Game Design Document

A Space Simulator Game for the Oculus Rift Virtual Reality Device

Alex Flight

29/10/2013

TABLE OF CONTENTS

1.0 Game Overview

This game places the player in the role of a combat spacecraft pilot, the Oculus Rift causing their cockpit to envelop and surround them, creating a sense of immersion not seen before in games without VR technology. The player will take control of their spacecraft and fly high above a planet between asteroids, space structures and debris to seek out and destroy their enemies in the cold vastness of space.

1.1 Game Concept

The concept behind this game is to take full advantage of the Oculus Rift VR headset that can provide the player with an all-encompassing view on the game world, immersing them inside the game in a way that is impossible with traditional display technologies. The player will feel as if they are surrounded by their cockpit, feeling a sense of depth from the stereoscopic 3D and able to look around their surroundings freely using the motion tracking provided by the Oculus Rift.

1.2 Genre

Science-fiction Space Combat Simulator using Virtual Reality technologies.

1.3 Target Audience

Those interested in VR, the Oculus Rift and Space-game fans.

1.4 Game Flow

The player controls their spacecraft and head independently, the former with the keyboard, mouse and game controllers (if possible) and the latter with the tracking of the Oculus Rift. They attempt to move their ship through an environment consisting of asteroids and celestial objects to locate and destroy enemy targets using their lasers.

The player is guided in this task with a HUD style interface and consoles in their cockpit detailing the status of their ship, objectives and other information.

The player may experience the freedom and immersion of flying through space using the Oculus Rift and choose to continue to destroy enemy targets if they so wish.

1.5 Look and Feel

The game will use a typical space environment, focusing on the reproduction of a skybox filled with distance stars, a local star system and a nearby planet. Within the gameplay area will be asteroid-type objects that act as obstacles and hiding places for enemies.

Graphics styles will lean towards realism with simple 3D meshes and textures used for spacecraft to minimise on development time.

1.6 Project Scope

The game shall consist of at least one level which the player is given freedom to fly around and seek out a number of enemy targets. These enemies will consist of varying types from ships similar to the players to 'drones' with limited health. Artificial Intelligence may be provided to the enemies if project time allows.

The player will have control over one ship that has at least one weapon system that is appropriate for a spacecraft (such as genre-typical lasers).

The main focus of the project shall be the development of the core gameplay elements to facilitate the use of the Oculus Rift to immerse the player in the cockpit of their spacecraft.

2.0 Gameplay and Mechanics

Typical gameplay will involve the player using the Oculus Rift to look around their cockpit/surroundings much as one would in a present –day fighter aircraft or typical car. From this perspective the player may then use their mouse and keyboard (plus game controller if possible) to maneuverer their craft in any direction using their thrusters. The player will then be tasked with destroying enemy targets around the level using their weapons (lasers, missiles etc.)

2.1 Game Progression

Progression will be tracked on the players UI with a counter describing how many enemy targets they have destroyed and how many remain with their objective to destroy all targets on the level. Gameplay will be open-ended to allow for the player to continue playing even after their targets are destroyed. To this effect some kind of spawn mechanism may be introduced to allow more targets to appear for the player if desired.

2.2 Mission/challenge Structure

A fixed number of enemy targets of varying types will be included in the level and the player will be tasked with destroying these enemies by navigating to their locations within the environment, dodging asteroids and debris in the process. Enemy AI for dogfighting and evasion will be introduced if project scope allows.

2.3 Mechanics

Typical mechanics in the game revolve around two core concepts: movement in a space environment and combat using sci-fi weaponry.

2.3.1 Physics

Physics in the game will obey rules similar to that of a flight simulator and other space combat simulators. No gravity will be present and 6-degrees of freedom will be allowed by the player. A freely available physics engine will be utilised to minimise development time although modifications will be made to achieve the desired style.

If time allows a simplified Newtonian physics solution will be implemented that allows the game to follow the core concept of the three laws of motion such that objects will continue moving until an opposing force acts on them in some way.

2.3.2 Movement

Movement in the game will be accomplished by the player's spacecraft utilising the concepts of pitch, roll, yaw and also translations in directly in x, y and z directions (much like strafing in space with the added vertical dimension). To this effect the player will have control over their acceleration using their main engine with a limit to the maximum velocity their ship may travel as well as limits to the pitch, roll and yaw rates of their craft. Additionally the translation thrusters will offer a small amount of acceleration allowing the player to perform manoeuvres impossible in traditional aerodynamic flight.

2.3.3 Expanded Movement

If time allows the movement system will be expanded to follow a Newtonian physics system, allowing the player to behave exactly as they would in a 'real' space environment, requiring them to compensate for acceleration appropriately rather than just pointing where they wish to fly and going there.

2.3.4 Actions

The player may perform a number of actions during gameplay, including movement as described, looking around their cockpit independently of their spacecraft's movement vector, firing of weapons and potentially the targeting of enemies using their on-board targeting computer.

