CS407: Project Specification

A. Dixon, D. Richardson, G. Taylor and J. Zawadzki

October 2016

1 Introduction

The aim of this project is to implement a virtual reality-based architectural design tool for the purpose of rapid prototyping. This tool will allow users to plan internal and external spaces in both 2 and 3 dimensions. 2-dimensional plans will be extrapolated to create 3D structures, and 3D structures will generate 2D plans. The spaces will be fully dynamic, with user-made changes rendered in real time. The 3D structures will be explorable through the use of virtual reality environments, allowing users to navigate the virtual world by walking in physical space. Creations will be rendered live and in high fidelity as close-to-life scenes.

1.1 Motivation

The concept of *virtual reality* (VR) first entered the mainstream in the early 1990s, with its roots firmly in the gaming sector. The cutting-edge technology of the day, such as *Sega VR*, was pioneering but rudimentary. Recent advances in computing hardware and display technology have once again made VR a viable prospect for consumers and hardware developers alike. The steady influx of VR hardware packages such as the HTC Vive and the Oculus Rift increases accessibility to this revitalised method of interaction, whilst also providing class-leading immersion. The quality and fidelity seen in even the first generation of consumer-grade headsets and controllers reveals countless opportunities for unique and innovative human-computer interaction, much of which can be seen in the first few computer games reaching the VR market. VR has also seen commercial and industrial application; it is often more cost-effective or efficient to train personnel in a virtual reality environment, in business sectors ranging from aviation to warehouse operations.

1.2 Problem Statement

The architectural design process usually requires many iterations of prototyping, modelling, and client input. This process can take months or even years depending on the number of iterations required. Much of this cycle stems from the client being unable to fully conceptualise the design until late in the process; the client will generally see little more than static images or short videos of their designs. A heightened level of immersion would allow for a clearer impression of the design earlier, and lower the communication barrier between designer and client.

To tackle these issues, this project will focus on the development of a virtual-reality architectural design package. The application will be geared towards rapid prototyping and usability, which will allow both architects and clients the ability to design and explore buildings intuitively. Buildings will be hand-crafted and rendered in real-time with “close-to-life" levels of realism by the designer and will be ready to be shown to the customer straight away. The software will be designed to operate in conjunction with a consumer VR hardware packages availible on the market, that include a headset for visualisation and head-tracking. Human-computer interaction will be designed around user interfaces, which are themselves centred on hand-held controllers. This will offer a high level of fidelity in control as well as improving immersion.

2 Background

The current generation of VR hardware started its development during 2012, starting with the first Oculus Rift prototype which included a sole head-mounted device (HMD). Since then, various organisations have continued to develop hardware with innovations such as room-scale VR. Major hardware including the HTC Vive, Oculus Rift, and PlayStation VR was released in mid 2016. Due to the hardware being new, the surrounding software ecosystem has not matured and there are very few applications demonstrating the capabilities of the hardware. Some proof of concept VR design applications have been created since release, such as that by Autodesk [1] or Google's Tilt Brush [2]. Arch Virtual [3] developed a VR architectural visualisation application that facilitated clients to navigate an already modelled space, but there does not exist an application that allows the user to design and architect a building. Neither there is one that is capable of populating it with furniture and altering appearances, taking advantage of the tracked controllers and perspective that VR offers. Other software aimed at the development of VR tools is also available, namely game engines such as Unreal Engine 4 and Unity3D [4], [5]. They handle visual rendering pipelines, hardware interfacing, and asset management.

Consumer grade VR hardware implementations vary in their designs and characteristics, with lower-end hardware such as the Google Cardboard being positioned as a mass-market introduction to VR. Higher-end hardware like the HTC Vive is aimed toward a more niche area of high-end PC gaming. Google's Cardboard has very few features, relying on a smartphone to provide functionality and tracking. The HTC Vive uses a set of infra-red emitting stations, bespoke controllers, and a moderately sized head-mounted display to act as a sensor to perform tracking and display to the user. Alternative products such as the Samsung Gear VR and Oculus Rift perform similar function in their respective categories of low-end and high-end VR hardware. The hardware also addresses the 'scale' of use, or the area within which the user may be able to interact. High-end hardware typically allows for 'roomscale' VR experiences. These experiences allow the user to physically move about the room as a method of interacting with the environment. A table listing the characteristics of currently available VR hardware is shown in Table 1.

