**SELP documentation** **Jaka Mohorko**
SELP Coursework Part 3
Lecturer: Due Date: December 13th, 2017
Stephen Gilmore Last Update: December 8th, 2017

# 1 Core algorithms and data structures

## 1.1 Download data parsing

All downloads were handled using the Android Asynchronous Http Client Loopj.

### 1.1.1 Song list

The song list was downloaded and parsed using a XmlPullParser. The data parsed was stored into objects of type Song containing data on the song title, artist, number and Youtube link. A list of Song objects was created storing all the song objects.

### 1.1.2 Map data

The kml map data of a specific song is downloaded and parsed using a XmlPullParser whenever the corresponding song is chosen to be played. Data on marker styles and their coordinates is extracted to later be manually placed onto the Google Map

### 1.1.3 Lyric data

The Lyrics of a song are downloaded and parsed again using a XmlPullParser whenever the corresponding song is chosen to be played. Songs are stored into an array list where each entry represents 1 line of the lyrics split on the "space" character.

## 1.2 Persistent storage of data

Across play sessions, due to the inclusion of Statistics and Achievements, data must be stored on the device.
All of the data is stored in several text files, which are parsed when needed. All of the modification of the data is done within the onDestroy() function of the main gameplay activity (MapsActivity) and is stored within the files data.txt, solved.txt and solvedbydifficulty.txt.

### 1.2.1 data.txt

Inside the data.txt file, information about gameplay statistics is stored.
Data stored: Total distance walked, time spent, number of puzzles solved, fastest times per difficulty, average times per difficulty, number of placemarks collected.

### 1.2.2 solved.txt

The file solved.txt contains data on songs that have already been solved.
When a new song is solved, it is appended to the file and the number of total songs solved is updated.
When a song that has already been solved at a different difficulty is solved, if the time taken to solve it is lower than the one before, the values are adjusted. Same applies for difficulties - if the highest difficulty at which it has been solved before is lower than the current difficulty, the

value is updated.
Data stored per song: Title, Artist, Link, hardest difficulty at which it has been solved, fastest time for it to be solved.

#### 1.2.3 solvedbydifficulty.txt

Inside the solvedbydifficulty.txt file, information on which songs have been completed for each difficulty is stored. The songs are labeled with the numbers associated to them.

### 1.3 Distance calculation

During gameplay, both distance walked and distance from placemarks must be tracked. As such, on each call of onLocationChanged(), both the distance traveled and the distance from each placemark is calculated using the Haversine formula.
If the user is within 25 meters of any placemark, the word it contains is automatically collected.

### 1.4 Song randomization and choice

When the user selects a difficulty they want to play a game at, a random song amongst the songs within the database is chosen. If that song has already been completed at the chosen difficulty, the app iterates through the song database, starting from the song chosen at random, until it finds a yet unsolved song.
If all of the songs at the chosen difficulty are already completed, the user is requested to pick a different difficulty to play at.

### 1.5 Song guessing

There is no specific algorithm in place for song guessing. The app checks if the guess is correct by matching the string containing the title of the song against the inputted string.
The string matching is not case-sensitive.

### 1.6 Camera tracking

Camera tracking is implemented by moving the map camera to the users location every time their location changes (onLocationChanged() is called).

## 2 Additional features not mentioned in my design document

### 2.1 Camera tracking

Within the main gameplay activity (MapsActivity) the user is presented with the Enable/Disable Camera Tracking buttons, which when enabled, keeps the users location in focus by panning the camera to the user whenever they move. Upon song completion, the user can, within the Song List, view their best time at which they solved that specific song, in addition to what the highest difficulty is that they solved the song at.

### 2.2 Youtube redirects for completed songs

Within the Song List view, the user can press on each of the songs listed there (already completed songs) to open a dialog window asking them whether they'd like to listen to the song on Youtube. If so, the corresponding video is opened in the Youtube app if installed, otherwise the link is opened in their browser.

### 2.3 Information displayed on song completion

After successfully completing a song puzzle, the user is shown data on how long it took them, how far they walked during gameplay nad how many placemarks they collected.

### 2.4 Return to menu confirmation

As the "continue game" function is not present within my application, the user loses gameplay progress data when starting a new game, or returning to the main menu from the gameplay view.
As such, before the user is allowed to do so, they must confirm their choice to exit the current play session.

### 2.5 Additional high score statistics

## 3 Features mentioned in my design document which are not implemented

### 3.1 Continue game

After the user exits a gameplay session, the session data is not saved and they cannot resume from where they left off at the time.

## 4 Acknowledgements


### 4.1 Android Asynchronous Http Client Loopj

Used the Loopj library to download of the song and map data.

### 4.2 MaterialDrawer by mikepenz

Used the MaterialDrawer library for the creation of the gameplay activity navigation drawer.

### 4.3 ListView tutorial on raywenderlich.com

Used the code snippets provided by the tutorial to implement the list views for the Achievements, Song List and Statistics activities.
Link to tutorial: https://www.raywenderlich.com/124438/android-listview-tutorial