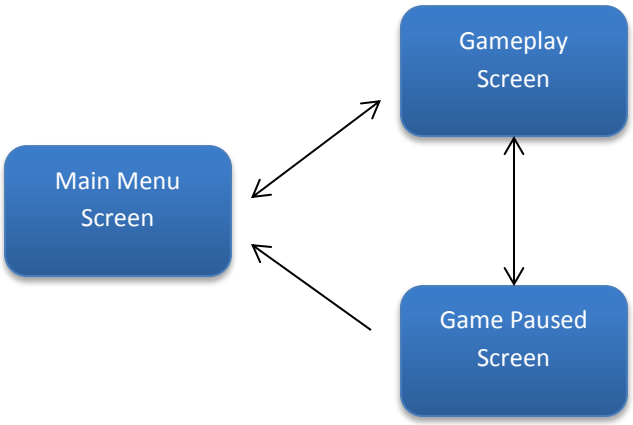
2.3.5 Combat

Combat will take place between the player and enemy targets/spacecraft. The player will control lasers/rail guns/missiles that fire from their ship in a fixed forward direction. When these projectiles hit an enemy object they will inflict damage and eventually destroy them.

If time allows enemy spacecraft will engage in combat with the player utilising basic AI to target, track and fire. In this case if the player gets hit their shields and hull health will be depleted until they are killed, In which case the game will end.

2.4 Screen Flow

2.4.1 Screen Flow Chart



2.4.2 Screen Descriptions

Game screens are divided into three simple states: the main menu, the gameplay and the pause screen.

Upon starting the game application the main menu screen will be presented to the player allowing them to proceed into the game, or exit the application. During gameplay the player may pause the game, allowing them to either exit to the main menu, or resume gameplay when desired.

2.5 Replaying and Saving

Saving of progress will not be considered in the first version of the game as there is no real progress to be saved as it consists of one level.

Replaying of the game will be possible whenever the player desires with the possibility for a command to allow the player to vary enemy types and difficulty if time allows.

3.0 Story, Setting and Character

3.1 Story and Narrative

Whilst there is no strict storyline in the game, a simple backstory is present to allow for game rationale and setting to be developed.

3.1.1 Back story

In approximately the year 2280, humans have learnt to travel to other worlds and star systems and space travel is a common activity for many. Some seek their fortune and fame amongst the stars, acting as mercenaries hired to sabotage and destroy rival targets along the frontiers of human colonisation efforts.

It is in such a frontier system that the game takes place, the player hired for their skills as a combat pilot and tasked with disrupting the operations of a rival faction to that of their employers.

3.1.2 Plot Elements

The player is a mercenary and tasked with destroying the assets and operations of a rival faction.

Enemy targets are other mercenaries, automated drones and industrial facilities/operations that offer competition to the player's employer.

3.2 Game World

3.2.1 General look and feel of world

The environment consists of a space scene with a majestic star field skybox surrounding the player. The star at the centre of the system is large and provides directional light that causes stark shadows between lit and unlit surfaces.

3.2.2 General Description

The game world takes place in a very high orbit of a nearby planet, large enough to fill a large are of space, yet far enough away as to be unreachable during gameplay. This planet will be defined by its type as either 'Earth-like', a gas giant, or a barren world.

In the immediate will be asteroids that rotate slowly and physically present in the game world. If the player collides with and asteroid they will take damage and may be destroyed. If possible asteroids will be randomly generated through a procedural method, creating unique rocky surfaces.

The skybox may contain nebulae to add colour and vibrancy to the scene, and shall consist of subtle variation in the colour, size and intensity of light emitted by each point.

3.3 Characters

The player is a nameless mercenary hired by a large corporation/faction to wage a war against their rivals. No details are developed beyond this in the initial proposal for the game.

4.0 Levels

One level will exist in the initial version of the game consisting of a collection of asteroids, a star field skybox, and a nearby planet.

If time allows various structures like space stations and variations on the main level will be included for variety.

5.0 Interface

The interface is of particular importance when considering this game for the Oculus Rift, and as such traditional interface design principles must be adapted to suit the Oculus.

5.1 Visual System

5.1.1 HUD

A simple HUD like display will be presented to the player as their main source of interaction with their spacecraft. This will display information relating to their velocity, orientation and other details.

Other HUD elements will be integrated to have a physical presence within the game's cockpit on various panels and displays in front of the player.

No strict game-style HUD will be present overlaying any of this, with emphasis on the HUD and displays being part of the ship rather that the screen that the player is viewing it through.

5.1.2 Menus

Menus will provide simple navigable controls allowing the player to exit an existing game, start a new one, or exit the application entirely.

5.1.3 Rendering System

The game will be rendered in full 3D using the *OGRE* rendering engine which will take advantage of current graphics technologies. As such the game should be compatible with both Direct3D and OpenGL graphics APIs and include the latest versions as such if possible.

Attention will be paid to the use of various post-processes and shaders to enhance the graphical fidelity and feel of the game to look like a modern game.

