Capstone Project 36:

Virtual Reality Livestream

Validation Document

Group 36:

Kaleb Graham

Alex Macdonald

Atif Murtaza Mahmud

Yingmin Tu

Andrey Varlamov

Table Of Contents

Changelog

Date	Author	Location	Change
2020-02-09	AV	Section 2.3	Added milestone 3 demo summary
2020-02-09	AV	Section 3	Minor changes
2020-04-05	AM	Section 4	Completed test results
2020-04-06	YT	Section 3.3	Framerate
2020-04-07	AMM	Section 3.3	Latency
2020-04-07	AM	Section 4.5	Added info about client side hardware
2020-04-08	KG	Section 2	Switch tense to present tense instead of future tense. Update Testing strategy and methodology to match what was performed. Remove summary of tests section
2020-04-08	KG	Section 3.1	Update unit-testing section
2020-04-08	KG	Section 3.4	Update user-testing section
2020-04-08	KG	Section 6	Add appendix
2020-04-08	AMM	Section 3	Added step-by-step guidelines to recreate tests

1. Introduction

Validation is important in this project, as unsatisfied functional and non-functional requirements might result in motion sickness and disorientation in users. This document outlines how each requirement has been validated. In addition, general testing strategies such as unit testing and user testing will be explained. Finally, the results of the tests will be covered in section 4.

Please note, throughout the rest of the document the terms “client” and “student” are used interchangeably, and “server” and “instructor” are used interchangeably.

2. Testing Scope

2.1. Testing Strategy

The testing strategy originally planned was to develop automated unit-tests for all software written, and to perform functional (see section 3.2), and performance (see section 3.3) tests that ensure that the functional and non-functional requirements are met. Lastly, (see section 3.4), user testing was planned. However, due to certain constraints, only functional and performance tests were completed. See individual sections for more detail.

2.2. Testing Methodology

All functional and performance tests were performed manually on personal devices with the final build of the project. Performance testing was performed on a Thinkpad X220 device, which has 8GB RAM, an Intel i5-2520M processor, and was connected to an external GeForce GTX 1050 Ti GPU.

Note that this is significantly worse specifications than the minimum specifications outlined in the Requirements document (**C4)**. Unfortunately, due to the Covid-19 pandemic, this was the most powerful device available for testing. Performance tests would be expected to be better if using a device that met the minimum requirements.

All scenes were run using a custom Unity project. See the appendix for more information about this project.

3. Test Plan

3.1. Unit Testing

Originally, a plan was devised to write unit-tests using Visual Studio's unit-testing framework. Unit tests are useful when a codebase needs to be maintained, as they can be run automatically after editing code to ensure that changes to one part of the code did not unintentionally break the functionality of any other part of the code. The core idea is that any developer can run unit tests after making changes to help ensure that the project will still meet functional requirements, without having to run manual tests for confirmation. However, due to time constraints, no unit-tests were written. Note that all functional and non-functional requirements were still tested (see section 3.2 and 3.3).

3.2. Functional Testing

Test plan for FR1 - VR Viewing

1. Launch the VR application on the server side
2. Connect the client device to the server IP
3. Take turns viewing the server and client sides

Pass Condition

The client and server sides show the same VR environment

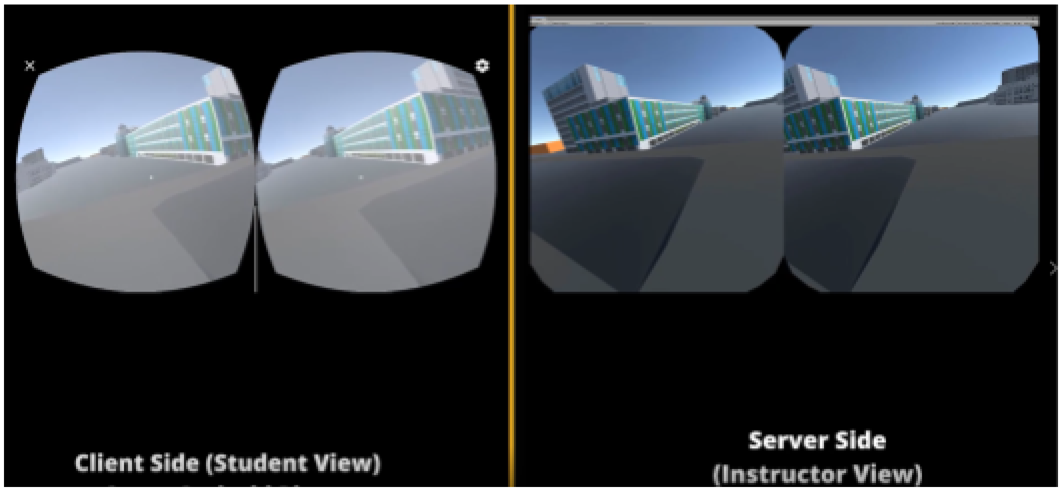


Figure 1: Pass condition for FR1

Test plan for FR2 - Student Degrees of Freedom

1. Launch the VR application on the server side
2. Connect the client device to the server IP
3. Maintain your position and look around from the server side
    a. Check to see that student view does not change
4. Look around in the client side
    a. Check to see if you can look around the scene

Pass Condition

Students can look around a 360 degree image, regardless of the instructor's camera angle.

