

CS 4720 - S18 - Final Project Documentation Report

Device Name: Nexus 7

Platform: Android

Name: Benjamin Fuhrman

Computing ID: bdf7kt

Name: Mark Manahan

Computing ID: mmm5ja

App Name: Melee Stats

Project Description:

Our app, Melee Stats, will provide a helpful aid to both competitive and casual gamers who play Super Smash Bros. Melee for the GameCube. The users will be able to record each match in order to obtain sufficient data to use when counter picking their opponent after a loss. Additionally, the app will provide match histories and win percentages that can fuel local rivalries as well allow the user to reflect on their previous results. Soon every player will have Melee Stats as their helping Master Hand.

What we propose to do is create an app that will do the following:

- The system shall allow a user to pick their top three used characters
- The system shall allow a user to pick an app color scheme (i.e light vs. dark)
- The system shall allow a user to assign a GPS location to a specific tournament / friendly setup location
- The system shall allow a user to record a match of Super Smash Bros. Melee into a database with fields like: my character, opponent's character, stage, win / loss, opponent's tag, location, date
- The system shall allow a user to receive a recommendation of which character to use and what map to play on versus either a player, character, or combination of the two
- The system shall allow a user to receive their win percentage based of match filters
- The system shall allow a user to take a picture of their opponent for easier searching in case of a forgotten username / tag
- The system shall allow a user to remove an entry from the database

We plan to incorporate the following features:

- 20 pts - Data storage using Core Data (iOS) or SQLite (Android)
 - A user can store a match into a SQLite database which will work as the backend of our system
- 10 pts - Data storage using key/value pair storage (SharedPreferences or UserDefaults)

- A user will be able to store their desired color scheme and their most used characters into shared preferences
- 15 pts - Camera
 - A user will be able to take a profile picture of their opponent for aesthetics and easier searching in case of a forgotten name
- 15 pts - GPS / Location-awareness (includes using Google or Apple Maps)
 - A user will be able to set their GPS location as a specific tournament / house with a range, for greater data analysis

Wireframe Description:

The wireframe for Melee Stats exhibits the design and flow scheme we envisioned for the application. At launch, users are presented with the opportunity to counter-pick opponents, review past matches, add a new match, and make use of the shared preferences function of our application. The settings/shared preferences screen allows users to store their most played characters as well as choose a desired color scheme. Adding a new match prompts the user to enter the character they used, their opponent and their character, and the location of the match. After adding preliminary information, users are prompted to take a picture of their opponent, if they are not already stored in the database, which is followed by map and result selection for the match. Entering the counter-pick screen allows users to enter opponent and opponent character information which returns win/loss analytics against the opponent with their selected character on certain stages. Reviewing past matches prompts the users to enter match information similar to adding a new match which is followed by noting a date range which returns win/loss analytics against the opponent, including match results with certain characters and stages as well as overall win/loss ratio. Examining a specific match displays all information in large match list with more information not displayed in the list, including the picture of the opponent.

Platform Justification - What are the benefits to the platform you chose?

The userbase of the Android platform is larger and more diverse than that of iOS in areas like population count and income levels; we believe that this is a benefit because this would allow the application to reach a larger audience. Analytics have shown that there are more application downloads on the Android platform compared to that of iOS; we believe that this would be an advantage because the greater amount of traffic would be beneficial to us as developers. The publishing process of applications is more lenient on the Android platform than on iOS; we believe that this was a benefit in that we would be able to push our application to audiences easier.

Major Features/Screens - Include short descriptions of each (at least 3 of these)

The "Settings" screen serves as the medium for the shared preferences functionality of our application. In this screen, users are able to select their top three most played characters, which

are used as defaults in the “Add Game” screens, as well as select their preferred color scheme for the application; the “light” color scheme sports light-blue accents with a white background while the “dark” color scheme sports dark-blue accents with a black background. Data chosen for each field is stored in a global shared preferences file which is accessed in other application functions.

The “Add Game” screens are a series of activities that allow a user to store a match in the application’s database. The first screen prompts the user for the character they used, the opponent they played against, the character that the opponent used, and the location of the match (GPS coordinates for the location of the match are taken to be the current GPS coordinates as of adding the match at the location in question; in the case of future match additions, the location field is auto-filled with the closest recorded location in the database in relation to the current GPS coordinates). In the case where the user has not played against the opponent, the following screen will be an activity that allows a user to take a profile picture of the opponent to save to the application database. Following the initial add screen and/or camera screen, users are prompted to select the map they played on for the match. Matches are able to be marked as resulted in a win or loss and tournament or friendly. Should users wish to store multiple matches at once, they can check “multiple” and continue to add as many matches and their results as they want. All information entered in these “add” screens are saved to a SQLite database specific to the application.