VR experiences must also maintain a constant quality of immersion through characteristics such as a minimum refresh rate, and consistent spatial or rotational tracking of the user. LaViola [12] highlights this, describing the side effects of lapses in quality assurance of a VR experience as "cybersickness" of the user of the VR equipment. The work by Barrett [13] classifies the side-effects into the categories of nausea, disorientation, and visual symptoms. Whilst relevant, these publications are dated and do not consider the increased quality of the positional and rotational tracking of modern VR hardware. Additionally, the high resolution of the head-mounted displays poses a challenge for commodity graphics hardware to deliver high quality renditions of the environment at a minimum of 90Hz.

The rendering of 3-dimensional spaces with near-realism is typically done utilising computationally expensive techniques or processes like raytracing and global illumination. Raytracing provides a physically-based rendition of the model, utilising 'rays' that traverse the scene very much like waves of light [14]. Recent advances in graphics hardware capabilities has made this expensive task more feasible for real-time use. Wald, Purcell, Schmittler, et al. [15] have shown it is possible to perform this raytracing in

Name	Platform	Positional Tracking	Control Interface	Roomscale
Google Cardboard	Android 4.1 or above [6]	None	None	No
Google Daydream	Daydream ready smartphones [7]	Smartphone dependant	Wireless handheld controller	No
Gear VR	Samsung flagship smartphone from 2015 onwards	None	Buttons on the headset	No
Playstation VR	Playstation 4	Yes - Stereoscopic cameras and IR sensors [8]	Dualshock 4 controller and Light Wands	2.4x1.9m [9]
Oculus Rift	PC	Yes - Oculus Constellation IR tracking sensors [10]	Xbox One and Oculus Touch controllers	Unspecified
HTC Vive	PC	Yes - 'Lighthouse' emitters [11]	HTC Vive controllers and OpenVR-compatible hardware	4.6x4.6m

Table 1: Virtual reality hardware comparison

real-time with interactive global illumination, accurately simulating the interaction of lights and objects in a scene. Game engines typically will employ deferred or forward rendering pipelines like that by Lauritzen [16] to achieve similar levels of realism. These types of renderers don't offer as much graphical fidelity as raytracing to attain a low frame rendering latency more suited to real-time tasks.

3 Objectives

The aim of this project is to create a VR tool that facilitates the rapid prototyping of interior and exterior spaces. It will allow the user to place walls, doors, windows, and furniture. The interior and exterior designs of these spaces can also be altered with the ability to create and manipulate the properties of furniture, fixtures, and the walls, floors and ceilings of rooms. To fulfil this plan, a number of functional and nonfunctional requirements have been derived. These are listed below, along with a system that describes both the requirement's priority (high, medium, low), whether it is a core objective or a stretch goal, and any dependencies on other objectives it may have.

3.1 Functional Requirements

F1 The tool will have a viewport through which the user can see a rendered world (high, core, independent)

F2 Interaction is provided through a user interface centred around physical VR controllers (high, core, independent)

F3 The world can be interacted with in a dynamic way, allowing the user to modify the environment by (eg.) adding, modifying or removing walls, building components, and objects (high, core, dependant on F2)

F4 Spaces in the world can be constructed using a 2D floor plan system or in a 3D sandbox environment (high, core, dependant on F2, F3)

F5 The tool must run with a minimum refresh rate of 90Hz [17] on the minimum hardware specified in section 4.1. (high, core, independent)

F6 The tool will provide capabilities for saving, loading, and modifying building models or plans (medium, core, independent)

F7 Runtime errors such as code faults should be handled gracefully and should not terminate the program where possible (medium, core, independent)

F8 Environment characteristics such as time of day and light temperature should be modifiable (medium, stretch, dependant on F2)

3.2 Non-functional Requirements

NF1 The software should be intuitive to use, even for new users (high, core, independent)

NF2 Full documentation for setup and use of the software and environment should be provided (medium, core, independent)

NF3 Saved designs should be transferable between different supported computers (medium, core, dependant on F6)

4 Tools

To meet the overarching aims of the project we will be using a number of hardware and software tools for development and testing. These are now discussed and related to the project objectives specified in section 3.