Test plan for FR3 - Instructor/Student Devices

1. Launch the VR server application on the server side
2. Launch VR client application on client device

Pass Condition

Server and client applications are running on different devices.

3.3. Performance Testing

Test plan for NFR1 - Resolution on the client side

1. Set the resolution parameter to be at least 2048x2048p on the server side and ensure that the server runs without hindering any other FR/NFR by running the other tests. Note that this is the smallest possible resolution that can be used in Unity that is greater than or equal to 1080p (1920x1080p).

Pass Condition

The server is able to transmit a minimum resolution of 2048x2048p without any of the other requirement tests failing.

Test plan for NFR2 - Latency

1. Place a UI element on the server side that displays the current time. A UI element for this test can be found in the project's GitHub repo under Assets-> Prefabs ->TimeSign.prefab
2. Run the livestream
3. Place the server and client devices side by side such that the timestamp on both are visible
4. Record the difference in timestamp (this will show how far behind the client is, hence the latency)
5. Repeat measurements multiple times
6. Take the average of the readings

Pass Condition

The measured latency is below the NFR2 as outlined in our requirements document

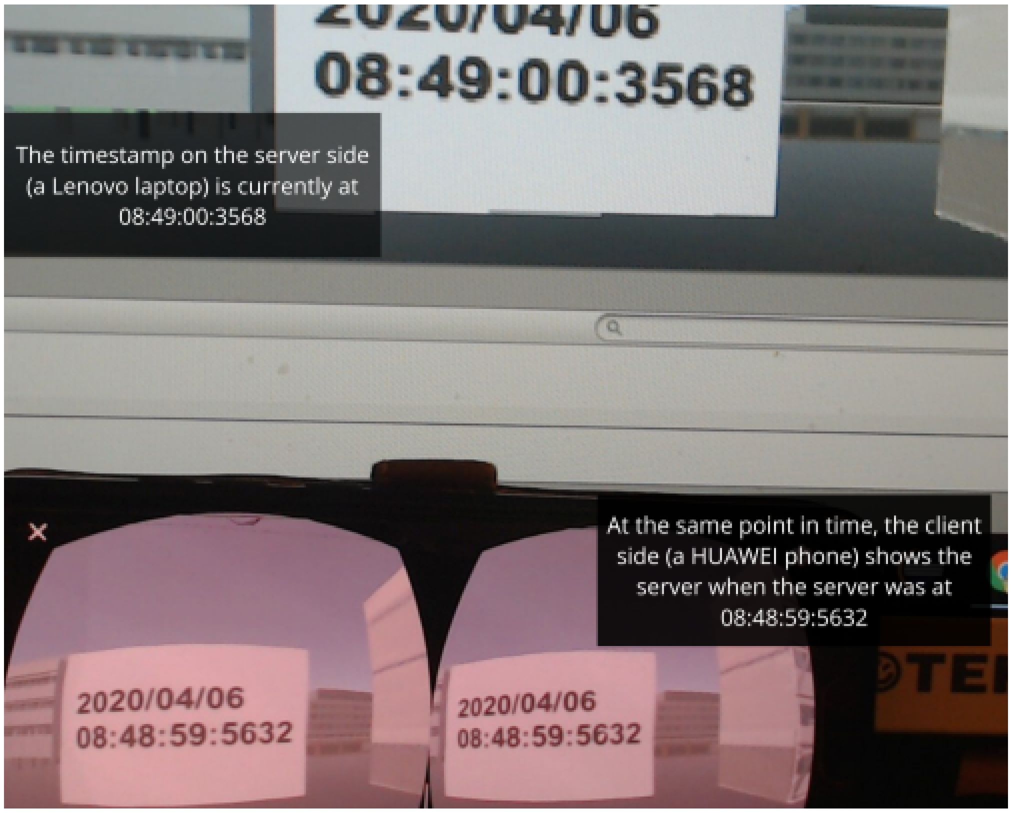

2020/04/06
08:49:00:3568
The timestamp on the server side (a Lenovo laptop) is currently at 08:49:00:3568
At the same point in time, the client side (a HUAWEI phone) shows the server when the server was at 08:48:59:5632
2020/04/06
08:48:59:5632
2020/04/06
08:48:59:5632

Figure 2. Latency Test Setup

Test plan for NFR3 - Framerate

1. Enable FPS debug info by toggling on the “DEBUG” flag
2. Run a demonstration with the frame rate test suite enabled.

Pass Condition

The measured frame rates are above the value outlined in NFR3 of our requirements document

Test plan for NFR4 - Number of Clients

1. Run the livestream demo with at least 3 mobile devices connected (as clients) to the server

Pass Condition

The demo can be completed successfully without failing any of the other FR/NFR tests

Test plan for NFR5 - Minimum Supported Specifications

1. Run the demonstration on the EML Alienware laptop and ensure none of the FR/NFR tests fail

Pass Condition

All the tests that were done in this validation ran on the hardware specified in NFR5.

