# Unity Optics
Automated Testing

Matthew Rhea
Revision 1: 7/18/2019
University of California, Santa Cruz

July 18, 2019

## 1 Testing Methodology

For automated testing, I integrated our project into the Travis Continuous Integration system. This process automatically builds and run the in-built unit tester Test Runner in the Unity 3D engine. Completing this required numerous scripts to be included within our directory: .travis.yml, install.sh, and build.sh.

### 1.1 .travis.yml

This file defines the language, operating system, global environmental variables, and which branches to test on for the Travis virtual machine (VM). Additionally, here we define which bash scripts are ran within the VM.

### 1.2 install.sh

Runs terminal commands with a Travis-CI OSX virtual machine to download and install the Unity3D Mac Editor (version 3.0a8).

### 1.3 build.sh

Within the OSX virtual machine, build.sh sets the options for the Unity Test Runner built-in unit tests and runs them. It stores the results into a .xml file that is then printed to the log in Travis-CI to show the results of the unit tests (either a pass or fail). Furthermore, it attempts to build and run the project for OSX in order to confirm there are no compilation errors.

## 2 Known Issues

As being still a work in progress, there are some known issues with Travis-CI and how it interacts with the Unity3D engine. Namely, without paying for Unity Pro we are unable to run any terminal activation commands for Unity beyond starting the engine via the build.sh script. Because of this, we experience a timeout error and fail the unit tests. However, after inspecting this issue I can confirm that if there existed a serial key of a Unity Pro version then this automated testing process would work.