Client Side

Location

Testing Plan

By John Paul Mardelli  
  
Last Updated:  
2/27/2014

[1. Objective](#h.2aq3iiifnibz)

[1.1 Purpose](#h.yqws9oou3hae)

[1.2 Scope](#h.a134w1f0h8fl)

[2. Testing Parameters](#h.bxvggnrjiett)

[2.1 Requirements](#h.imvktuunnqg2)

[2.1.1 Distance Requirements](#h.66fx7sweoo4e)

[2.1.2 Trilateration Requirements](#h.s7fdugoxnaat)

[3. Testing Strategy](#h.tiozhw7j6ewk)

# **1. Objective**

## 1.1 Purpose

This document describes the plan for testing Minimap's client side location tracking software.

This document will accomplish the following objectives:

* Identify existing project information and what specific capabilities of our software are to be tested
* List the testing strategies to be employed and the reasoning as to why they are used
* Establish requirements as to the accuracy of our tracking that we should compare test data against
* List the deliverables elements of the test activities

## 1.2 Scope

This test plan describes the accuracy testing we will be doing on our signal strength/distance estimation algorithm and on our tri-lateration positioning algorithms.  
  
This test plan assumes that all of our code is adequately tested already through unit testing and integration testing.  
  
Because the tri-lateration positioning accuracy relies so heavily on the accuracy established by the signal strength tests, it is necessary to run these two test sequentially. If the first test is not up to the current milestone's standard, then we cannot proceed with the next test until we change our system to meet the first's criteria.  
  
The most critical performance measures to test are:

* The accuracy of our distance measurements
* The accuracy of our positioning algorithms
* The minimization of variance in readings (jitter)
* The accuracy our system while the client device is in motion

# **2. Testing Parameters**

The following is a list of what will be tested under this testing plan. This includes both functional and non-functional requirements, as well as the list of accuracy milestones we are setting for ourselves.

## 2.1 Requirements

### A detailed description of each requirement can be found in the “Client Schedule (Spring Semester) document.

### 2.1.1 Distance Requirements

* Linear smoothing (beacon distance)
* Determine amount of accelerometer movement
* Variable smoothing based on accelerometer
* Future position estimation based on accelerometer
* Basic propagation constant fingerprinting

### 2.1.2 Trilateration Requirements

* Linear smoothing (beacon distance & position)
* Trilateration of three closest beacons
* Least-squares estimation
* Weighted least-squares estimation based on distance
* Determine amount of accelerometer movement
* Variable smoothing based on accelerometer
* Future position estimation based on accelerometer
* Basic propagation constant fingerprinting
* Automatic distance calculation curve calibration
* Weighted least-squares estimation based on jitter amount
* Weighted least-squares estimation based on map environment
* Position correction based on map environment
* Position estimation based on map environment

**2.2 Accuracy Milestones**

**Accuracy Benchmarks for Sprint #1:** Trilateration via Least-Squares Estimation

By March 17

|  |  |  |  |
| --- | --- | --- | --- |
|  | **10m open grid** | **20m open grid** | **40m open grid** |
| **5m precision** | 90% | 80% | 60% |
| **3m precision** | 60% | 50% | 30% |
| **2m precision** | 50% | 30% | 10% |

**Accuracy Benchmarks for Sprint #2:** Accelerometer and Weighted Least-Squares Estimation

By April 7

|  |  |  |  |
| --- | --- | --- | --- |
|  | **10m open grid** | **20m open grid** | **40m open grid** |
| **5m precision** | 90% | 90% | 70% |
| **3m precision** | 70% | 60% | 40% |
| **2m precision** | 60% | 50% | 15% |

**Accuracy Benchmarks for Sprint #3:** Map Environment Contribution

By April 21

|  |  |  |  |
| --- | --- | --- | --- |
|  | **10m obstacle grid** | **20m obstacle grid** | **40m obstacle grid** |
| **5m precision** | 90% | 90% | 70% |
| **3m precision** | 70% | 60% | 40% |
| **2m precision** | 60% | 50% | 15% |

# 3. Testing Strategy

The Test Strategy presents the recommended approaches for the testing of both the signal strength/distance software and the tri-lateration positioning software.   
  