5.1.4 Camera

The camera is of particular importance for this project, and will be directly controllable with the Oculus Rift's head tracking capabilities provided through the Oculus Rift SDK. As such the SDK will be implemented into the game and allow for full 1:1 tracking of the players head, translating it directly into the camera's orientation around the cockpit.

Attention will be paid to rendering the game to two side-by-side images for output to the Rift using a camera offset technique to create stereoscopic 3D. Additionally barrel distortion will be applied to each image as they are rendered in order to be compatible with the Rift's display.

5.1.5 Lighting Models

A simple directional lighting model will provide the main source of light from the main star of the star system the player is located within (as the Sun does). This shall remain static throughout play and attention should be paid to the shadows cast by such a light.

In addition a low-intensity global light will be present to ensure that those areas not lit by the star are not completely unlit and therefore unrecognisable.

5.2 Control System

Controls will be provided in the form of keyboard and mouse controls that allow the player to pitch up and down, roll left and right, and yaw left and right. In addition keys will allow the player to accelerate forwards and decelerate, as well as move to the left, right, up and down independently of the direction of travel.

Additional controls will fire the weapons systems of the spacecraft, allow for targeting of enemies and call up the game menus.

5.3 Audio

Simple open-source audio samples will be located for the various engine, weapon and miscellaneous sounds within the game. If an appropriate sound cannot be found, a placeholder may be used or the sound omitted entirely. Those sounds that cannot be 'heard' outside of the players craft will not be included as is realistic in a vacuum.

5.4 Music

No music will initially be included during gameplay although some may be present within the menus if such a track can be located that fits the genre and feel of the game. Any music chosen will be licensed for free use in the game.

5.5 Help System

Simple tutorial tooltips may be added for the player if time allows.

6.0 Artificial Intelligence

Artificial intelligence will be included in the game if time allows during the project extension phase of development. All AI will be kept simple and follow a Finite-State-Machine style approach where AI exists in various states depending on a number of factors.

6.1 Enemy AI

Basic AI will be implemented for enemy spacecraft if time allows. Basic AI states will include evasion of the player, patrolling of a fixed area, and firing at the player when in range.

7.0 Technical

7.1 Target Hardware

Target hardware is a typical Windows PC based on the x86/64 architecture and the game shall be developed with such hardware in mind. The development will make use of current graphics hardware and RAM availability in order to bring the best quality available from the game engine.

7.2 Development hardware and software

The game will be developed on a Windows PC with an Intel 2600K CPU running at ~4GHz, an NVIDIA GTX 580 GPU and 8GB DDR3 1600MHz RAM and this system will be used as a metric for testing.

Microsoft Visual Studio 2012 will be used for C++ programming along with its integrated VC110 compiler.

7.3 Development procedures and standards

Development will follow an Agile-like approach of software revisions and improvements and comply to both the standards of the British Computing Society and current C++ coding standards.

7.4 Game Engine

The engine used will employ the use of the *OGRE* 3D rendering engine as the foundation of a basic custom game engine that will incorporate existing libraries for sound, physics, AI and miscellaneous areas as necessary to ensure that the focus of programming is on the creation of the game rather than an engine.

7.5 Scripting Language

If possible, the integration of the LUA scripting language for game events may be implemented in order to take a look at a common industry practice of using C++ for engine and core programming, with LUA used for high-level game scripting.

8.0 Game Art

Art shall remain simple and functional and make use of free models, textures and other assets where possible to minimise on asset creation time in order to focus on pure software development for the project.

If asset creation is necessary, simple interpretations of sci-fi style will be produced as quickly as possible using 3DS Max 2013, Photoshop CS5 and other relevant programs. Emphasis will be made on rapid content creation if this is the case and art will not be a focus in this initial development of the game.

8.1 Style Guides

Art shall follow a 'typical' science fiction style seen in many such space combat games with particular focus on broad geometric shapes used for spacecraft and facilities. Planets and celestial bodies may employ the use of basic interpretations of geological features mainly through the use of noise producing algorithms.

Neutral colours shall be present in artificial/spacecraft design with warmer colours for planet surfaces and starlight.

8.2 Environments

Environments will be generated procedurally if possible with a focus on including varied environments that may include colourful nebulae/galaxy swirls/glows in skyboxes for example to break up the monotony of a space skybox.

9.0 Management

9.1 Schedule

The project timeline will run from November 2013 to April 2014 with completed revisions estimated to be competed roughly every 3 weeks with improvements added on each iteration. A feature-complete product is desired by the end of January 2014.

9.2 Test Plan

The primary objectives of the game will be used as testing metrics for each feature introduced into the game. With each development cycle different features will be implemented and thoroughly tested before moving onto the next development cycle. In this manner every feature will be implemented in turn and tested well before the complexity of the game increases in the next phase of development.

Once all primary objectives are met the game will be enhanced as much as possible until the final deadline of April 2014 in which case each enhancement shall be tested as it is implemented.