Working Prototype Known Problems Report

Project: Unity Optics

Team Unity
Revision 1 (Last Revision July 17, 2019)
University of California, Santa Cruz

July 22, 2019

1 List of Not Working Functions

Here we list non-working functions (or functions that could not be made to work as intended) for the project. We separate these issues based on their respective module in the project. Hence, we have identified four core modules of our project: the Unity Game Engine, the static website, Firebase, and Travis Continuous Integration.

1.1 Unity

Overall, there were few issues with developing in Unity. The only non-working function is an internal Unity Engine function that builds our project for a specific platform (such as WebGL for our purposes). Our intention was to build for WebGL; however, because of the choice of using Firebase we are only able to build for iOS and Android.

In order to fix this issue we would have to remove Firebase and integrate with a different database service. The difficulty here is finding another service that supports Unity integration.

1.2 Website

The website where we decided to host and showcase a sample client-side API had a key feature that was decidedly depcreated: real-time updates. Due to

time constraints, we were unable to integrate a real-time update of firebase telemetry data from the Unity demo and similarly integrate these real-time updates to the website.

In order to solve this, we would have had to divert development time towards researching the real-time features of Firebase Storage and integrating these. It would require us refactoring and re-integrating this storage technicaly into both Unity and the website; thus, we chose to keep this as a non-working function of our final product.

1.3 Firebase

Known issues for Firebase include a function we are using within the Unity plugin that is meant to upload data to Firebase. This function is specifically meant to transmit advertisement data to Firebase. The issue itself is still undetermined as it does not occur on every run. Sporadically, Firebase will refuse to accept the transmitted data.

1.4 Automated Testing with Travis-CI

Automated Testing via Travis-CI is incomplete. The necessary .yml and bash scripts are included within the project and are technically functional. Travis-CI properly executes the .travis.yml and install.sh scripts; however, it fails to run the unit test Test Runner within Unity via the install.sh script. This is entirely due to using the free Personal version of Unity.

After closer inspection, Unity3D Personal Edition does not allow terminal activation commands to be executed within the Travis-CI virtual machine. What occurs is that in the Travis VM Unity Mac Editor will be properly installed on an OSX image and once ran via command line (which the Personal version does allow for) it asks for a license. This license requires username, password, and serial key. Even when provided with these, the Unity Engine will hang because there exists no way for the Personal Version (which does not include a serial key) to accept the user license in the command line.

The fix for this issue required extensive restructuring and we decided that it was not high enough priority to finish. In order to work around this issue it would require us to switch to an Ubuntu image and install a Virtual Network Computer (VNC) server in this VM. Furthermore, we would have to run a VNC Client alongside this and generate a license file that would then be

passed to the server running Ubuntu Unity. This fix is out of the scope of the work we wanted to accomplish.