Grabble

The Location-Based Scrabble Game

1. Content Overview

The Grabble was inspired by the games *Pokemon GO* and *Scrabble*. The main objective of the game is to collect the letters scattered around the George Square campus, and use the letters collected to spell seven-letter words, and earn points during the process.

2. Features

In addition to the basic gameplay, the Grabble will have following features:

❙ Point System

Each letter used in spelling have a point value, where common letters have low points, and rare letters have a higher level of points (using scrabble rule, A and E worth 1 point, Z and Q worth 10).

❙ Item System

During the gameplay, the player may acquire items with special effect. For example, some of the item allows player to collect letter from a far distance; some allows player to have an increased field of view to look around for desired letters.
These items are given through various methods, there's a 50% chance to get a random item after player spelled a word with 18 or higher point.

❙ Level System

When player accumulates enough points, they would level up, which gives them a larger "grab radius" or wider zoom level. Also, this awards player items.

❙ Achievement System

The game records the statistics of the player, achievements are given to player when certain requirements are reached. This provides a goal for the player, and award player with points and items upon complete.

3. Design Preview

Currently, the Grabble haven't go through the visual design part, the following app preview only shows the basic function of the application.

- Main menu and map:

- Scrabble and Achievement:

4. Potential Difficulties

Most part of the application should run smoothly, except for the initialization of the maps and the dictionary check in the scrabble part.
For the map part, there will be hundreds of markers on it, so initializing the map may took a long time. Furthermore, it may have poor performance when the user tries to move around the map. One way to improve is to add a field of view, by only showing letter that are in certain radius around the player, and refresh it on an interval of 1 minute (depend on the settings of player).
For the dictionary part, there are 23869 words in the dictionary used, searching one word in it might took a noticeable time for the user. Not to mention impact on implementing further features like penalty for repeat word use. It is still undecided what kind of optimization might be used.

5. Hardware Specifications

Current specifications:

Target SDK Version: 23
Minimal SDK Version: 16
Target Android Version: Android 6.0
Minimal Android Version: Android: 4.1
Required Google Play Service Version: 8.4.0
Target Screen Resolution: 480x800
Target Screen Density: hdpi

I selected SDK version 16-23 due to these version has the largest amount of market share, and to balance between code availability and backward compatibility. The Google Player Service version is not yet decided, further tests on lowest possible version are needed to acquire more information. The screen resolution and density I chose were low on purpose, because these might have better compatibility on larger screens than smaller ones. Again, these specifications need more tests to be decided.