4.1 Hardware

The hardware required to run the final software will be everything included in a typical VR setup using a HTC Vive headset as an I/O device and a desktop computer which will handle all the processing. This kind of setup requires at least 2x1.5m of empty space, but in practise should be slightly greater than that. It is reasonable to expect that the hardware setup might require a dedicated space to run, an example of this is displayed in figure 1. The requirements for the HTC Vive as listed in [17] are the following:

- NVIDIA GeForce GTX 970 /AMD Radeon RX 480 equivalent or greater
- Intel Core i5-4590 equivalent or greater
- 4 GB RAM or more
- Compatible HDMI 1.3 video output
- USB: 1x USB 2.0 port or newer
- Operating system: Windows 7 SP1, Windows 8.1 or later or Windows 10

The development of the tool itself will take place on personal machines and does not require any specific hardware apart from a standard PC.

4.2 Software

The core of the tool will be developed in Unity, which is a game engine that provides a high level of support for developing VR software. The alternatives like the Unreal Development Kit, despite also providing good support for VR lack other features that would make the development using this tool less suitable. In the case of UDK this

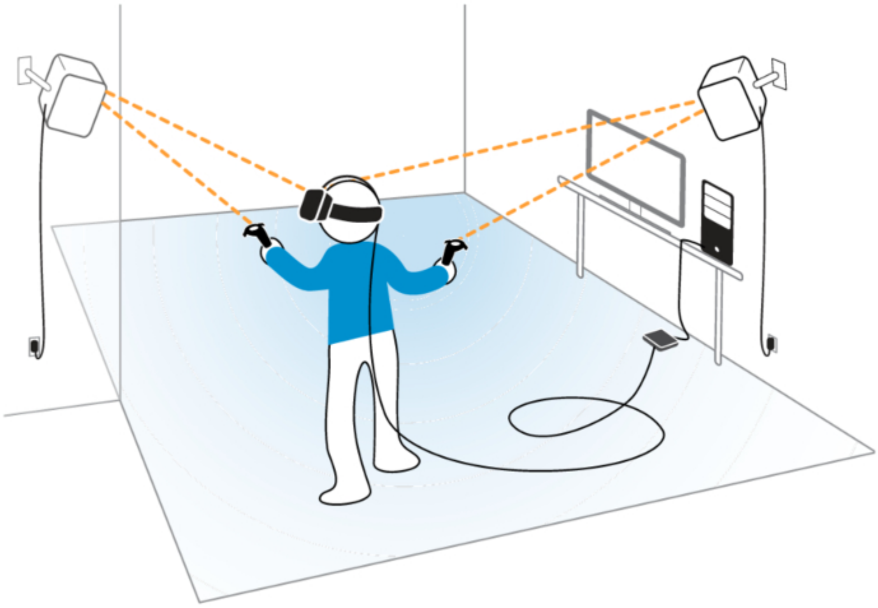

Figure 1: Vive setup

manifests itself as a worse support for dynamic lighting, which would be an important feature for a tool that creates and modifies a lot of objects and aims to provide a realistic look. Steam VR library will be used in addition to that as it provides a set of useful tools that make development of VR tool easier.

In addition to engines and libraries that the final tool will run on, the team will also use other pieces of software that aid in running a group project, like git for code repository or sharelatex.com for collaborative creation of the reports.

