**The Lab 2**

Final Report for CS39440 Major Project

*Author*: Shankly Richard Cragg ([shc27@aber.ac.uk](mailto:shc27@aber.ac.uk))

*Supervisor*: Dr./Prof. David Hunter (dah56@aber.ac.uk)

26th April 2017

Version 1.0 (Draft)

This report is submitted as partial fulfilment of a BSc degree in  
Computer Science (G401)

Department of Computer Science

Aberystwyth University

Aberystwyth

Ceredigion

SY23 3DB

Wales, UK

**Declaration of originality**

I confirm that:

* This submission is my own work, except where clearly indicated.
* I understand that there are severe penalties for Unacceptable Academic Practice, which can lead to loss of marks or even the withholding of a degree.
* I have read the regulations on Unacceptable Academic Practice from the University’s Academic Quality and Records Office (AQRO) and the relevant sections of the current Student Handbook of the Department of Computer Science.
* In submitting this work I understand and agree to abide by the University’s regulations governing these issues.

Name Shankly Richard Cragg

Date 26th April 2018

**Consent to share this work**

By including my name below, I hereby agree to this dissertation being made available to other students and academic staff of the Aberystwyth Computer Science Department.

Name Shankly Richard Cragg

Date 26th April 2018

**Acknowledgements**

I am grateful to…

I’d like to thank…

I’d like to thank my supervisor, David Hunter, for being a valuable source of knowledge, and great sounding board for ideas.

**Abstract**

Include an abstract for your project. This should be no more than 300 words.

With the ever-increasing presence and development of virtual reality in the modern world [1], exploring virtual reality as a medium is of paramount importance as mainstream appeal grows.

This project takes inspiration from Valves “The Lab” [2], and Owlchemy Labs “Job Simulator” [3], and looks to create an experience which is difficult or dangerous to recreate in physical reality, taking full advantage of the strengths inherent in virtual reality as a medium.

“The Lab 2” is therefore an introductory tool to the possible experiences virtual reality can give which cannot be taught better any other way. Specifically, this project looks to give the unusual experience of being a Steam Engine Fireman, who must keep his train running under the Drivers orders.

With a coherent play space, and audio/visual feedback, this experience should be functional as a user’s first virtual reality experience, while offering something to those who are familiar with the platform.

This report details the process of a developing with virtual reality for the first time, and the process used to ensure development went as smoothly as possible. With analysis of the challenges a 1-man team faces when trying to develop a feature-rich and complete experience.

**Contents**

1. Background, Analysis & Process 4

1.1. Background 4

1.2. Analysis 4

1.3. Process 4

2. Design 4

2.1. Overall Architecture 4

2.2. Detailed Design 4

2.2.1. Even More Detail 4

2.3. User Interface Design 4

2.4. Other Relevant Sections 4

3. Implementation 4

4. Testing 4

4.1. Overall Approach to Testing 4

4.2. Automated Testing 4

4.2.1. Unit Tests 4

4.2.2. User Interface Testing 4

4.2.3. Stress Testing 4

4.2.4. Other Types of Testing 4

4.3. Integration Testing 4

4.4. User Testing 4

5. Critical Evaluation 4

6. Appendices 4

A. Third-Party Code and Libraries 4

B. Ethics Submission 4

C. Code Samples 4

Annotated Bibliography 4

# Background, Analysis & Process

This section should discuss your preparation for the project, including background reading, your analysis of the problem and the process or method you have followed to help structure your work. It is likely that you will reuse part of your outline project specification, but at this point in the project you should have more to talk about.

**Notes:**

* All of the sections and text in this example are for illustration purposes. The main Chapters are a good starting point, but the content and actual sections that you include are likely to be different.
* Look at the document on the MMP: Final Report and Technical Work [7] for additional guidance.

## Background

What was your background preparation for the project? What similar systems did you assess? What was your motivation and interest in this project?

Valve Corporation is an American video game developer and digital distribution company. They are the developers of a video game released in April 5th, 2016 called “The Lab”. This was Valve attempting to “Understand existing genres though the lens of VR” [Rock paper shotgun article]. The Lab features 8 short and varied experiences, ranging from Archery to a visual representation of the Solar System [include steam link to the lab].

