Testing Results

1. Overview

In order to test the core functionality of our application, we used a combination of techniques. We used the built-in J-Unit framework for automating some simple tasks, such as button presses, scrolling, and navigation. The Android framework was a little too cumbersome for more complicated testing cases, so we switched to a manual debugging approach in order to test more complicated application functionality, such as route generation and behind-the-scenes data management.

In addition, we tested the application on several different devices, including several different virtual devices in the Android emulator and two physical devices (the personal cell phones of Tom and Vinai). We tested the functionality of the application on these different devices, but placed a greater focus on the usability and user experience on them. The devices had different resolutions, OS versions, and input methods (touchscreen, keyboard, both...) in order for us to have a broad sampling size for our tests.

1. Techniques

We used two main methodologies in our testing: J-Unit automation and Manual testing.

Android has special J-Unit test classes for use in the testing of android applications. This method proved useful when automating simple, individual tasks or interactions, such as a screen drag or a check box press. The Android emulator also allowed us to visually inspect the progress of these J-Unit tests on the emulator or physical device we were testing, and to manually verify the progress as the test progressed.

However, those tests were too limited and a little too cumbersome for more detailed testing, so we chose to do manual functional testing on some of our core features, including the path routing algorithm and map rendering. For these tests, we recreated the steps of each functional test case in our functional requirements document and verified that the application’s output coincided with the expected results.

1. Results

For each of the functional tests specified in our functional test documentation, we performed verification of the functionality according to one of the two methodologies described above (J-Unit automation or manual debugging). The final version of the application passes all functional tests specified in the document, but there is one unresolved issue regarding image display on high-resolution displays. The application performs well under environments running Gingerbread and lower, but has some image display issues on certain devices with high resolution displays running on later versions of Android (which supports higher res displays).

For a detailed list of bugs that were discovered during the testing phase, reference the Bug Tracker spreadsheet, which documents the bugs we discovered, where they were located, and by who and when they were fixed.