5 Project Management

This project will adapt an iterative design process with frequent releases and constant testing and evaluating of the progress so far. Meetings and development sessions will be held at least weekly, during which the responsible team members (see section 5.1) will inform on current progress and any pressing points including tasks to be delegated. This development method will allow for a constant exchange of expertise between team members in case of a roadblock, and a quick response to user feedback later on in the project.

5.1 Team Roles

The team roles listed in table 2 represent the major areas of consideration within the project. The team member(s) listed as responsible will not be the only people working on each area, but they are given ownership over these areas and are responsible for steering the development and organisation of that component. Roles not explicitly listed in table 2, such as managing version control, are considered the joint responsibility of all team members.

Responsibility	Team Member(s)	Description
Project Manager	David Richardson	Team organisation, setting up meetings
UI Design	Alex Dixon	Design and operation of menus, heads-up display and other system elements
Asset Modelling	Alex Dixon & David Richardson	Creation of models for use in the world, including materials and furniture
Audio Engineering	Jakub Zawadzki	Design and implementation of soundscape
Systems Programming	Glynn Taylor	Architecture of underlying data model, its operations, and its interactions with the user interface
Quality Assurance	Jakub Zawadzki	Organisation of testing sessions, and developing a testing methodology

Table 2: Team roles

5.2 Risk Assessment and Mitigation

VR headset access

The primary concern for development and testing is access to the Virtual Reality hardware. The hardware itself is expensive and therefore the group is unable to source a bespoke set. However, we have mitigated this risk by obtaining permission to access three different sets of the same hardware (the HTC Vive) from the Warwick Manufacturing Group, Computing Society and the Game Design Society. This should allow us to use at least one of these at any given time.

Illness and absence

The risk of team member absence is always great. A key member being unable to attend meetings or development sessions may hamper development significantly. The primary method of mitigating this concern is to reduce the amount of restraint and responsibility given to any one person - while they may have areas of concern all team members should be aware of the state of all constituent parts.

Conflict of Interest

Some team members may have differing views on the driving forces and underlying ideals of the design. The impact of this risk is reduced by giving different people ownership of different constituent parts; the individual areas are arbitrated by the listed person. The areas are compatible enough that a difference between two ideals will not restrict development.

Feature creep

The project itself is intentionally open-ended which leaves room for feature creep, the addition of unnecessary or out-of-scope features which may make the final product more bloated. The implementation of a *feature freeze* (no further feature additions) at the end of Project Week 24 will limit the spread of any feature creep.

5.3 Timetable

The project timetable displayed as a Gantt chart in figure 2 gives a general overview of the task scheduling in the project. Where present the arrows between tasks constitute the logical flow of tasks that are dependent on others. Major task milestones are also given for deliverables such as prototypes and for the point of feature freeze to prevent any further feature creep. The deadlines set by the department for the project are then listed in table 3.

5.4 Action Plan

A basic proof of concept of the tool will be developed quickly in term 1 to verify the initial assumptions and see if this kind of tool can work well in practice. With the