“The Lab” has the difficult task of being the first virtual reality experience for a huge number of consumers, not only in 2016 when the game was released, but even now in 2018 and beyond. Valve accomplish this with comical slapstick humour and simple intuitive controls and tutorials aplenty. The Lab is easy to pick up and play with no prior experience with the virtual realm to this day, with the game’s polish and wide appeal a testament to the developers.

This project aims at acting as a spiritual successor to the fun fuelled mini-game collection by Valve, with a similar focus on a unique scenario and fun intuitive interaction with Objects and the Environment.

The inspiration for the theme and feel of “The Lab 2” and the Steam Engine Fireman idea came from Owlchemy Labs “Job Simulator”. This game features a range of real world jobs turned comical such as “Auto Mechanic” and “Gourmet Chef”. These exaggerated experiences take place in an environment many users will be intimately familiar with. For the project a decision was made to take the dynamic of hectic interactive gameplay in a workplace and look for a dangerous or difficult to recreate scenario in the real world to apply these criteria to.

After some brainstorming, I came upon the role of a “stoker”, or “fireman”, whose job it is to tend to the fire of a steam engine, by shovelling coal into the engines firebox.

My motivation for this project stems from the exciting range of experiences afforded to me during my Industrial Year. I had the chance to work with Augmented Reality (AR) using the Microsoft HoloLens, which similarly to virtual reality development can make use of cross-platform game engine Unity. This experience with AR development was a joyous one and having been introduced to virtual reality for the first time with their internal HTC Vive kit gave me great curiosity into immersing myself into the world of virtual reality development. This project is a great opportunity to further develop this skillset.

## Analysis

Taking into account the problem and what you learned from the background work, what was your analysis of the problem? How did your analysis help to decompose the problem into the main tasks that you would undertake? Were there alternative approaches? Why did you choose one approach compared to the alternatives?

There should be a clear statement of the objectives of the work, which you will evaluate at the end of the work.

In most cases, the agreed objectives or requirements will be the result of a compromise between what would ideally have been produced and what was felt to be possible in the time available. A discussion of the process of arriving at the final list is usually appropriate.

As mentioned in the lectures, think about possible security issues for the project topic. Whilst these might not be relevant for all projects, do consider if there are relevant for your project. Where there are relevant security issues, discuss how they will this affect the work that you are doing. Carry forward this discussion into relevant areas for design, implementation and testing.

The project is to create a virtual reality experience, and as such my research involved the playtesting of existing games found on the Steam platform.

The research done showed that the best VR experiences focus on interaction as main motivation of play. For example, in figure 1 below, Valves “The Lab” allows users to travel to exotic and beautiful spaces to explore, including a Venice town square, or the top of a mountain in the Icelandic wilderness. However, the captivating thing to do in these areas is play with some sticks present in all locations. Throwing them, juggling them, watching them bounce off things. This focus on physical interaction and gameplay came as the standout experience even in the face of such beautiful vistas.

Figure "Postcards" from The Lab



This ensured that the very core of my gameplay loop was going to encourage constant interaction with the elements in the world.

Something Owlchemy Labs “Job Simulator” had to offer was the vibrant and expansive experience that can be offered in small spaced.

Figure Job Simulator Office Level



The variety of interaction in such a small space was compelling gameplay. As a 1-man project, it made a huge amount of sense to limit the design space to as small a space as possible in order to reduce the amount of environment design necessary, an area I feel particularly uncomfortable with going into the project.

Regarding hardware there are 2 main competitors when it comes to virtual reality Systems and Head Mounted Displays (HMDs). These are the HTV Vive, and the Oculus Rift. These direct competitors have a lot in common, and the choice of platform for most users comes down to price point and experiences which are exclusive to each system.

For this project, I am developing with the HTC Vive, as the University Campus features a room complete with a HTC Vive system. Because of this, I will also acquire a HTC Vive to aid development off-site and keep playtesting as convenient and available as possible.

Similarly, there are 2 options concerning game engine to develop on as well. These are Epic Games “Unreal Engine 4”, and Unity Technologies own “Unity”. Both are cross-platform game engines free to use for development by individuals or for educational purposes. Once again this comes down to personal preference. I have some prior experiences working within Unity, and when researching development kits, the most popular one found named “VRTK” (Virtual Reality Toolkit) is exclusive to Unity.

