Interim Report

A Space Simulator Game for the Oculus Rift Virtual Reality Device

Author: Alex Flight (77525)

Supervisor: Marco Gilardi

18/10/2013

TABLE OF CONTENTS

Abstract

This project is undertaken with the intent to produce a fully interactive 'space simulator' game that ~~clearly demonstrates the abilities of the Oculus Rift [1] device as a significant step towards achieving better immersion within a computer game, taking full advantage of the features offered by the Oculus Rift device. The game itself~~ allows for the player to control a spacecraft in a space environment complete with various celestial bodies, semi-realistic controls allowing for 6 degree of freedom and various enemy targets that the player must seek out and destroy in order to progress within the game.

The player will interact with the game using mouse, keyboard and the Oculus Rift Device [1], for which the game will be specifically engineered and designed

Introduction

Do not write in the past tense for the interim, you have not completed the project yet

This project was undertaken with the intent to fulfil the requirement that a game be produced for the Oculus Rift Virtual Reality device, and hence explore the capabilities of the device therein when utilised as a primary interface method for a game. To that end this project explored the Oculus Rift itself as both an input and output interface device, comparing it to traditional methods of interacting with a game as well as exploring game programming and engineering techniques in order to produce a final product that is complete, complex and offers a compelling gameplay experience to users.

Through this report first the type of game to be produced was described and critical gameplay elements identified before exploration of the Oculus Rift device itself as well as the technologies behind it. From here the report then looked at the design and planning of the space game project, taking into account software engineering principles and methods before detailing the implementation process itself in detail. Following on from the completion of the game implementation, this report then highlighted the successes and failures of the project as a whole and critically reflects on the various stages of project development, before reaching a conclusion based on the original goals of the project when compared to what was produced over the project lifetime.

Genre and Typical Gameplay Description

The type of game chosen to demonstrate the creation of a game for the Oculus Rift was a 'space simulator/combat' game which usually entails the use of a player controlled spacecraft as the equipped with the ability to navigate around a space setting involving various celestial objects such as asteroids, nebulae, planets and moons/satellite bodies. Typically level design in such a game is very non-linear and has a large area in which the player may explore freely in contrast to more rigidly designed levels in genres like the First-Person Shooter (FPS).

Within the described setting the player is typically equipped with a weapon system mounted to their spacecraft that allows them to engage various other targets that include but are not limited to: enemy ships, asteroids and debris and space stations and installations. The players' progress in destroying their targets is recorded in some manner until the overall objective of the scenario is met and the player advances to complete their 'mission' or objectives for the game. Excellent examples of such gameplay mechanics are demonstrated in games such as the X-Wing/Tie Fighter series [2] and FreeSpace 1/2 [3] in the 1990's and more recently Strike Suit Zero in 2013 [4], although the genre has stagnated in recent history.

Similar to the 'space combat simulator' genre of games exists a sub-genre that gives the player the ability to freely choose what activities they want to pursue, sometimes labelled as a 'sandbox' or 'free-form' game. In such a game the player controls one or more ships that they may upgrade, purchase and sell and can also engage in trading, exploring and building an 'empire' of property and manufacturing, in addition to typical space combat. Examples of free-form space games include the classic Elite series [5], first produced in 1984 and more recently the X Series [6], famed for its relentless complexity.

In recent months the genre has seen a resurgence of interest [7] after a period of relative inactivity with many projects rebooting old franchises on their way to developments through crowd-funded efforts from their fan-base and renowned pioneers of the genre such as Chris Roberts [8] with 'Star Citizen' [9] and David Braben [10] with 'Elite: Dangerous' [11]

The Oculus Rift Virtual Reality Device

The Oculus Rift itself is best described as a 'virtual reality 3D headset interface device', meaning that the device is worn on the head in order for the user to become full immersed inside a fully 3D virtual environment. At its very essence of functionality it is both an input and output device for a computer system, acting as both a fully 1:1 head orientation tracking device and a stereoscopic 3D output for the images rendered by the computer to be projected to fill a significant portion of the wearers field of view.

With these two elements combined the Oculus Rift fulfils [12] a fully realised implementation of a long sought-after virtual reality headset device that has been the goal of many since the very introduction of computer graphics over 50 years ago [13]. Given its long history various implementations have been seen over the years from training [14] [15] to entertainment [16]. Virtual reality in time developed somewhat of an infamous reputation as something that was never quite ready for mainstream consumption by the average consumer. Says who? However with the introduction of the Oculus Rift expressly designed for video games, a resurgence in interest in the field of virtual reality has remerged in a resounding atmosphere of excitement and anticipation of 'the next big thing' for games technology and interaction [17].

Sensors and Input

Whilst it is beyond the scope of this report to analyse the engineering and functions of the Oculus Rift at a detailed level, a simple description of the various parts of the device is necessary in order to understand the programming and implementation challenges and requirements imposed through the use of the Oculus Rift as an interface device. At the core of head tracking functionality of the Rift are a suite of three different sensors running at an incredibly high frequency of 1000Hz allowing for incredibly accurate translation of orientation in three dimensions into raw digital values. These sensors consist of a combination of solid state gyroscopes, accelerometers and magnetometers that act together to record movement, angular velocities and 'drift' of the headset to produce discreet measurements of rotation and acceleration in the cardinal x, y and z axes of motion; commonly referred to as pitch, roll and yaw [18]. For more information in the technical aspects and challenges faced by the Oculus Rift, many articles are available on the Oculus VR company website [19] [20].

Display and Output

In recent years the popularity of and other portable electronics devices has led to a massive leap forward in the technologies behind small displays in terms of overall quality and, most importantly, value for money. These displays offered an ideal candidate for the Oculus Rift as a head-mounted device, with the need for a compact screen in order to make the device light and wearable without discomfort. It is through the use of lenses in front of such a display that the Oculus Rift offers the wearer a truly wide field of view that encompasses the majority of their vision.

With this physical system in place the challenge for integrating the Oculus Rift into games is to produce an output that projects two discrete images (facilitating a stereoscopic 3D image) that are offset by a small amount and then warped using a barrel distortion projection in order to compensate for the 'pincushion' applied by the lenses in front of the display. When implemented successfully the image viewed through the Oculus is one of almost total immersion encompassing a large portion of an individual's field of view and hence leading to a feeling from the wearer that they are really 'inside' the game in question [21] [22].

