sont donc par conséquent moins détaillées mais plus personnalisées. Grâce à l'outil tampon de ce logiciel, on peut facilement recopier un motif de l'image d'un endroit à un autre, ce qui est très pratique pour recopier les vitres des immeubles, plutôt que de faire des copier/coller/déplacer. Il suffit juste de sélectionner le point de départ du recopiage de motif et de colorier la zone voulue. Il a aussi utilisé des dégradés sur une surface délimitée, qui sont plus

esthétiques que le remplissage de surface. Pour prendre une surface précise avec photoshop, on utilise l'outil baguette magique qui sélectionne une zone par sa couleur, comme le fait le pot de peinture de paint mais juste pour sélectionner. On peut après faire des modifications uniquement dans cette zone, ce qui est très utile avec l'UV-mapping qui délimite les zone à dessiner pour les textures.



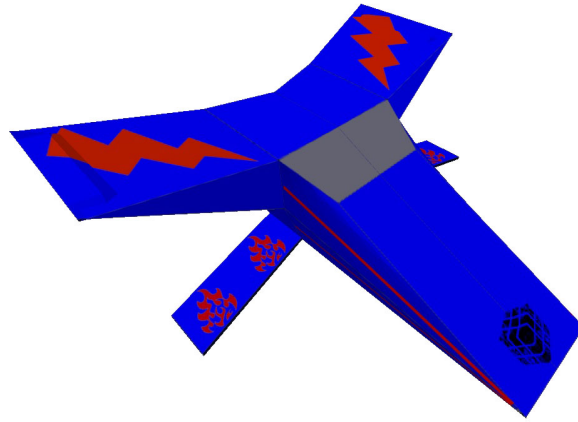


FIGURE 3.1 – L'un des vaisseaux fait par Arthur

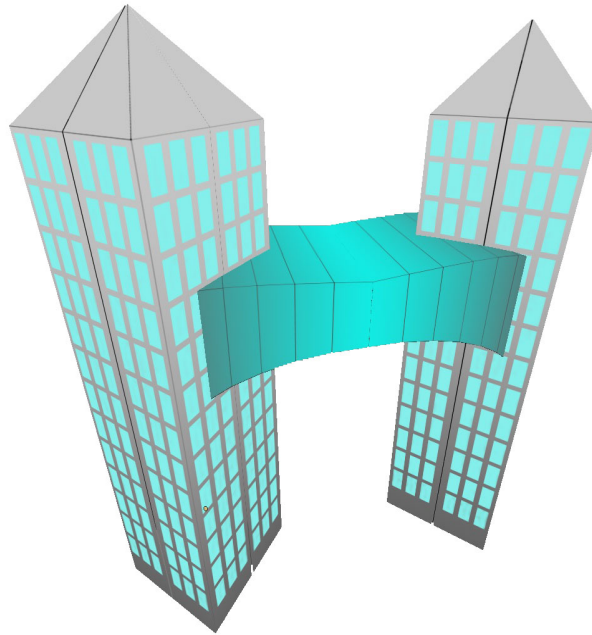


FIGURE 3.2 – L'un des immeubles fait par Arthur

## Partie 4

# Pour la prochaine soutenance

Tout d'abord, nous allons améliorer le contenu : nous allons rajouter des vaisseaux, des niveaux, des bâtiments...

Le menu sera aussi complété, notamment dans les options et dans le choix de son vaisseau pour que le joueur puisse plus facilement personnaliser sa manière de jouer.

Le site web sera commencé en HTML, pour pouvoir tenir au courant tous les fans qui suivent notre projet et veulent télécharger les dernières versions de notre jeu.

Mais surtout, la prochaine soutenance verra l'apparition des **I.A.** Avec cela, le joueur pourra jouer contre des ennemis gérés par l'ordinateur qu'il devra détruire, tout en essayant de ne pas se faire exploser par eux !

Et surtout, le **multijoueur en réseau local** sera commencé ! Il ne sera sans doute pas déjà opérationnel à 100% pour la prochaine soutenance, .