Note on NFR5: At the time of testing, we did not have access to the required hardware, and instead tested on a personal laptop. This laptop has much lower specifications than the minimum, so any requirements that pass on the personal laptop can be assumed to also pass on the required hardware.

3.4. User Testing

Ideally, user-testing would have been performed to ensure that the final product is well received by end-users and to validate that the solution results in a good user experience. However, due to the Covid-19 pandemic, in person meetings could not be conducted and therefore user-testing was not possible.

4. Test Results

4.1. Functional Testing

ID	Requirement	Test Result
FR1	VR Viewing	**Pass -** Same scene was viewable on instructor-device and client-device
FR2	Student Degrees-Of-Freedom	**Pass -** Student’s field-of-view is independent of instructor’s field-of-view. Changes in the instructor's position were matched on the student-side.
FR3	Instructor/student Device(s)	**Pass -** Instructors streaming software ran on a laptop computer. All clients connected using separate mobile phones.

4.2. Performance Testing

ID	Requirement	Test Result
NFR1	Resolution	**Fail** - Resolution has been set to 1024x1024 pixels, which is less than the 1080p resolution specified. Note that 1080p resolution is 1920x1080 pixels. Due to technical reasons, the next available resolution is 2048x204 pixels, which is not runnable while simultaneously meeting NFR2 and NFR3.
NFR2	Latency	**Pass -** Average latency recorded: 0.63 seconds < 5 seconds For more information**,** please see Table 1.
NFR3	Framerate	**Pass** - Instructor-side framerate: 39 fps. Student-Side framerate ~33fps.
NFR4	Number of Clients	**Pass** - Ran simultaneously on 3 separate mobile phones.
NFR5	Minimum Supported Specifications	**Pass** - This test was performed on a laptop that is worse than the minimum supported specifications. It was performed on a Thinkpad X220 (8GB RAM, Intel i5-2520M processor), with external GPU GeForce GTX 1050 Ti.

	Server Timestamp Value	**Client Timestamp Value**	**Difference (Latency in seconds)**
Test 1	08:48:56:3826	08:48:55:9561	0.4265
Test 2	08:48:59:0157	08:48:58:4017	0.5983
Test 3	08:49:00:3568	08:48:59:5632	0.7936
Test 4	08:49:01:5637	08:49:00:9329	0.6308
Average	**-**	**-**	***0.6310 seconds***

Table 1. Latency Test Results

5. Conclusion

A limited form of testing had to be carried out due to the Covid-19 pandemic. No user-testing could be performed and the performance testing had to be tested on significantly weaker hardware. Despite these limitations, all of the requirements were successfully tested, and only the resolution non-functional requirement (**NFR1**) was not met. However, it is possible that running the performance tests on a machine meeting the minimum specifications would have resulted in all requirements passing.

6. Appendix

6.1 Test Unity Project

The test Unity project used was a 3D scene of UBC's Main Mall. This project lets you move around Main Mall and look at, but not enter, the different buildings. This project was confirmed with EML to be suitable for validation and demo purposes. Refer to figure 4 for a screenshot of the Unity scene.