Project Objectives

In the next page

Do not restate what the interim specification says

~~The objectives of this project included a selection of 'primary objectives' which were considered to be the base features that the final implementation of the game must include in some way in order to be feature complete and meeting the requirements of the project as part of a BSc degree course as well as offer a sufficient challenge in terms of programming skills required and time taken to implement them to a certain extent. In addition to these there were also several 'extensions' that were specified as extra features that would be produced if time allowed after the primary objectives were met. These included many more ambitious goals that would enhance the project as a whole but were not considered as vital components of the game itself.~~

Which language are you going to use? Which Game Engine? Which libraries? Which IDE?

Primary Objectives

- PO. 1 Implement a simple space shooter/simulator game using an appropriate game engine (Specify) to allow for the player to control a spacecraft with independent head tracking provided by the Oculus Rift.
- PO. 2 Include 'enemy' targets that the player must locate & navigate to then shoot and destroy in order to complete the game/level.
- PO. 3 Provide a method of maintaining a 'score' or objectives for the player to complete through destruction of targets. Upon destruction of targets the player will be closer to completing their objectives.
- PO. 4 Produce a space environment consisting of a star field skybox and other celestial bodies such as a nearby planet, asteroids, nebulae etc. in which gameplay will take place. Procedural generation of these assets will be considered where possible and appropriate.
- Produce a working cockpit-based camera that takes direct input from the Oculus Rift motion tracking sensors to allow for orientation of the head to look around in every direction. A simple 3D model of the cockpit interior is needed for the basic functionality to make sense and immerse the player.
- Integrate a solution that allows for output to an Oculus Rift display either through custom integration of the Oculus Rift API or an existing method within the chosen engine.
- Implement/include a control scheme for the spacecraft based on a simple physics model of simplified flight-controls allowing for pitch, roll and yaw. In addition allow for translation along x, y and z axes which will allow for 6 degrees of freedom.
- Produce a simple user-interface that is designed to be read from the various consoles and features of the cockpit itself much like a modern military fighter plane. This could include display monitors and a HUD like interface projected in front of the player. Traditional UI elements tend to be outside the OR's Field of View at the edge of the screen after barrel distortion has been applied.
- Implement a weapons system that allows the player to shoot at various targets within the environment in order to destroy them to meet a specified objective (e.g. destroy x numbers of targets in y time)

Extensions

- E1 Implementation of basic enemy AI that allows for the player to dogfight with enemy ships in a simple manner. This could be expanded to increase difficulty and the sophistication of the AI as time allows.
- E2 Design of a fully immersive UI that integrates well with the Oculus Rift as a believable interface directly tied to the spacecraft itself.
- E3 Implementation of advanced graphics and lighting techniques to take advantage of modern graphics hardware where appropriate.
- E4 Expansion of control and physics to incorporate a fun representation of Newtonian flight through space (e.g. turning to face the enemy while conservation of momentum/inertia allows movement along current movement vector)
- Investigate use of procedural generation of meshes and textures to generate realistic asteroids and planetary bodies.
- Exploration of shaders and current graphics trends to enhance visual experience as well as explore what works well with the Oculus Rift.

- Expansion of core gameplay to include scripted scenarios and missions as time allows.
- Include varied environments in which to fly in.
- Enhance the experience of the player in the cockpit with various effects such as g-forces pulling the head around, vibrations, creating a sense of acceleration and speed.
- A missile system that allows the player to lock a target by looking at them from any direction within the cockpit making full use of the Oculus Rift as a targeting device while the player continues to track a target regardless of the spacecraft's facing.
- A menu and interface implementation.
- Adjustable difficulties.
- Looking at tessellation to enhance detailed asteroid and terrain meshes produced on the GPU.
- Gameplay & visual enhancements where appropriate to allow for a more immersive product.
- Inclusion of more advanced shaders etc.

PROFESSIONAL CONSIDERATIONS

Throughout this project various considerations were made in order to ensure that the project met ethical standards and guidelines as defined by the Code of Conduct and Code of Practice of the British Computing Society [23] [24]. Such considerations included the decision to ensure the project itself was technically achievable and challenging in terms of a BSc final year project and dissertation through the inclusion of various technical and engineering challenges that had to be overcome in order to develop a good game implementation. These can be seen during the software engineering approach to the analysis, design, implementation and testing of the project as well as the use of new and current techniques and technologies in order to develop the systems in use by the game itself.

The ethical considerations of the development of the game in terms of content were also applicable to the project as a whole. These considerations ensured that the theme, story, background and core gameplay of the game was not in any way culturally insensitive or intended to cause offense or harm to an individual, group or society. In addition the use of the Oculus Rift device was also considered during the development cycle and therefore attention was paid to ensuring the user would not experience any adverse reactions or effects during gameplay due to the effects of the hardware and in particular the images projected to it. For example it would have been possible to cause the images projected to the Oculus Rift to be misaligned or offset to too great a degree and intentionally cause discomfort and nausea to the user and therefore steps were taken in order to minimise or remove this effect completely during development [25].

A final professional consideration of the project was that a warning should be included for users that like with any interface device with a computer, ergonomics should be observed and regular rest periods should be taken, especially considering the nature of the current form of the Oculus Rift as a development kit and hence not including fully realised health studies and implications from the prolonged usage of the device.

Requirements Analysis

Before a large software project is undertaken a detailed analysis of the various requirements of the system must first be undertaken, with this project being no exception. Such requirements identified during this project could be classified into several categories: customer requirements, those requirements which a user/player of the game would need in order for the game to be complete; functional requirements, those features and aspects of the game which must be solved or completed in order to have a functioning end product that both makes sense and offers viable gameplay; and non-functional requirements which include all the miscellaneous factors external to the system as a whole, but have an influence on how well the game would perform.

Customer Requirements

In the case of games development the customer may be more helpfully labelled as a 'player' or 'user', yet they are still a customer for all intents and purposes. In terms of this project the customer was considered as a typical player of the game that would be interested in using the system purely for entertainment and immersion in 'another world'.