VRTK is self-described as “A productive VR Toolkit for rapidly building VR solutions in Unity3d”. It includes a lot of basic models and interactions which are necessary to even begin developing an VR experience. Including but not limited to:

* Locomotion within virtual space
* Interaction
  + touching
  + grabbing
* Body physics
* 3D models for the HMD and controls

This allows us to focus more on creating an immersive experience, rather than being bogged down in the complexities of modelling interaction between the real-world controls and the digital one.

## Objectives

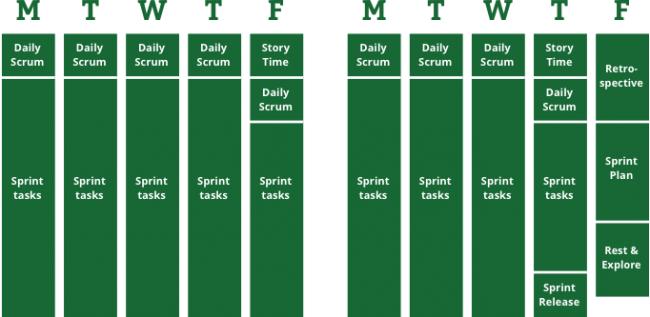
* Consistent frame rate greater than 90 frames per second.
* Dangerous or difficult to recreate scenario.
* Consistent interaction enforced through game design.
* The program must run on the HTC Vive.

## Process

You need to describe briefly the life cycle model or research method that you used. You do not need to write about all of the different process models that you are aware of. Focus on the process model that you have used. It is possible that you needed to adapt an existing process model to suit your project; clearly identify what you used and how you adapted it for your needs.

For this project, I decided to use a modified version of the agile methodology Scrum. When researching single team development practices, I came across an article from Alex Andrews titled “Scrum Of One: How to Bring Scrum into your One-Person Operation”. [SCRUM REFERENCE HERE] Here he discussed a 2-week sprint cycle, shown in figure 3 to visualize his process.

Figure Alex Andrew's 2-week sprint

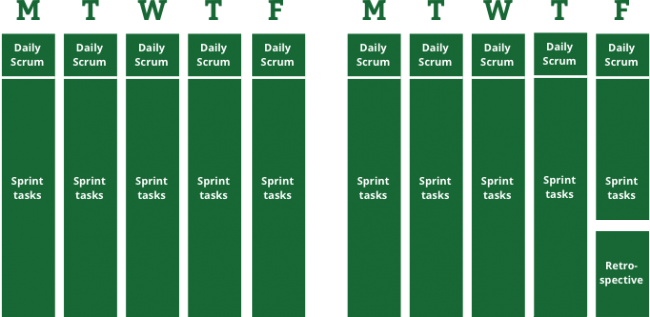


There were a few changes made to this schedule. The major made change to the schedule was the removal of “Story time” as a time dedicated to the direction of the project. As the project was working in unfamiliar space, sizings and the direction of the project was going to be incredibly fluid. This was instead done dynamically during the Daily Scrum at the start of each day, and reflectively at the end of the day as I wrote my daily diary.

The “Big picture” would instead be thought about in the bi-weekly retrospective. At the end of every 2 weeks the previous week would be reviewed, with concern regarding about which tasks were easier, and which tasks were harder than expected. This allows maximum focus on development time and gaining experience with the technology and give more leeway to on the fly decisions being made as new knowledge is gained.

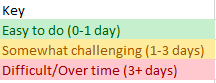
This agile approach suits the project due to the open-ended nature of the project, and the lack of experience working with the technology. An agile process allows more flexibility over the day to day activities compared to a plan driven approach. The streamlined process used can be seen in figure 4.

Figure Adapted variation on the 2-week sprint



To keep track of tasks to be done, were in the process of doing, and had been completed, a task list was kept in an excel document. This had a key for the time it would take to accomplish a task seen in figure 5.

Figure Key for the task list



These tasks were considered at the start and end of every working day as I wrote my daily diary. As iterations continued, new tasks could be added, or their expected sizing changed based on previous experiences. The main goal of this was to ensure that what amounts to a sprints worth of work was being completed every sprint. Naturally this wasn’t always the case due to inaccurate sizings, but this made sure progress was consistent, and kept track of while multiple tasks were being developed.