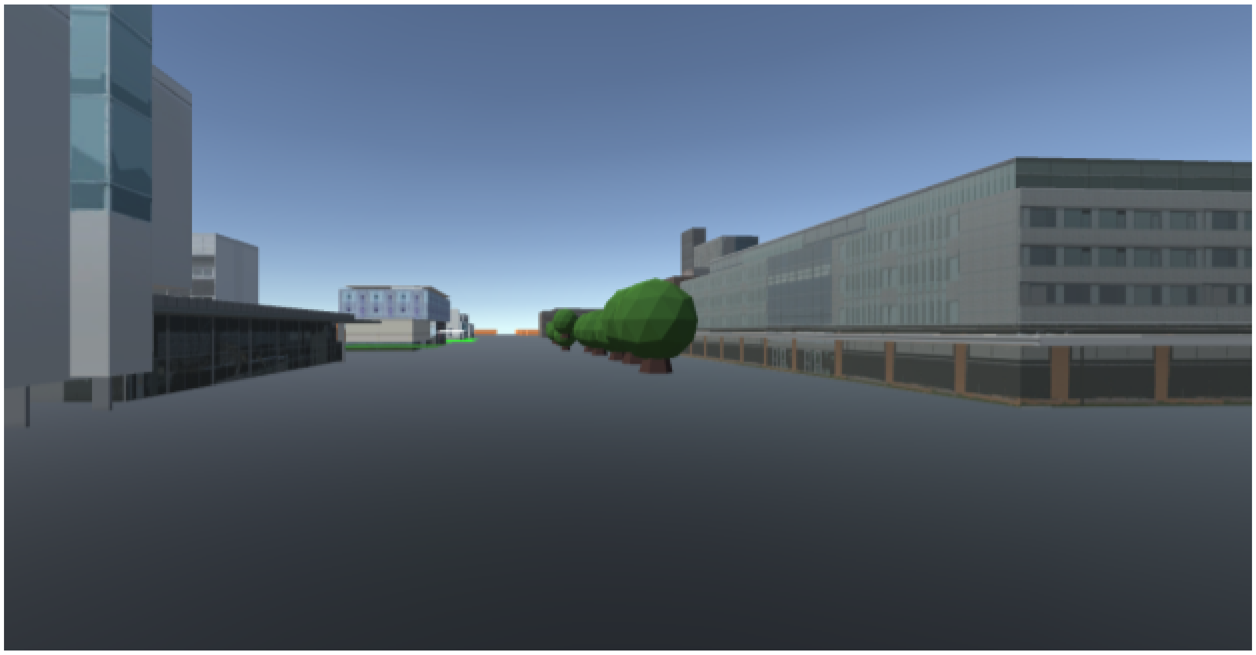
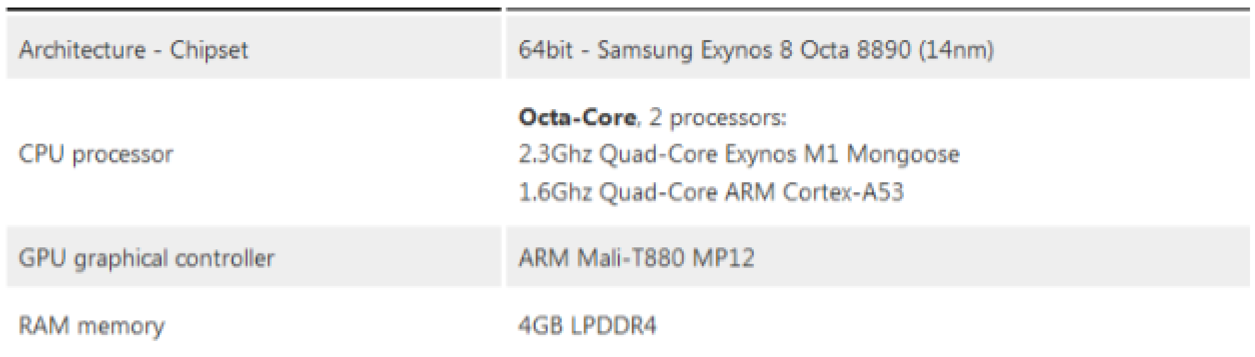


Figure 4. Main Mall Unity Project

In terms of demos, this scene is a good choice because all UBC students and faculty members should be familiar with Main Mall, so it should be immediately clear what is being viewed. In terms of validation, the scene is reasonably complex for a VR demo in terms of size and texture complexity, so it should require a reasonable amount of processing power to render and encode the 360 video. Additionally, because the scene is reasonably complex, the size of encoded video frames will likely be larger than for a simpler scene, which makes this a good scene for testing non-functional requirements that may rely on the networking component.

6.2 Notes on Client Side Testing

Note that there was no non-functional requirement pertaining to the processing power of the client-side devices. However, multiple client-side test devices were used for validation. The device with the least processing power available for testing was a Samsung Galaxy S7 SM-G935W8. The relevant specifications for this device are shown in the table below.



Architecture - Chipset	64bit - Samsung Exynos 8 Octa 8890 (14nm)
CPU processor	**Octa-Core**, 2 processors: 2.3Ghz Quad-Core Exynos M1 Mongoose 1.6Ghz Quad-Core ARM Cortex-A53
GPU graphical controller	ARM Mali-T880 MP12
RAM memory	4GB LPDDR4

Figure 3. Samsung S7 Specifications

The minimum Android version tested was Android version 6.0. It should be noted that this is an old Android version, with the most current version at the time of writing being 10.0. The results on the client-side are not guaranteed if using a weaker device or a lower Android version.

6.3 Testing Environment

Testing was performed in a private home setting. Details on the wifi router that was used are not available. Wifi router was roughly 6-8 meters away from client phones and server device. Client phones and server device were all within 2 meters of each other.