With this typical user in mind several goals for the game to meet when completed were considered for the system to be a successful implementation of a game as an entertainment product:

- *The game should be enjoyable and fun* – The absolute goal of the project should be to produce a game that the player will enjoy playing. To this effect the gameplay should include interactive elements that allow for the player to pursue a goal with any reasonable method that they choose as provided to them by the controls and entity that the player controls. This is perhaps the hardest of the requirements to meet in the design of a game as the concept of fun is different for everyone.
- *A challenge should be presented to the player* – The game should be challenging in some aspects of the gameplay and should ensure that the player has a goal which is not trivial to reach. Therefore the game must force the player to act and react critically to the scenario in which they are placed in order to meet their objectives.
- *The game should be easy to understand* – Controls, general gameplay and objectives should be straightforward and easy for a player to understand so as to not get in the way of their enjoyment or provide a barrier to cause unnecessary frustration from poor design.
- *Prompts and instructions should be provided to the player* – In order to instruct the player on the use of the game appropriate instructions should be accessible to the player in some manner so as to allow them to learn how to start, play and end the game.
- *The player's health and wellbeing should be considered* – Particularly true in the case of the use of the head-mounted Oculus Rift, the game should not cause unnecessary strain on the eyes of the player through screen effects or artefacts and motion sickness should be considered as a real factor to compensate in the design of the camera in the game in regards to field of view, stereo offset distances etc.

Mention these things without
giving them so much space, they
are important but not the point
of the project

Functional Requirements

In this section map each item to the objectives of your project, to this end is better to have a numbered list on the objectives section

Characterising the game as a final product should be the functional requirements that the system as a whole should achieve in order to be a successful implementation. When undergoing testing the functional requirements are considered the goals which the system must reach in order to be considered successful.

Do not state the obvious

- *~~The game must run~~* – Upon initiation of the game application the game should load all executable code and run successfully on x86 architecture Windows PC.
- *A game world must be rendered in 3D representing a space scene in which the player experiences the game* – An implementation of a playable 3D space scene should be created within the game including necessary entities befitting to such an environment. This environment should then be the level upon which the gameplay itself is built. Such assets included in the game world must be created either through the importing of pre-built meshes and textures or utilising the CPU and GPU to generate the various game entities procedurally where possible.
- *The player should inherit direct control over a spacecraft vehicle within the game world* – The player should be represented within the game world as a spacecraft entity that is directly controllable through the interface of their choosing. At a minimum basic keyboard controls should be provided to the player allowing them to maneuverer their craft and fire at enemy targets in order to destroy them.
- *The player camera should be set up to accommodate the Oculus Rift VR interface device* – The main player camera in the world should be located within the cockpit of the player spacecraft and should offer the player the sense that they are surrounded by the environment using the unique features offered by the Rift. Display output to the Oculus Rift should accommodate the barrel distortion of two discrete images to the left and right eyes of the Oculus Rift using a 'side-by-side' stereoscopic rendering technique with appropriate offset for two eyes to create an effective 3D image.
- *The player should be able to orient their camera through use of the head tracking capabilities of the Oculus Rift* – Orientation of the player camera within the cockpit should allow for the player to pitch, yaw and roll their whole head in order to look around their surroundings with as close to a 1:1 input response as possible.
- *Basic 'space physics' should be implemented in the game* – A simplified interpretation of Newtonian movement should be implemented into the game engine to allow for relatively realistic movement according to laws of motion and conservation of momentum where thrust, mass and acceleration are all taken into account in order to modify the relative velocities of the objects within the game. As such the player should be allowed 6 degrees of freedom in the movement of their spacecraft, allowing them to pitch, roll, yaw, accelerate, decelerate and translate their craft in the along the cardinal x, y and z axes. This should be expanded where time and features allow.
- Be more specific
  A genre-appropriate weapon system should be controllable by the player in the game – A weapon system suited to a sci-fi spacecraft should be represented in some way by the game and allow for damage and destruction of eligible player targets in the game.
- *Appropriate enemy entities should exist within the game world for the player to destroy* – Enemy entities such as other spacecraft or target drones should exist in the game world and be able to be destroyed through the interaction with the player's weapons systems.
- *A suitable method of recording the score of the player should be implemented* – Upon destruction of enemy targets the player's 'score' should increase in some way either through allocation of points or the progression towards the completion of some defined goal of destroying n specific enemies before the player's mission is complete.
- *Various visual and audible effects should be considered to enhance gameplay* – The inclusion of various graphical and audio effects and elements should be included in order to produce an effective and believable space combat simulator environment. Such effects could include post-processing to enhance the graphical fidelity of the scene, ship, weapons and engine sounds etc.

Non-Functional Requirements

Alongside the functional requirements that are necessary for the game to make sense and be playable, non-functional requirements must also be considered in order for the experience of the game to be successful and enjoyable.

- *The game should run at a good frame rate* – The final game should run at a frame rate that allows for effective reproduction of motion and full interactivity from the player in order for them to have an enjoyable experience.
- *The game should have fast load times* – The loading of the game should be fast in order to prevent player frustration from long waiting times in order to play.
- *The game should comply with modest system requirements in order to facilitate play across a moderate range of computer specifications* – The game should be optimised to the best extent in order to run on a computer without top-end specifications that require expensive components.
- *Text and user interface elements should be readable and easily interpreted* – The graphical user interface within the game should comply with basic design paradigms in order to be understandable to the player.
- *The system requires an Oculus Rift device in order to be playable* – The player should own and connect an Oculus Rift VR device to their computer in order to play the game.
- *The player should be using an appropriately supported operating system and architecture* – The player is required to run the game on a Windows PC running the x86/x86-64 instruction set architecture in order to run the game executable.

Again mention these things but do not give them too much space, as you are stating the obvious

UML Diagrams

With the system requirements clearly defined the use of UML diagrams allows for a diagrammatic representation of the system requirements to be presented effectively and succinctly. These diagrams include Use Case models, Class Diagrams, Sequence Diagrams, Activity Diagrams and State Diagrams.

Use Case Models

Use case models offer an overview of the system with regards to the users and how they interact with it. In the case of this project a typical user is a 'Player' and they are interacting with the game in order to play it for personal enjoyment.