At the start of the project, a few user stories were created to give some concrete end goals for the project. Some of these are epics which struggled to be narrowed down to smaller concrete user stories, such as;

*“As a player, I want a high FPS so that I avoid feelings of motion sickness and lack of immersion.”*

This is not a goal which can be completed within a single sprint, and it cannot be compartmentalised into smaller stories, as optimisation of individual parts of the system can only come about after each in game element is added. The stories mentioned can be seen in Appendix [APPENDIX WITH STORIES HERE].

# Design

You should concentrate on the more important aspects of the design. It is essential that an overview is presented before going into detail. As well as describing the design adopted it must also explain what other designs were considered and why they were rejected.

The design should describe what you expected to do, and might also explain areas that you had to revise after some investigation.

Typically, for an object-oriented design, the discussion will focus on the choice of objects and classes and the allocation of methods to classes. The use made of reusable components should be described and their source referenced. Particularly important decisions concerning data structures usually affect the architecture of a system and so should be described here.

How much material you include on detailed design and implementation will depend very much on the nature of the project. It should not be padded out. Think about the significant aspects of your system. For example, describe the design of the user interface if it is a critical aspect of your system, or provide detail about methods and data structures that are not trivial. Do not spend time on long lists of trivial items and repetitive descriptions. If in doubt about what is appropriate, speak to your supervisor.

You should also identify any support tools that you used. You should discuss your choice of implementation tools - programming language, compilers, database management system, program development environment, etc.

Some example sub-sections may be as follows, but the specific sections are for you to define.

## Overall Architecture

## Detailed Design

### Even More Detail

## User Interface Design

## Other Relevant Sections

# Implementation

The implementation should look at any issues you encountered as you tried to implement your design. During the work, you might have found that elements of your design were unnecessary or overly complex; perhaps third party libraries were available that simplified some of the functions that you intended to implement. If things were easier in some areas, then how did you adapt your project to take account of your findings?

It is more likely that things were more complex than you first thought. In particular, were there any problems or difficulties that you found during implementation that you had to address? Did such problems simply delay you or were they more significant?

You can conclude this section by reviewing the end of the implementation stage against the planned requirements.

# Testing

Detailed descriptions of every test case are definitely not what is required here. What is important is to show that you adopted a sensible strategy that was, in principle, capable of testing the system adequately even if you did not have the time to test the system fully.

Provide information in the body of your report and the appendix to explain the testing that has been performed. How does this testing address the requirements and design for the project?

How comprehensive is the testing within the constraints of the project? Are you testing the normal working behaviour? Are you testing the exceptional behaviour, e.g. error conditions? Are you testing security issues if they are relevant for your project?

Have you tested your system on “real users”? For example, if your system is supposed to solve a problem for a business, then it would be appropriate to present your approach to involve the users in the testing process and to record the results that you obtained. Depending on the level of detail, it is likely that you would put any detailed results in an appendix.

The following sections indicate some areas you might include. Other sections may be more appropriate to your project.

## Overall Approach to Testing

## Automated Testing

### Unit Tests

### User Interface Testing

### Stress Testing

### Other Types of Testing

## Integration Testing

## User Testing

# Critical Evaluation

Examiners expect to find in your dissertation a section addressing such questions as:

* Were the requirements correctly identified?
* Were the design decisions correct?
* Could a more suitable set of tools have been chosen?
* How well did the software meet the needs of those who were expecting to use it?
* How well were any other project aims achieved?
* If you were starting again, what would you do differently?

Other questions can be addressed as appropriate for a project.

Such material is regarded as an important part of the dissertation; it should demonstrate that you are capable not only of carrying out a piece of work but also of thinking critically about how you did it and how you might have done it better. This is seen as an important part of an honours degree.

There will be good things and room for improvement with any project. As you write this section, identify and discuss the parts of the work that went well and also consider ways in which the work could be improved.

In the latter stages of the module, we will discuss the evaluation. That will probably be around week 9, although that differs each year.

# Appendices

The appendices are for additional content that is useful to support the discussion in the report. It is material that is not necessarily needed in the body of the report, but its inclusion in the appendices makes it easy to access.

For example, if you have developed a Design Specification document as part of a plan-driven approach for the project, then it would be appropriate to include that document as an appendix. In the body of your report you would highlight the most interesting aspects of the design, referring your reader to the full specification for further detail.

