

## **Working Title**

A Space Simulator Game for the Oculus Rift Virtual Reality Device

## **Project Description**

This project is intended to produce a fully interactive 'space simulator' game that clearly demonstrates the abilities of the Oculus Rift 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.

## **Aims and Objectives**

### ***Primary Objectives***

- Implement a simple space shooter/simulator game using an appropriate game engine to allow for the player to control a spacecraft with independent head tracking provided by the Oculus Rift.
- Include 'enemy' targets that the player must locate & navigate to then shoot and destroy in order to complete the game/level.
- 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.
- 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 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***

- 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.
- Design of a fully immersive UI that integrates well with the Oculus Rift as a believable interface directly tied to the spacecraft itself.
- Implementation of advanced graphics and lighting techniques to take advantage of modern graphics hardware where appropriate.
- 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.

**Project Relevance**

The proposed project is considered to be relevant to the degree course as it explores a game concept that requires programming knowledge to implement game logic, entities, 3D graphics, I/O and GUI design, as well as a focus on the Oculus Rift itself as both an input and output device. By the end of the project a large amount of programming should have been completed to allow for the core functionality of the game itself alongside a full set of associated documentation, code commenting, version control history and various other documents entailing the full software engineering process.

**Resources Required**

Aside from the Oculus Rift device which has already been obtained for the duration of the project, no other resources are anticipated to be required. The possibility of a Unity-Pro license may be considered but a 4-month trial is available if that engine is chosen for development with.

## Engine Considerations

- Unity Engine 4 – very appropriate for Oculus Rift projects with new built-in support for the Oculus SDK, however the ‘Pro’ version is required for use of the Oculus Rift in the development of a project [1]. A 4-month trial is available with the Oculus Rift on request as well as a cheap educational license which can be purchased by me for approximately £80. [2]
- Irrlicht 3D Engine – open source MIT licensed rendering engine without current Oculus Rift support, though could be integrated as part of the project. Provides the simplest form upon which to build a game from scratch [3]
- CryEngine 3 FreeSDK – Oculus Rift support added with the latest revision and full source code access to the engine components offer a surprisingly fully featured engine to begin development with. However updated source code is not yet available for current revision with no firm ETA on the next release to include source code. Possibility for use if updated source code is re-released with Oculus Integration before project work begins in earnest. [4]
- Ogre 3D Engine – Open source with plenty of documentation and an active community that has already integrated Oculus Rift support. However Ogre 3D is itself just a 3D rendering engine and so would require integration of other external libraries and components to produce a fully realised game engine. [5]
- Torque3D Engine – Another full engine with full MIT open source code though reliance of proprietary scripting language could prove prohibitive [6].
- Unreal Development Kit (Unreal Engine 3) – Proven engine with a long history of developed games, however full source is not available and relies on UnrealScript scripting language to produce custom games. Whilst an excellent option in essence, limitations in ‘map’ size may prove too great.
- Custom written engine – would comprise a large task that may encompass too much time devoted to engine programming over the development of the game itself. If this project weren’t focused on producing an Oculus Rift game this would be the best exploration of programming skills but a focus on the Oculus Rift itself would be preferable. [7]

## Language Considerations

- C++ presents the primary language for game development worldwide due to its inherent status as a ‘lower level’ language than C#, Python, Java etc. Use of C++ would be preferred in order to gain experience with the language and allow for demonstration of use of the language in future employment/projects. Current familiarity with C++ is not as high as C# however. Necessary if using Irrlicht, Ogre3D, CryEngine 3 or another/custom implementation.
- C# offers a higher-level language when compared to C++ and presents a more comfortable style when working with Unity in particular. By its nature efficiency can suffer but not majorly when compared to C++ unless working on heavy optimisations of engine design etc. Better familiarity with C# already exists. [5]

## Timetable

Study Timetable 13/14, Week starting << 23-Sep-2013 >>					
Term: Autumn Teaching 13/14 Week: 2 Year:			Today iCal Help		
Timetable data maintained by your School Office			Total time: 9 hours		
	Monday 23 Sep	Tuesday 24 Sep	Wednesday 25 Sep	Thursday 26 Sep	Friday 27 Sep
08:00					
08:30					
09:00					
09:30					
10:00					
10:30					
11:00		3D Animation (Laboratory 1) Chichester 1 CI021-LAB1B			Video Production Techniques (Laboratory 1) Richmond 3B11
11:30					
12:00					
12:30					
13:00					
13:30					
14:00				3D Animation (Lecture 1) Jubilee Building JUB-144	Video Production Techniques (Lecture 1) Richmond 3B11
14:30					
15:00					
15:30					
16:00				Human-Computer Interaction (Seminar 3) Pevensey 1 1B8	
16:30					
17:00	Human-Computer Interaction (Lecture 1) Pevensey 1 1A7				
17:30					
18:00					
18:30					
19:00					
19:30					
20:00					
20:30					