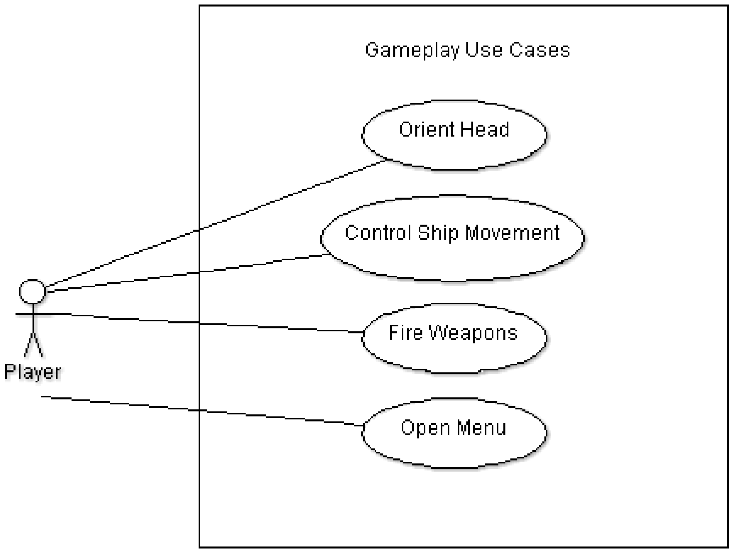


FIGURE 1 - GENERAL GAMEPLAY USE CASE

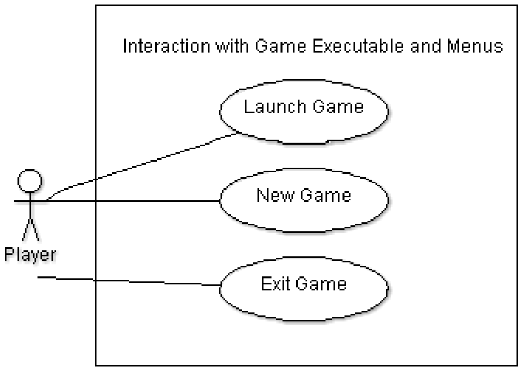


FIGURE 2 - MENUS USE CASE

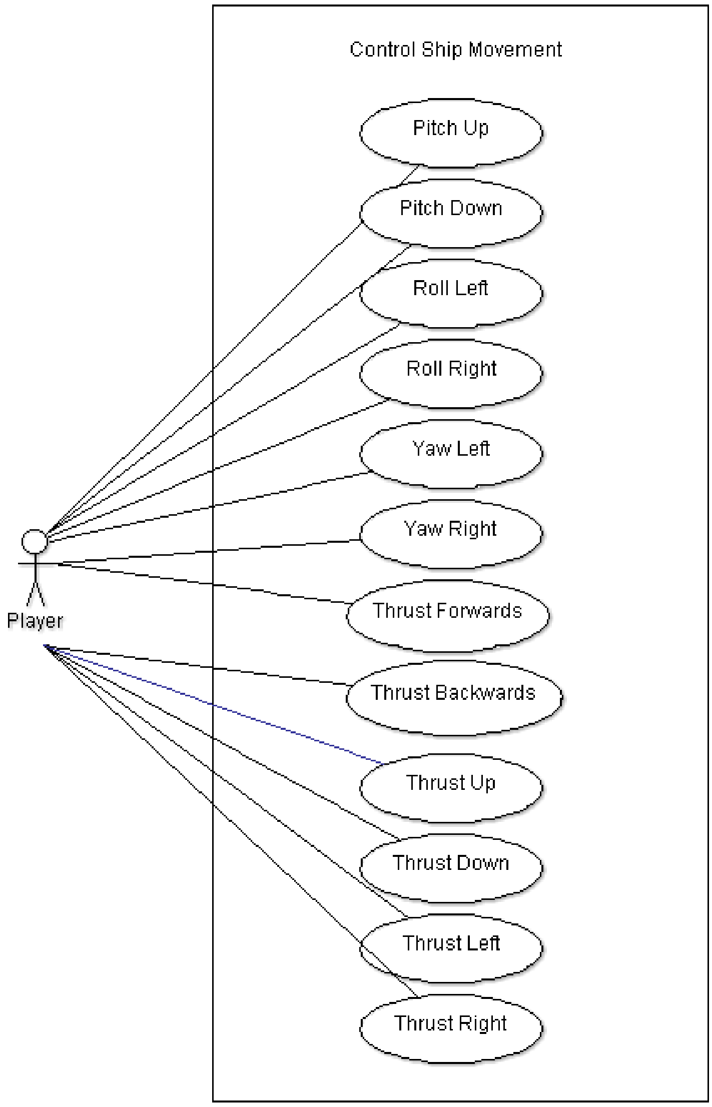


FIGURE 3 - PLAYER CONTROL USE CASE

Explain the diagrams, do not leave to the reader free interpretation of your intentions, is bad practice. Diagrams serves as support for the explanation they are not the explanation.

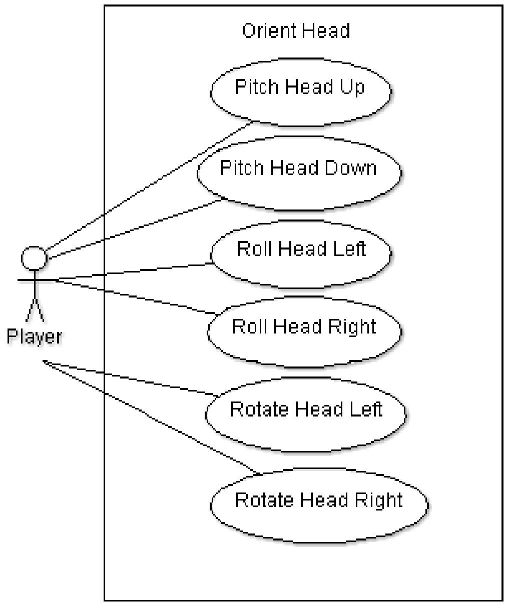


FIGURE 4 - ORIENT HEAD/CAMERA USE CASE USING THE OCULUS RIFT

State Diagrams

Accompanying the use case diagrams is an overall state diagram which represents the various states the game is within when it is running from initialisation through to termination of the program.

