

Test Plan Document

1. Introduction

We are creating the Android application Base Run as a project for the Fall 2016 Course: Software Intensive Engineering ECE373 taught by Professor David Irwin. Base Run is a team-based smartphone GPS game where teams compete to capture and defend bases in order to strategically out-score one another in a set location and time limit. The user will choose the following parameters: location, time limit, radius of where bases can be, and a number of bases (up to a specific maximum bound). The purpose of Base Run is to use the smartphone GPS as a way to create a new, fun, and easy game for users to quickly learn and play.

2. Quality Control

2.1 Test Plan Quality

We focused on testing methods presented in class and recommended for the specific test cases we wanted to use. We used tools like GDB and Travis because they were recommended by Professor Irwin to use.

2.2 Adequacy Criterion

The tests are meant to verify the efficacy of the server's functions as well as to detect when a change made to the repo doesn't work and/or breaks existing functions. Our test cases verifies the integration of the client and the server as well as the efficacy of the RPC functions that we have implemented.

3. Test Strategy

User Interface - We used manual black-box testing, meaning we tested the interface based on its requirements by using it on an emulator as well as with various phones and tablets.

Java Client - Also manual black-box testing, but this one uses the required I/O between the client and server using code.

Server - We use white-box testing, meaning we use the code to select the test cases. It is done using a test file in c++ and runs on Travis anytime someone commits to the master branch of the repo or opens a pull request.

4. Test Cases

Provide a table of test cases, one per row. For each test case, the table should provide its purpose, the steps necessary to perform the test, the expected result, the actual result, and any additional information that you think is relevant. Note that all groups must implement their test cases using

Travis, so that they are invoked on each git commit. You may use unit testing frameworks specific to your language or write your own drivers and stubs.