Figure 1 - Personal timetable for Autumn Teaching Term

## Gantt Chart

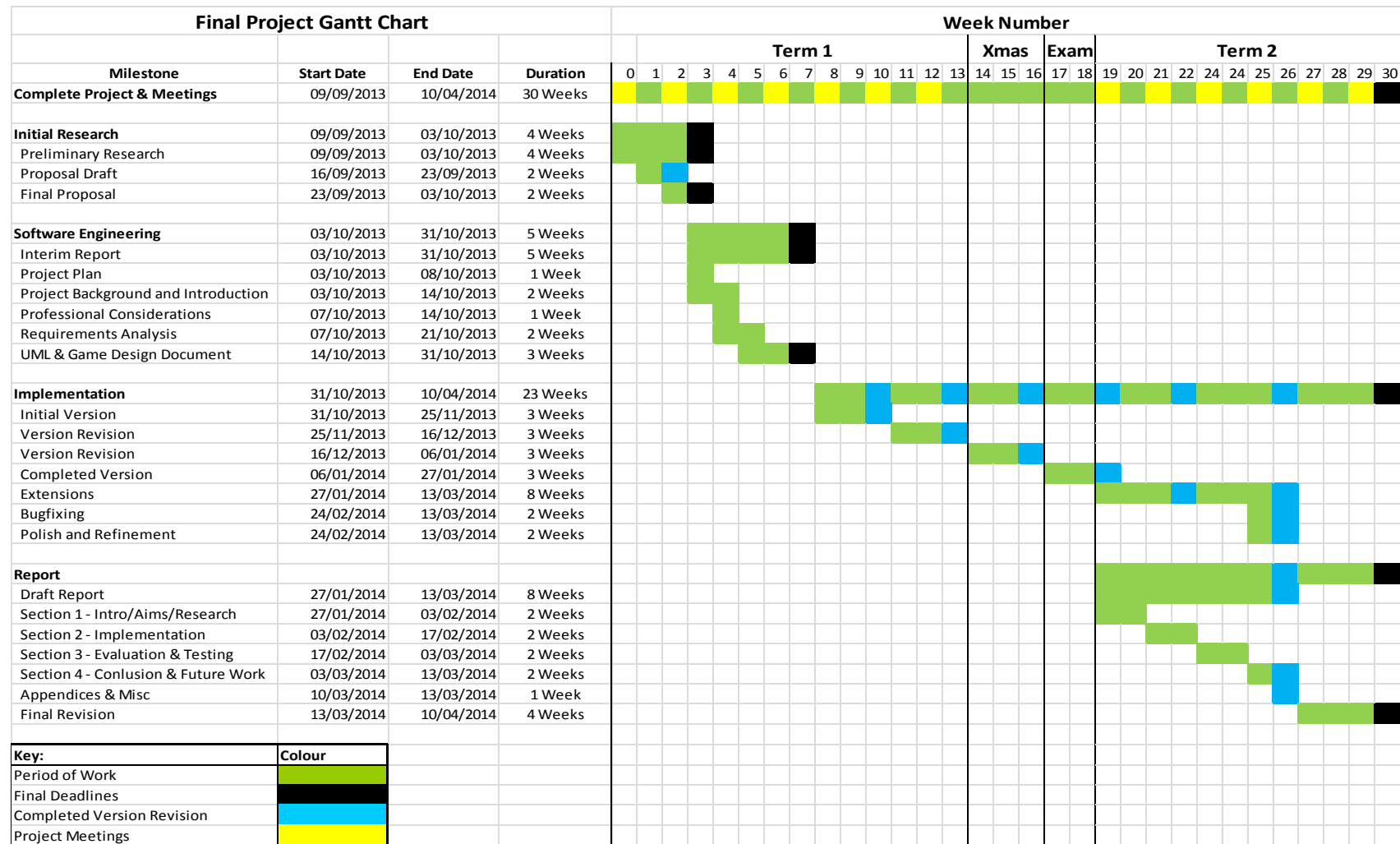


Figure 2 - Gantt Chart of Preliminary Project Schedule

## Bibliography

- [1] How to work with Oculus Rift and Unity? [Online] Unity Community Forum. "AlexFormoso". 2013. Available from: <http://forum.unity3d.com/threads/198219-How-to-work-with-Oculus-Rift-and-Unity> [Accessed September 2013]
- [2] Unity 4 Pro Game Development and Visualization - Student [Online] Studica. Studica. 2013. Available from: <http://www.studica.com/uk/en/unity/unity-pro-4-student.html> [Accessed September 2013]
- [3] Irrlicht Engine [Online] Irrlicht Engine. Irrlicht Engine. 2013. Available from: <http://irrlicht.sourceforge.net/> [Accessed September 2013]
- [4] CRYENGINE - Now with Oculus Rift and Cinemizer support! [Online] Crydev.net. Crytek UK Ltd. 2013. Available from: <http://www.crydev.net/viewtopic.php?f=355&t=114108> [Accessed September 2013]
- [5] Ogre3D Integration [Online] Oculus Developer Center. "tbowren". 2013. Available from: <https://developer.oculusvr.com/forums/viewtopic.php?f=39&t=91&start=20> [Accessed September 2013]
- [6] Torque 3D | Products | GarageGames.com [Online] GarageGames.com. GarageGames.com. 2013. Available from: <http://www.garagegames.com/products/torque-3d> [Accessed September 2013]
- [7] Oculus Developer Forums [Online] Oculus VR Developer Center. Oculus VR. 2013. Available from: <https://developer.oculusvr.com/forums/viewforum.php?f=39> [Accessed September 2013]
- [8] 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> [Accessed September 2013]
- [9] Former Apple Talent Developing Oculus Rift Headtracker, "9 DOF" and 760Hz Gyro Sampling [Online] ROAD TO VR. Lang, B. 2012. Available from: <http://www.roadtovr.com/2012/10/25/oculus-rift-headtracker-adjacent-reality-1695> [Accessed September 2013]
- [10] Oculus VR on why John Carmack joined the Rift revolution [Online] Edge Online. Long, N. 2013. Available from: <http://www.edge-online.com/news/oculus-vr-on-why-john-carmack-joined-the-rift-revolution/> [Accessed September 2013]
- [11] Gregory, J. Game Engine Architecture 1st edn. A K Peters, Ltd., Natick, Ma, USA (2009)

## 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)