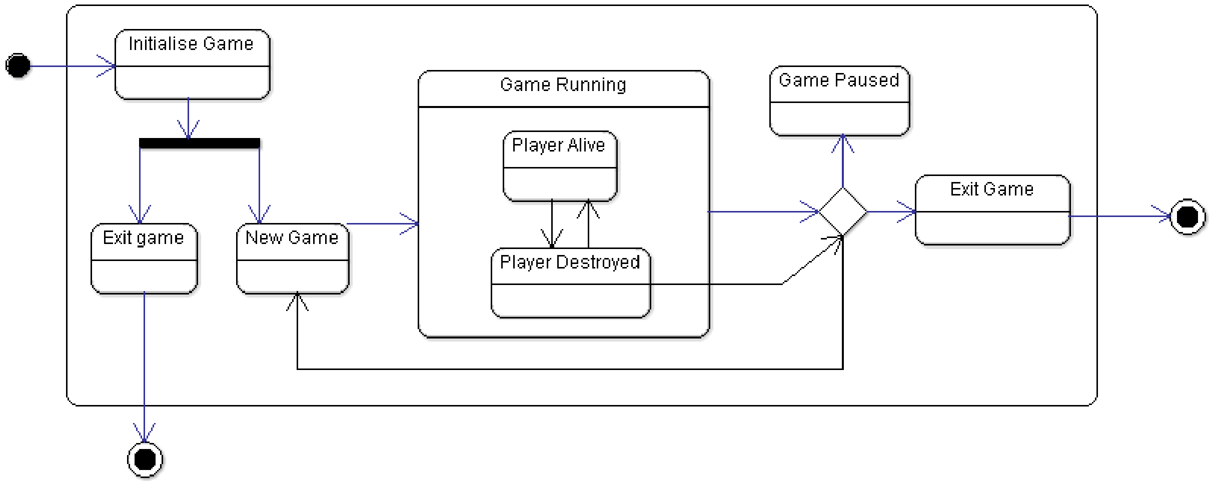


FIGURE 5 - STATE DIAGRAM OF THE SPACE GAME

CLASS DIAGRAM

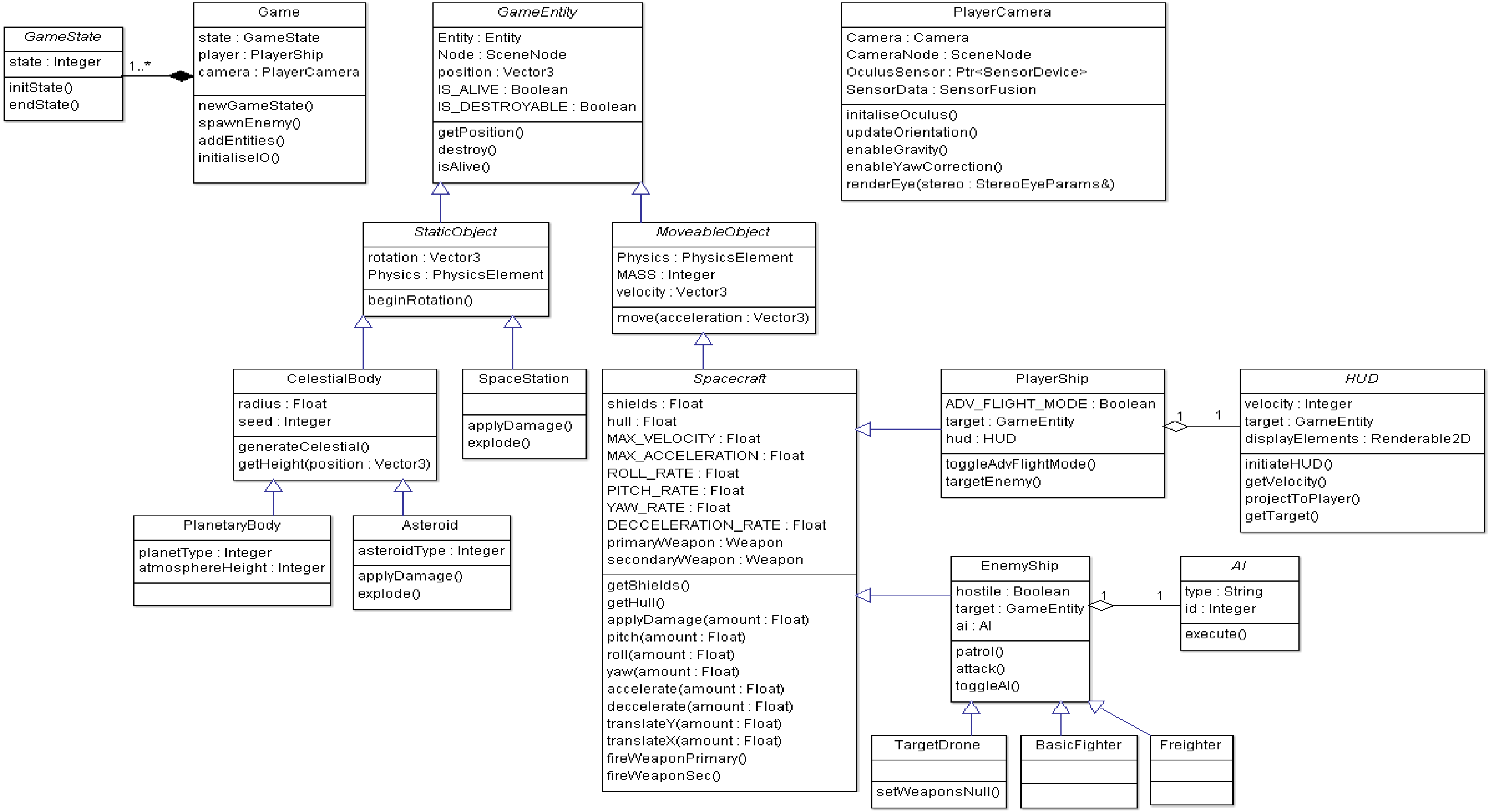


FIGURE 6 - OVERALL CLASS DIAGRAM IDENTIFYING VARIOUS COMPONENTS OF THE GA

SEQUENCE DIAGRAM

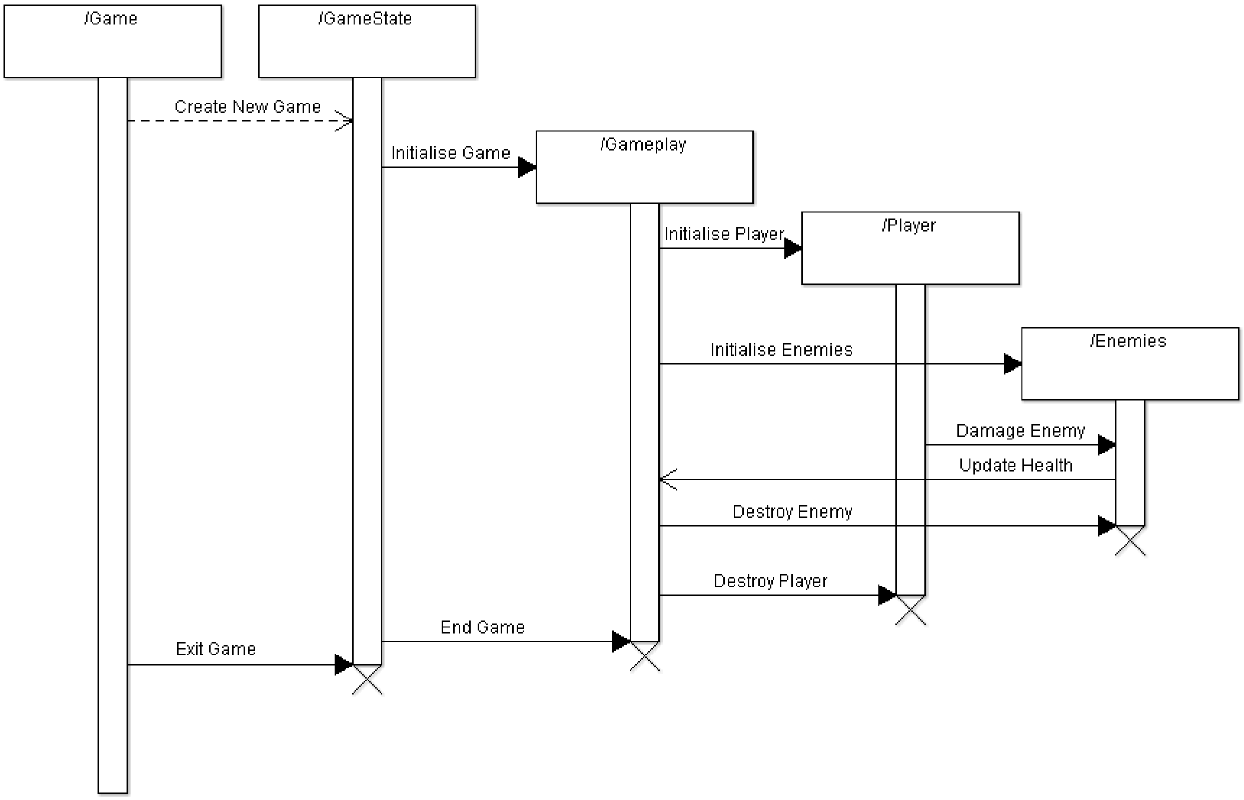