Regardless of Test strategies, these test should all be conducted in the same open environment when available. This environment should be a large, open indoor room, that doesn't have people regularly walking through it and interrupting testing.  
  
**3.1 Distance Testing Strategies**

**3.1.1 Idle Testing Strategies**

The first step to testing our distance algorithms is to make sure the measurements between two static points are accurate. We will position one beacon in the room and measure it at 4 points, spread out evenly between the beacon and the maximum range of it's distance. At each of these points, the tester will record 10 measurements. Post-testing, the data will be analyzed and the accuracy and overall variance will be recorded for each point.  
  
**3.1.2 Moving Testing Strategies**Once we have static accuracy down, we need to make sure the algorithms can also be accurate when the device is mobile. A person carrying the device will walk the length of the devices range and drop a token each time a scan is initiated. The distance between the token will be measures to get the actual distance. This test will require 30 data points to gauge accuracy correctly. Once they are collected, the data points will be analyzed to see how far off each gathered location was off from the actual position  
  
**3.1.3 Stopping/Starting Testing Strategies**The last thing to test for is a combination of the two previous: stopping and starting. This test will make sure that as the device stops moving, the tracking stops accurately with it; same goes for starting movement. The Tester will gather data by starting movement from a static state, gathering 5 data points of movement, before stopping and gathering 5 data points of standing. This will be repeated 5 times for a total of 50 data points.  
  
**3.1 Tri-lateration Testing Strategies**

The testing of the tri-lateration systems will use a combination of reconfiguring environments and device movement to thoroughly test the system for all foreseeable uses of our software. As there's 3 strategy for configuration and 3 for movement, this will lead to 9 sets of data. The quickest way to do this is to do all movement tests for a configuration before setting up a new environment.

### 3.1.1 Environment

#### 3.1.1.1 Open Environment

#### The basic testing ground for our system will be in the environment we've chosen, completely free of any obstacles save for the tester. This allows us to get the absolute best case performance of our system, free of anything that might interfere with the bluetooth signal propagation.

#### 3.1.1.2 Crowded Environment

#### The next step will be to add people to the equation, simulating a crowded room. This will present some difficulty as we can't actually get a large crowd of people on demand whenever we want to test. The best way to attempt this is to do the test in a group of 4, with the tester as close to the middle of the group as possible when grabbing readings 3.1.1.3 ObstacleEnvironment

Finally, we have to account for having physical objects in the room interfering with signal propagation. For this, we'll use folding tables that can easily be moved around and repositioned. The tables should be positioned in such a way that they will be between the beacons and the tester, allowing for any signal interference that might happen. Ideally they should be setup in a lane style like a science fair, so the tester has room to move about  
  
  
**3.1.2 Movement**

#### 3.1.2.1 Standing

#### This test strategy is very similar to the one for distance. We will divide a grid box into 4 further quadrants. We will then stand still in each of the quadrants for 10 data points each. Post Testing, data will be analyzed for accuracy and jitter.

#### 3.1.2.1 Moving

#### The tester will back and forth along a straight line, parallel to one of the axis' of the grid, at a steady pace. This will be repeated for as long as is necessary as to get 15 points of data. The tester will then switch and walk along a line parallel to the other axis for 15 points of data. The data will then be checked for how accurately it matches the corresponding axis of the line walked. 3.1.3.1 Starting/stopping

#### The tester will walk along a line, stopping and starting similarly to the distance method, until they have obtained 50 points of data (5 for each start and stop). This data will then be analyzed to see how accurately the system responds to the user's movement

4. Testing Deliverables

This section will contain the deliverables needed to be completed to finish a test of the system. The deliverables will detail what tests need to be done, as well as allow space for the raw data and tester calculations to be entered.