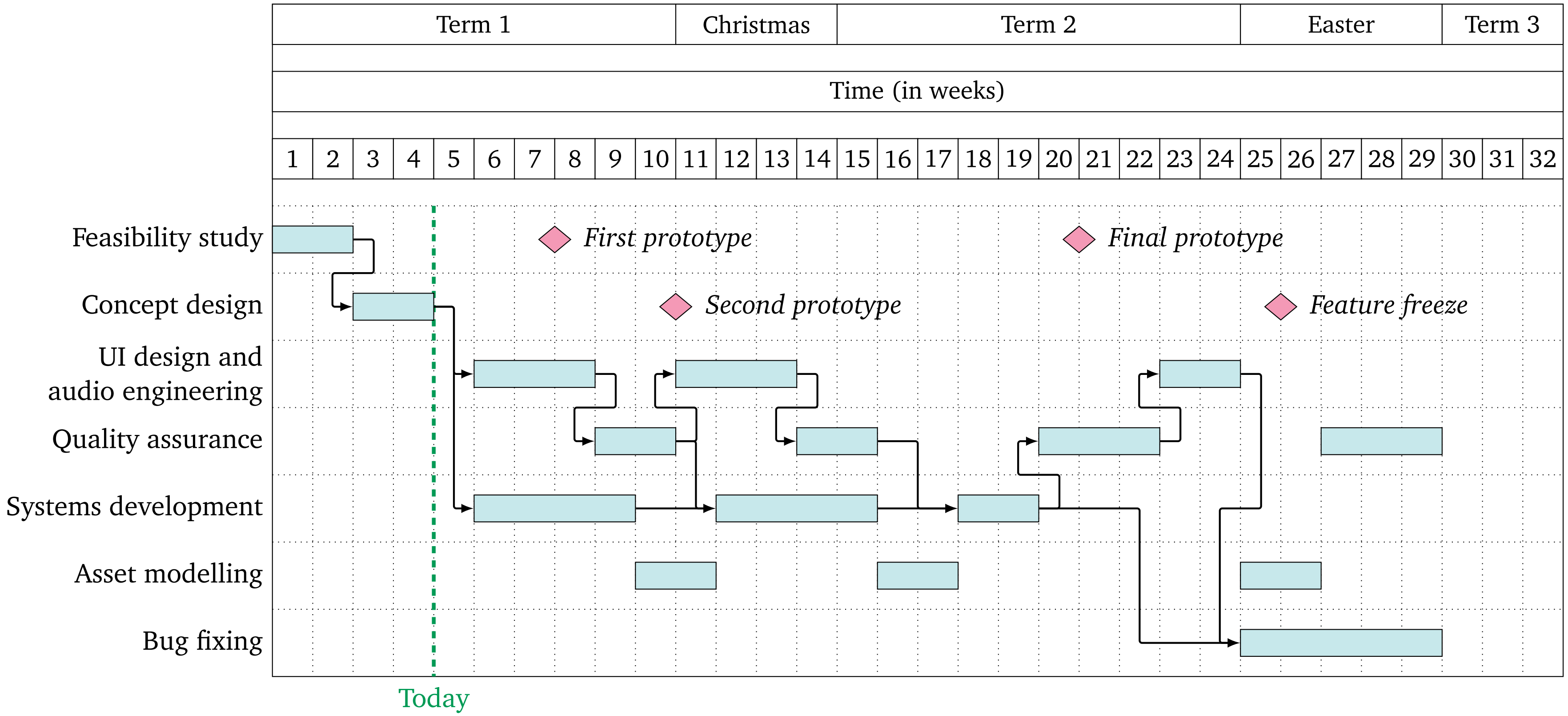

Figure 2: Project timetable from week 1 term 1 to week 3 term 3

Date	Deadline
Thursday 27th October 2016	Project specification hand in
Thursday 6th December 2016	Project progress poster presentation
Friday 17th March 2017	Suggested software development completion date
Friday 28th April 2017	Group and individual project report hand in
Tuesday 9th May 2017	Final project presentation

Table 3: Departmental project deadlines

basic tool created the project will enter a stage of iterative design and new features will constantly be added. Towards the end of term 1 testing of the tool will begin by people from outside the development team to gather feedback on the functionality and interface of the tool.

6 Legal, Social, Ethical and Professional Issues

Our testing of the Architectural tool will mainly be conducted by team members. However, to get an external viewpoint on progress and overall quality of the product it may be necessary to have other people use the system.

While giving the system to the users for testing we will have to bare in mind that there are recorded cases of Virtual Reality technologies having negative effects including nausea and headaches; a waiver would be required on of any testers to ensure they are aware of these potential problems. As negative effects usually only present when the technology is used for protracted periods (more than 30 minutes) [18], testing sessions will be restricted to no more than approximately 20 minutes, with at least 10 minute breaks in between as recommended in the Healthy and Safety guidelines for the Oculus Rift.