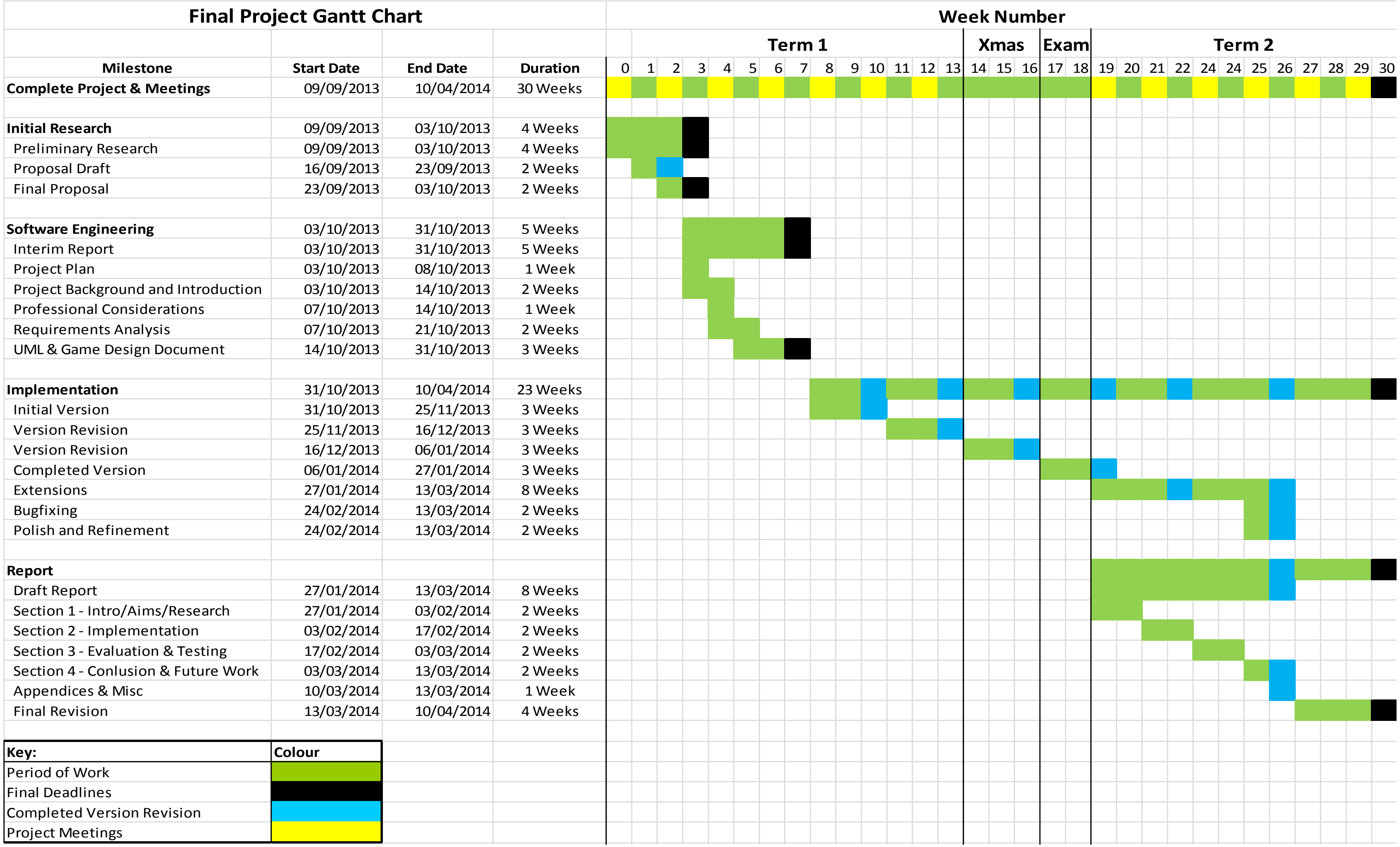
FIGURE 7 - SEQUENCE DIAGRAM OF GAME EVENTS

PROJECT PLAN

The overall project plan is best represented in a Gantt chart as included in the proposal document; it is reproduced here for convenience.