Choosing to counter-pick an opponent prompts a user to specify the opponent and the character they used. The following counter-pick screen references the database for the opponent and the specific character to bring up character recommendations on certain stages based on positive win rates for that character against the opponent with their specified character on that stage.

Reviewing past matches prompts users to enter search criteria for the matches they want to review, character used, opponent and their character, as well as location. The following review screen prompts users for a date range to search for and whether to filter by tournament matches only. The result of the search pulls up a complete list of matches that reflect the search criteria, each match listed with the characters fought and the stage as well as the match result, and overall win rate among the matches in the list. Selecting a specific match brings up more information of the match not displayed in the larger list: location name, GPS coordinates for location, opponent and their profile picture, and tournament/friendly status.

Optional Features - Include specific directions on how to test/demo each feature and declare the exact set that adds up to ~60 pts

- 20 pts - Data storage using Core Data (iOS) or SQLite (Android)
 - *A user can store a match into a SQLite database which will work as the backend of our system*
1. Tap floating action button marked with “plus” on home screen

2. Choose character played in Spinner of all characters, type in opponent name, choose opponent character played in Spinner of all characters, type in location name (→ tap “NEXT”)
 3. If the opponent entered has not yet been played against, the camera activity follows; tap “take picture” and take a picture of the opponent (→ tap “OK” if satisfied with picture taken)
 4. Following the initial add screen and/or camera activity, select the map that the match was played on and the match result; check if the match was a friendly or tournament match (→ tap “ADD” when ready to add)
(if adding multiple matches, check “multiple” and continue to add as if singular matches; when adding the last match, uncheck “multiple” and tap “ADD” to conclude adding)
- 10 pts - Data storage using key/value pair storage (SharedPreferences or UserDefaults)
 - *A user will be able to store their desired color scheme and their most used characters into shared preferences*
 - 1. Tap settings button in top-right corner of home screen, select settings
 - 2. Select a character for each of the three character spinners to save as character defaults for the “Add Game” screen series
 - 3. Toggle the color scheme toggle button until desired color scheme is selected (→ Tap “OK” when satisfied with selections)
 - 15 pts - Camera
 - *A user will be able to take a profile picture of their opponent for aesthetics and easier searching in case of a forgotten name*
 - 1. See “Data storage” section to see how to make use of the camera in our application
 - 2. In order to view picture taken for an opponent, begin by tapping “Review” button on home screen
 - 3. Enter relevant search criteria for the match(es) to be reviewed
 - 4. Enter relevant data ranges and check whether to filter only tournament matches or not, tap “SEARCH MATCHES” when satisfied with selections
 - 5. Tap on a specific match to pull up its details; the picture taken of the user will be displayed here, the path of which is referenced from the database and accessed through this activity
 - 15 pts - GPS / Location-awareness (includes using Google or Apple Maps)
 - *A user will be able to set their GPS location as a specific tournament / house with a range, for greater data analysis*
 - 1. See “Data storage” section to see how to specify a location in our application
 - 2. In order to view the GPS coordinates for the location entered at the time of adding a match, begin by tapping “Review” button on home screen
 - 3. Enter relevant search criteria for the match(es) to be reviewed

4. Enter relevant data ranges and check whether to filter only tournament matches or not, tap “SEARCH MATCHES” when satisfied with selections
5. Tap on a specific match to pull up its details; the GPS coordinates for the location will be the coordinates that were pulled when the specific match was initially added to the database at the location in question

Testing Methodologies - What did you do to test the app?

We followed an incremental approach for developing/testing our application. We would slowly develop pieces of functionality and then run the application at several unfinished points, supplying inputs to the application that both fit and did not fit the functionality that was being tested. In the case where an error occurred, we would investigate error messages and research hotfixes to the errors to apply to our code, repeating the step(s) that caused the error and proceeding with further incremental development. At the conclusion of functionality development, we would run the application and supply inputs to the functionality in all of its stages to make sure that it ran properly.

Usage - Include any special info we need to run the app (username/passwords, etc.)

If running a demo of the application on an Android device that does not come with a GPS tracker (e.g. Nexus 7), make sure to connect the device to Wi-Fi in order to allow the device to retrieve GPS coordinates through Wi-Fi.

Lessons Learned - What did you learn about mobile development through this process?

We learned a lot about the technical aspects of mobile development through the making of this application. We learned how applications are compartmentalized into separate functionalities and “screens” in order to provide a unified user experience. We learned how to send, manipulate, and make use of data in the mobile application environment. We learned how to implement mobile device sensors and retrieve their data in order to provide some sort of application functionality.

Separate from the more technical aspects of mobile development, we also learned the importance of user experience and its role in the functionality of a mobile application. We learned that a mobile application should be polished, easy to follow, and user-friendly.