Virtual Reality hardware is almost by definition exclusionary to people with some types of disability- notably the visually impaired, and those lacking limb motor function. Unfortunately these issues are far from solved in the general case, and accommodations for these users will not form part of this project. We intend to provide some functionality which minimises difficulty for certain users however, such as ensuring the software is colour-blind-accessible by avoiding matching colour pairs.

References

[1] Ethan Arnowitz and Phan Lee and John Scheeler and Dave Yan. *Autodesk Summer Interns: Design in Virtual Reality*. https://www.youtube.com/watch?v=xUZXFbeTskI. Online; accessed 17th October 2016. 2016.

[2] Google. *Introducing Tilt Brush*. https://www.tiltbrush.com/. Online; accessed 17th October 2016. 2016.

[3] Jon Brouchoud. *BIM Goes Virtual: Oculus Rift and virtual reality take architectural visualization to the next level*. http://archvirtual.com/2014/01/19/bim-goes-virtual-oculus-rift-and-virtual-reality-take-architectural-visualization-to-the-next-level/. Online; accessed 16th October 2016. 2014.

[4] Epic Games, Inc. *VR is here - What's your vision?* https://www.unrealengine.com/vr. Online; accessed 22nd October 2016. 2016.

[5] Unity Technologies. *Unity for VR and AR - Leading the revolution*. https://unity3d.com/unity/multiplatform/vr-ar. Online; accessed 22nd October 2016. 2016.

[6] Google. *Requirements for using Google Cardboard*. https://support.google.com/cardboard/answer/6295091?hl=en. Online; accessed 22nd October 2016. 2014.

[7] Google. *Phones built for virtual reality - Daydream ready from the start*. https://vr.google.com/intl/en_uk/daydream/phones/. Online; accessed 22nd October 2016. 2016.

[8] Sony Interactive Entertainment. *PlayStation VR*. https://www.playstation.com/en-gb/explore/playstation-vr/. Online; accessed 22nd October 2016. 2016.

[9] Ben Kuchera and Charlie Hall. *PlayStation VR supports room-scale experiences, clearing the way for Vive ports*. http://www.polygon.com/2016/3/17/11254170/playstation-vr-vive-rift-tracking-area. Online; accessed 22nd October 2016. 2016.

[10] Jamie Feltham. *Palmer Luckey Explains Oculus Rift's Constellation Tracking and Fabric*. http://www.vrfocus.com/2015/06/palmer-luckey-explains-oculus-rifts-constellation-tracking-and-fabric/. Online; accessed 22nd October 2016. 2015.

[11] Sean Buckley. *This Is How Valve's Amazing Lighthouse Tracking Technology Works*. http://gizmodo.com/this-is-how-valve-s-amazing-lighthouse-tracking-technol-1705356768. Online; accessed 22nd October 2016. 2015.

[12] Joseph J. LaViola Jr. "A Discussion of Cybersickness in Virtual Environments". In: *SIGCHI Bull.* 32.1 (Jan. 2000), pp. 47–56. ISSN: 0736-6906. DOI: 10.1145/333329.333344. URL: http://doi.acm.org/10.1145/333329.333344.

[13] Judy Barrett. *Side effects of virtual environments: A review of the literature*. Tech. rep. DTIC Document, 2004.

[14] Andrew S Glassner. *An introduction to ray tracing*. Elsevier, 1989.

[15] Ingo Wald, Timothy J Purcell, Jörg Schmittler, et al. "Realtime ray tracing and its use for interactive global illumination". In: *Eurographics State of the Art Reports* 1.3 (2003), p. 5.

[16] Andrew Lauritzen. "Deferred rendering for current and future rendering pipelines". In: *SIGGRAPH Course: Beyond Programmable Shading* (2010), pp. 1–34.

[17] Digital Trends Staff. *Spec Comparison: Does the Rift's Touch Update Make It a True Vive Competitor?* http://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/. Online; accessed 22nd October 2016. 2016.

[18] Occulus. *Health and Safety*. https://static.oculus.com/documents/310-30023-01_Rift_HealthSafety_English.pdf. Online; accessed 24th October 2016. 2016.