To this date the project is on schedule according to the Gantt chart and all tasks as indicated have been completed up to week 6. Not included on the chart is the research and prototyping that has been completed up until this point, this includes:

- Research into background material for the Oculus Rift itself; this includes technical specifications and background of the technologies and engineering involved in the device, implementations of the Oculus Rift in various demos and engines and experimentation with integration of the Oculus Rift to the Ogre, Unity, Unreal and CryEngine game/rendering engines.
- Research of and experimentation with procedural algorithms for planet generation both using CPU and GPU techniques.
- Research and prototyping of gameplay mechanics in various engines including Unity, Ogre and CryEngine.
- Background reading of appropriate material when and where it was possible.



Final Project Gantt Chart

Milestone	Start Date	End Date	Duration
Complete Project & Meetings	09/09/2013	10/04/2014	30 Weeks
Initial Research	09/09/2013	03/10/2013	4 Weeks
Preliminary Research	09/09/2013	03/10/2013	4 Weeks
Proposal Draft	16/09/2013	23/09/2013	2 Weeks
Final Proposal	23/09/2013	03/10/2013	2 Weeks
Software Engineering	03/10/2013	31/10/2013	5 Weeks
Interim Report	03/10/2013	31/10/2013	5 Weeks
Project Plan	03/10/2013	08/10/2013	1 Week
Project Background and Introduction	03/10/2013	14/10/2013	2 Weeks
Professional Considerations	07/10/2013	14/10/2013	1 Week
Requirements Analysis	07/10/2013	21/10/2013	2 Weeks
UML & Game Design Document	14/10/2013	31/10/2013	3 Weeks
Implementation	31/10/2013	10/04/2014	23 Weeks
Initial Version	31/10/2013	25/11/2013	3 Weeks
Version Revision	25/11/2013	16/12/2013	3 Weeks
Version Revision	16/12/2013	06/01/2014	3 Weeks
Completed Version	06/01/2014	27/01/2014	3 Weeks
Extensions	27/01/2014	13/03/2014	8 Weeks
Bugfixing	24/02/2014	13/03/2014	2 Weeks
Polish and Refinement	24/02/2014	13/03/2014	2 Weeks
Report			
Draft Report	27/01/2014	13/03/2014	8 Weeks
Section 1 - Intro/Aims/Research	27/01/2014	03/02/2014	2 Weeks
Section 2 - Implementation	03/02/2014	17/02/2014	2 Weeks
Section 3 - Evaluation & Testing	17/02/2014	03/03/2014	2 Weeks
Section 4 - Conlusion & Future Work	03/03/2014	13/03/2014	2 Weeks
Appendices & Misc	10/03/2014	13/03/2014	1 Week
Final Revision	13/03/2014	10/04/2014	4 Weeks

Key:	Colour
Period of Work	🟩
Final Deadlines	⬛
Completed Version Revision	🟦
Project Meetings	🟨

FIGURE 8 - GANTT CHART OF PROJECT TIMELINE

Log of Activities

Interim Log

Week 0 – 09/09/2013 – 15/09/2013

- Had initial meeting with project supervisor, discussed project goals and outline as a whole. (30 minutes)
- Began investigating Oculus Rift itself and existing games that utilise the device already. (10 hours)

Week 1 – 16/09/2013 – 22/09/2013

- Began initial draft of project proposal. (2 hours)
- Investigated engines for use with Oculus Rift. (8 hours)
- Explored forums, articles and blogs online for impressions in implementing the OR with various games and engines. (6-8 hours)
- Gathered information on matching genres and gameplay types that worked well with the Oculus Rift based on reviews and opinions. (4 hours)
- Experimented with Oculus Rift in CryENGINE 3 SDK, UDK, Torque 3D and Irrlicht Engine. (2 hours)
- Began development of 'space shooter/simulator' idea for project proposal taking into account impressions and experiences with the OR. (2 hours)
- Continued research (approx. 15 hours)

Week 2 – 23/09/2013 – 29/09/2013

- Had second meeting with project supervisor, looked at current proposal documents and discussed possible engine usage, as well as refinements to the proposal document itself. (25 minutes)
- Investigated Ogre 3D engine usage with the Oculus Rift. (2 hours)
- Developed a small prototype in Unity, could not get Oculus Rift to work without Pro version and the integration it provides. (2 hours)
- Wrote official project proposal document. (2 hours)
- Developed project Gantt chart, considered layout not well formed for Agile-style development of game product opposed to waterfall project structure, but broad statement of time allocation conveyed in an effective manner. (1 hour)
- Continued research into extensive amount of forums and websites detailing Oculus Rift use, implementation, and similar games in various engines etc. (approx. 20 hours)
- Read chapters 1-10 of 'Game Engine Architecture' by Jason Gregory. (approx. 12 hours) [26]

Week 3 – 30/09/2013 – 06/10/2013

- Began researching past examples of similar projects and games. Analysed the upcoming releases and noted the recent resurgence of popularity in the space combat simulator genre with examples such as Star Citizen and Elite: Dangerous (4 hours)
- Located various resources on the Ogre forums devoted to similar projects including space environments and space combat simulator gameplay. (4 hours)
- Read more of 'Game Engine Architecture' by Jason Gregory. (10 hours)
- Began refreshing knowledge of C++ (4 hours)
- Researched procedural methods of terrain and planet generation in great detail, implemented a shader-based procedural generator in Unity using a cube-mapped sphere. (20+ hours)

- Explored existing examples of procedural planet generation in Ogre, paid particular attention to previously mentioned cube-mapped spheres as the most suitable method for generating procedural spherical bodies. (10+ hours)

Week 4 – 07/10/2013 – 13/10/2013

- Continued to look at extensive procedural planet generation resources. (15 hours)
- Compiled Interim Report template document ready for addition of sections. (1 hour)
- Collated various bibliographical sources using Bib.me web service to aid in management of bibliography which was growing significantly large. (1 hour)
- Remainder of time spent on two other assignments as deadline approached. Limited time available.