If you have taken an agile approach to developing the project, then you may be less likely to have developed a full requirements specification. Perhaps you use stories to keep track of the functionality and the ’future conversations’. It might not be relevant to include all of those in the body of your report. Instead, you might include those in an appendix.

There is a balance to be struck between what is relevant to include in the body of your report and whether additional supporting evidence is appropriate in the appendices. Speak to your supervisor or the module coordinator if you have questions about this.

* 1. Third-Party Code and Libraries

If you have made use of any third party code or software libraries, i.e. any code that you have not designed and written yourself, then you must include this appendix.

As has been said in lectures, it is acceptable and likely that you will make use of third-party code and software libraries. If third party code or libraries are used, your work will build on that to produce notable new work. The key requirement is that we understand what is your original work and what work is based on that of other people.

Therefore, you need to clearly state what you have used and where the original material can be found. Also, if you have made any changes to the original versions, you must explain what you have changed.

As an example, you might include a definition such as:

**Apache POI library** – The project has been used to read and write Microsoft Excel files (XLS) as part of the interaction with the client’s existing system for processing data. Version 3.10-FINAL was used. The library is open source and it is available from the Apache Software Foundation [5]. The library is released using the Apache License [6]. This library was used without modification.

* 1. Ethics Submission

This appendix includes a copy of the ethics submission for the project. After you have completed your Ethics submission, you will receive a PDF with a summary of the comments. That document should be embedded in this report, either as images, an embedded PDF or as copied text. The content should also include the Ethics Application Number that you receive.

* 1. Code Samples

This is an example appendix. Include as many appendices as you need. The appendices do not count towards the overall word count for the report.

For some projects, it might be relevant to include some code extracts in an appendix. You are not expected to put all of your code here - the correct place for all of your code is in the technical submission that is made in addition to the Final Report. However, if there are some notable aspects of the code that you discuss, including that in an appendix might be useful to make it easier for your readers to access.

As a general guide, if you are discussing short extracts of code then you are advised to include such code in the body of the report. If there is a longer extract that is relevant, then you might include it as shown in the following section.

Only include code in the appendix if that code is discussed and referred to in the body of the report.

Random Number Generator

The Bayes Durham Shuffle ensures that the pseudo random numbers used in the simulation are further shuffled, ensuring minimal correlation between subsequent random outputs.

// Some example code here…

# Annotated Bibliography

This final section should list all relevant resources that you have consulted in researching your project. Each reference should also include a brief annotation.

1. Sylvia Duckworth. A picture of a kitten at Hellifield Peel. <http://www.geograph.org.uk/photo/640959>, 2007. Copyright Sylvia Duckworth and licensed for reuse under a Creative Commons Attribution-Share Alike 2.0 Generic Licence. Accessed August 2011.  
     
   This is my annotation. I should add in a description here.
2. Mark Neal, Jan Feyereisl, Rosario Rascunà, and Xiaolei Wang. Don’t touch me, I’m fine: Robot autonomy using an artificial innate immune system. In *Proceedings of the 5th International Conference on Artificial Immune Systems*, pages 349–361. Springer, 2006.   
     
   This paper…
3. W.H. Press et al. *Numerical recipes in C*. Cambridge University Press Cambridge, 1992.  
     
   This is my annotation. I can add in comments that are in **bold** and *italics*and then further content.
4. Various. Fail blog. <http://www.failblog.org/>, August 2011. Accessed August 2011.  
     
   This is my annotation. I should add in a description here.
5. Apache Software Foundation (2014) “*Apache POI - the Java API for Microsoft Documents*” (Online) Available at: <http://poi.apache.org> Accessed: 14th March 2014.
6. Apache Software Foundation (2004) “Apache License, Version 2.0” (Online) Available at: <http://www.apache.org/licenses/LICENSE-2.0> Accessed: 14th March 2014.
7. Neil Taylor, “MMP: Final Report and Technical Work”, 2017 (Online) Available at: <http://blackboard.aber.ac.uk/> Accessed 26th April 2017.

A document that outlines information about the marking guide for the Final Report and Technical Work. This document was referred to as Structure of the Final Report before academic year 2016-2017. This is published in the Assignments folder. If you are logged in to Blackboard, you can access the folder using <http://jump.aber.ac.uk/?sxxpt>.