Augmented Reality Game

Test Plan

COP 4331C, Fall 2015

Team Name: Project Pals

Team Members:

* Eric Peralli - Eric.peralli@gmail.com
* Connor Heckman – Connor.heckman@me.com
* Clayton Cuteri – Cuteri.clayton@knights.ucf.edu

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Who | Comment |
| V0.0 | 10/04/15 | Clayton Cuteri | Original Test Plan Draft |
| V1.0 | 10/07/15 | Clayton Cuteri  Eric Peralli  Connor Heckman | Updated Draft |
| V2.0 | 10/07/15 | Connor Heckman | Added Links to other documents. Reformatted and edited some grammar mistakes. |

**Contents of this Document**

SECTION 1 - Introduction

* Overall Objective for Software Test Activity
* Reference Documents

SECTION 2 - Description of Test Environment

SECTION 3 - Overall Stopping Criteria

SECTION 4 - Description of Individual Test Cases

**SECTION 1 – Introduction**

**Overall Objective for Software Test Activity:**

The overall objective in testing our software is to determine whether it has functionality, user friendly GUI, and networking. We will want to make sure that the software is actually able to work as an AR game. We also plan to ensure that the GUI has easy navigation so that users are able to setup up the game quickly and effectively. The last thing we will test, is the game’s ability to network. For this we will make sure that at least two different devices are able to interact via the game.

**Reference Documents:**

[Concept of Operations](https://docs.google.com/document/d/1x4QyHfMRQdcK_UUwJiVtlEQeu1ca7ca3vmrsmUwGgYY/edit?usp=sharing)

[Project Management Plan](https://docs.google.com/document/d/1_ODvCtTKPSJEnTv0dxVnQx6iGd40UmvQDcgeTsX3Kjo/edit?usp=sharing)

[Software Requirements Specifications](https://docs.google.com/document/d/1bq6qFGaFix2dHopm9GFpDvjj5qe3KKfZXPA6zyiA4v4/edit?usp=sharing)

**SECTION 2:****Description of Test Environment**

**Hardware**

The platform for our application will be any Android supported smartphone. Tests will primarily be conducted on the developers own smartphones, but a random sample of 3 other Android supported smartphones will be selected from the developers’ friends and acquaintances to include a wider variety of devices in the testing phase. The application should show no differences in GUI appearance or game functionality across the smartphones being tested.

**Software**

The game will be designed for Android OS Marshmellow 5.0, though updates may be required if newer versions of the Android OS are released during development. Tests will be conducted using smartphones with the most recent update of the Android OS. The developers will explore the possibility of conducting tests using emulators on their own computers, but only if this can be done without adding a large overhead to the testing phase.

**Testers**

The developers themselves will do preliminary testing. These preliminary tests will ensure the basic functionality and appearance of the game. Once these tests are run with satisfactory results, the developers will expand the test base to include their friends and acquaintances. Each developer will be responsible for having at least 3 friends test play the game. Each developer will receive their friends’ input and improvement suggestions and share them with the team.

**SECTION 3** **- Stopping Criteria**

**If errors are found during testing**

* The developers will record the errors and continue testing the software until a fatal error is encountered that renders the game unplayable. Once the game has been rendered unplayable by a fatal error it will be sent back to development. While the application contains any fatal errors (errors which render the game impossible to play or cause the application to crash) the developers will consider the game “Broken and Flawed”
* When the developers resume testing they will continue to record any new errors they encounter until they reach another fatal error. If there are no longer any fatal errors in the game the developers will consider the game “Functional with Flaws”.
* When the developers have listed all the remaining minor errors in the game, they will send the application back to development. When development has corrected the noted errors, the developers will resume testing and continue to search for new minor errors. When they can no longer find minor errors that directly effect gameplay, (examples of minor error problems with hit registers, inconsistent enemies, incorrect health deductions, etc.) the developers will consider the game “Fully Functional”.
* The last portion of testing will be the process of exploring the aesthetics and enjoy-ability of the gameplay. Developers will examine the game to see if there is any room for improving the user’s experience without producing major overhead. Examples of minor improvements that the developers may consider would be more customization features, better artwork for enemy sprites, or more in depth heads up display. If the developers feel the game is optimally enjoyable and no reasonable improvements can be made to its aesthetics, the developers will consider the game “Functional and Fun”.

**If no errors are found during testing**

* If there are no discernable errors found during one of the testing phases developers will consider it “one pass”.
* In order for the game to move on to the next phase of testing (i.e. consider the game without errors affecting functionality), the game must receive three passes from each developer.
* After the game has received a successful pass, the developer testing the game will exit the application in reopen it, beginning a new instance of the game. This repetition of testing will ensure a through job is done at each phase of testing. The developers want regressing to an earlier testing phase if at all possible.

**What is “good enough to deliver”?**  
In order for the developers to consider the game “good enough to deliver” the game must be pronounced “Fully Functional” at the very least. Obviously, the developers would prefer that the game be considered “Functional and Fun” before delivering it, but it is accepted that there may be some disagreement between developers on whether or not the game has room for improvement at the time of delivery. This may stem from a lack of time to add on an extra, or purely a difference in personal taste. Pressure from other commits or classes may result in the developers delivering the application in the “Functional with Flaws” state. This is undesirable, and indicates a failure at some level of development, but the developers will still consider code that has reached this level in the testing period to be “good enough to deliver”. The developers will not under any circumstances deliver the application while it is considered “Broken and Flawed”. An application in this state is considered by the developers to be unacceptable and not “good enough to deliver”.

**Section 4 - Description of Individual Test Cases**

|  |  |  |  |
| --- | --- | --- | --- |
| Objective | Description | Test Conditions | Expected Results |
| Observe that the game is able to load, run, and play on a device that has no internet connection. | Open the app and observe that the game can start up. Test that they can launch the high scoreboard to display the record high scores of the local user. | User’s mobile device has no internet connection. | The game should respond to commands as normal, and not attempt to display online high scores. |
| Observe that the game is able to load, run, and play on a device that has an internet connection. | Open the app and observe that the game can start up. Test that they can launch the high scoreboard to display the record high scores of online users. | User’s mobile device has internet connection. | The game should respond to commands as normal, as well as retrieve and display online high scores. |
| Test that the application runs in the background properly. | Open the app, minimize it, and then open it again. | None | The game should suspend the state, and resume from said state once the app is launched again. |
| Test that the application exits correctly. | Open the app, minimize it, and end the process | None | The game should have attempted to save local high scores, and push global high scores to the server before exiting. |
| Test the changing the theme of the application responds correctly. | Open the app, select Themes, then select a different theme than the currently selected one. | None | The game should operate as normal, replacing normal textures with textures appropriate to the selected theme. |