Week 5 – 14/10/2013 – 20/10/2013

- Initial period of week spent on other projects for deadlines limited time available.
- Continued research into procedural generation. (10 hours)
- Produced introductory section for Interim Report. (4 hours)
- Professional considerations section. (1 hour)
- Requirements analysis section. (5 hours)
- Remaining sections of Interim Report. (5 hours)
- Continued background reading and miscellaneous research. (10 hours)

BIBLIOGRAPHY

[1] Oculus Rift - Virtual Reality Headset for 3D Gaming | Oculus VR [Online] oculusvr.com. Oculus VR. 2013. Available from: http://www.oculusvr.com/

[2] Wookiepedia [Online] Star Wars: TIE Fighter. Wookiepedia. 2013. Available from: http://starwars.wikia.com/wiki/Star_Wars:_TIE_Fighter

[3] Reinstall: FreeSpace 2 [Online] PC Gamer. Mahood, A. 2012. Available from: http://www.pcgamer.com/2012/12/22/reinstall-freespace-2/

[4] Strike Suit Zero [Online] Strike Suit Zero. Born Ready Games. 2013. Available from: http://strikesuitzero.com/

[5] The Making Of: Elite [Online] Edge Online. Edge Staff. 2012. Available from: http://www.edge-online.com/features/making-elite/

[6] EGOSOFT [Online] X3: Albion Prelude Info. EGOSOFT. 2013. Available from: http://www.egosoft.com/games/x3ap/info_en.php

[7] 14 Space games you should be excited about right now [Online] PC Gamer. Schilling, C. 2013. Available from: http://www.pcgamer.com/gallery/14-space-games-you-should-be-excited-about-right-now/

[8] MobyGames - Chris Roberts [Online] MobyGames. MobyGames. 2013. Available from: http://www.mobygames.com/developer/sheet/view/developerId,6555/

[9] Star Citizen by Cloud Imperium Games Corporation - Kickstarter [Online] Kickstarter. Cloud Imperium Games Corporation. 2012. Available from: http://www.kickstarter.com/projects/cig/star-citizen

[10] Bobbie Johnson speaks to David Braben, creator of Elite | Technology | theguardian.com [Online] theguardian.com. Johnson, B. 2008. Available from: http://www.theguardian.com/technology/2008/jul/08/games.interviews

[11] Elite: Dangerous by Frontier Developments - Kickstarter [Online] Kickstarter. Frontier Developments. 2012. Available from: http://www.kickstarter.com/projects/1461411552/elite-dangerous

[12] Heaven, D.: New Scientist - Virtual reality rises again., 20 (2013)

See note below*

[13] A Critical History of Computer Graphics and Animation [Online] Wayne Carlson Personal Website - Ohio State University. Carlson, W. 2006. Available from: http://design.osu.edu/carlson/history/lesson17.html

[14] Seymour, N., Gallagher, A., Roman, S., O'Brien, M., Bansal, V., Andersen, D., Satava, R.: Annals of Surgery - Virtual Reality Training Improves Operating Room Performance., 458–464 (2002)

[15] NATO Defense Research Group Virtual Reality, Training's Future?: Perspectives on Virtual Reality and Related Emerging Technologies 1st edn, Plenum Press, New York (1997)

[16] Zagal, M.: Challenges for success in stereo gaming: a Virtual Boy case study. In : ACE '09, New York, pp.99-106 (2009)

[17] Telegraph, T. Oculus Rift 3D virtual reality headset steals show at Eurogamer expo. Available from: http://www.telegraph.co.uk/technology/video-games/10340432/Oculus-Rift-3D-virtual-reality-headset-steals-show-at-Eurogamer-expo.html (Accessed October 2013)

[18] Sensor Fusion: Keeping It Simple [Online] Oculus VR. LaValle:, S. 2013. Available from: http://www.oculusvr.com/blog/sensor-fusion-keeping-it-simple/

[19] The Latent Power of Prediction [Online] Oculus VR. LaValle, S. 2013. Available from: http://www.oculusvr.com/blog/the-latent-power-of-prediction/

[20] Building a Sensor for Low Latency VR [Online] Oculus VR. Oculus VR. 2013. Available from: http://www.oculusvr.com/blog/building-a-sensor-for-low-latency-vr/

[21] What Does It Look Like In the Oculus Rift? [Online] Road To VR. Lang, B. 2013. Available from: http://www.roadtovr.com/what-does-it-look-like-in-the-oculus-rift/

[22] Digital Foundary: Hands-on with Oculus Rift [Online] Eurogamer.net. Leadbetter, R. 2013. Available from: http://www.eurogamer.net/articles/digitalfoundry-hands-on-with-oculus-rift

[23] BCS Code of Practice [Online] British Computer Society. British Computer Society. 2004. Available from: http://bcs.org/upload/pdf/cop.pdf

[24] Code of Conduct for BCS Members [Online] British Computer Society. British Computer Society. 2011. Available from: http://www.bcs.org/upload/pdf/conduct.pdf

[25] Designing around motion sickness [Online] Oculus Developer Forums. Oculus Developer Forums. 2013. Available from: https://developer.oculusvr.com/forums/viewtopic.php?f=32&t=15

[26] Gregory, J. Game Engine Architecture 1st edn. A K Peters, Ltd., Natick, Ma, USA (2009)

[27] Inside Oculus: a tour of virtual reality's would-be savior [Online] The Verge. Hollister, S. 2013. Available from: http://www.theverge.com/2013/3/19/4122388/can-oculus-rift-save-virtual-reality

*
Referencing follows specific rules for books and journals,
see the library website on this matter

APPENDIX

PROPOSAL DOCUMENT