CS 4720
Final Project Report

Team Members: Alishan Hassan, Zach Showalter, Hawk Haines

Device Name/Platform: Blastoise / Android

App Title: Mission Quest

Project Pitch: Ever wanted to tell the world you're better at getting meaningless quests done than anyone else? Well, now you can bring a scoreboard to the real world and level up with it. Mission Quest, in a nutshell, consists of a list of missions that the user can complete. The user levels up by earning experience points for completing each mission, which amass to a glorious total score that the user can boast to the world. Missions mostly consist of reaching certain, exciting locations. When the user reaches the location, he/she has the option to turn it in and accumulate points. Never has life been so fulfilling.

Why Android?: Our group chose Android because we were more familiar with Java and the Android platform than we were with Swift and Xcode. During the iOS project, some of the intricacies of Xcode's interface caused us a lot of frustration when building our app, and the lack of online documentation available made it difficult to find solutions to our problems. Furthermore, we were all able to develop for Android without being restricted to only using the lab computers in Rice 340 (something we were not able to do with iOS, since none of us own a Mac). This also applied to testing the builds for the project. Two of us owned Android phones, and it was much easier to test our apps with multiple physical devices rather than with simulators.

Key Features:

- Incorporates functionality with the Android GPS.
- Incorporates the accelerometer on the splash screen and in the mission requrements.
- Uses internal storage to track score for the app.
- Incorporates Weather Underground web service.
- POLISH! 

Basic Usage Instructions:
On startup, the user is shown a splash screen which prompts them to "Rock On to Continue." This is a prompt for an accelerometer input, and shaking the device will progress the app to the home page. The home page presents the user with three options: "Available Missions," which leads to the list of available missions to complete, "Completed Missions," which leads to a list of missions that have been accomplished, and a small "About" page that provides a simple description of the app. The home page also displays the current number of points the user has accumulated, as well as the device's current latitude and longitude. The user is able to click on

missions in the “Available Missions,” and if the device is within a specific distance of the designated coordinates for the mission then the mission is added to the “Completed Missions” list and the point value associated with the mission is added to the user’s score.

Special Info for Running the App:

We’ve added more missions that can be completed in Rice to make the app easier to test:

“Rain or Shine, meet THE Professor Sherriff”,
“Olsson is Dark and Scary, Go Somewhere Happy”,
and “Rock out in Rice Auditorium.”

Note: for the mission in the Auditorium we included extra feature that applies bonus points earned based on how much the device is being shaken (the speed at which it is currently moving).

Key/Highlight Features:

Since the last build from the final Android project, we’ve added multiple missions that can be completed within Rice Hall in order to give the TAs an easier time when testing the app. We also added an extra degree of polish to the mission list and to the previously vague toasts that appear when the user tries to complete a mission that they are out of range from, as well as a menu icon for the Android menu. Previously the app had just redirected the user back to the home screen, but the new message displayed allows the user to close the message and return directly to the missions list. We’ve also added a new “Dashboard” activity which provides the user with their location, the weather at the location, and tells them which mission is closest to where they are.

Lessons Learned:

During this final iteration we learned a lot more lessons about the App Development process, especially pertaining to user experience and realizing when ideas for a feature aren’t going to work. For this final build we had wanted to add a separate “completed missions” list, but while building it we realized that it would be a feature that wasn’t really that necessary. A simpler solution was to replace completed missions on the “available missions” list with a line that says “completed”. Likewise, we had wanted to incorporate a QR code scanner in order to incorporate a real-world element to the missions: a user could turn in missions by scanning the QR code that was stuck at the mission location. However, we didn’t consider the cost of the material assets in actually getting permanent QR stickers made (as well as getting them to Everest….), but we also ran into the issue that some Android devices, like our 1st Gen Nexus 7, only have a camera on the front. That issue eliminated QR scanning as a viable testing option.

Recognizing setbacks like this also speaks about how to go about App Development as a whole. Rather than implement the QR scanner, we focused on creating the “Dashboard” activity in order to display more relevant information to the user, as well as addressing polishing issues that were identified by the TAs in grading. While it may sound nice in theory to try to add

a bunch of additional functionality, it is often more important to adapting the design to actual user feedback, even when that involves diverging from what may have been originally intended.

Wireframe: