GRABBLE

The Location-Based Scrabble Game

Design Document

Overview

Grabble is a scrabble game inspired by *Pokémon GO*, allows player to explorer around George Square, gather letters and use it to make 7-letter words.

How the Grabble will be played

The primary experience is to gather letters and compose words. As the player walk around George Square, they only need to click on the letter to gather them. The letter collect design allows player to collect letter one-handed, and the process need to be smooth so the player won't need to stop to play the game.
For the Scrabble part, the player just need to type on the integrated keyboard, when he finish the word, he will press "Complete" to transfer the letter he used to score, and award the player with item that helps game progress. Same as above, the typing part cannot lag to ruin the player's experience.
Example: The player is a University of Edinburgh student, he have almost no time for playing games as he timetable is filled with lectures. The Grabble game allows him to relax while walking around George Square for lectures, and assemble words while waiting the lecture to be started.

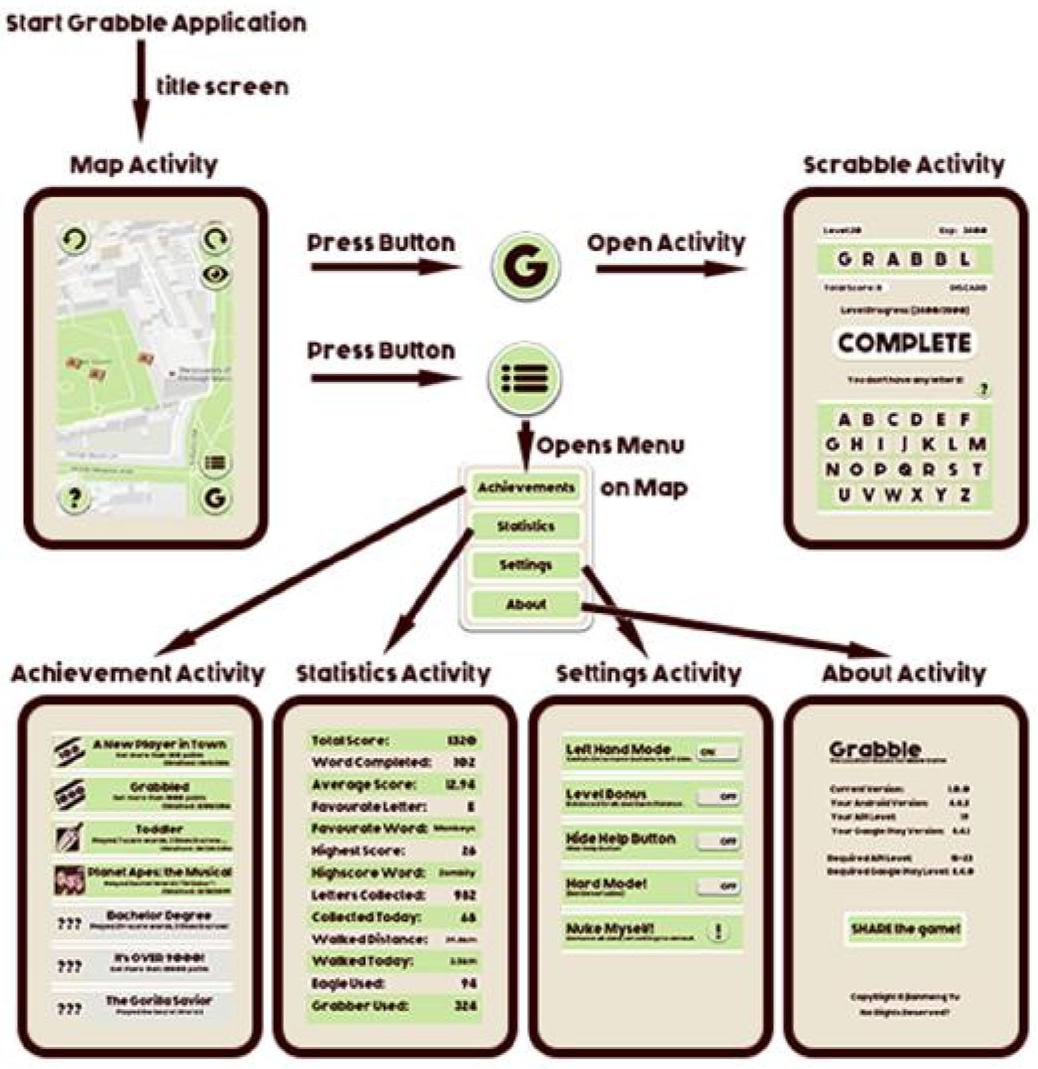
Features

1) Item System
To make the game process easier, I introduced two items to speed up the collection process. Firstly, the "Eagle Eye", an item allowing player to zoom out briefly, to find out if there's any far-away letters the player want. Secondly, the "Grabber", an item that allows player to catch any letter he can see, regardless the distance, so the player won't need to trespass to play the game.
These items will be awarded during the gameplay, by completing hard words, complete achievements, picking up letters also gives a chance to earn items.
2) Level System
The level system exists to give the game a sense of progress, when the player earns enough scores, they level up to have a greater range of collecting letters. They also earn items and achievement doing so. If the player think this feature breaks game balance, they can disable it in the settings.
3) Achievement System
The achievement system give the player a target to play. Upon completion, the game pops a toast to notice player and giving them awards. The player can also check the achievements in the corresponding menu, recording player's progress as a milestone.

4) Statistics
The simply records player’s game profile, it records player’s distance walked, words completed. It also track if the player prefer some words or letters.
5) Left/Right Hand mode
As mentioned before, the letter collect part can be played one-handed, because all the necessary button is on the right hand side. The player can switch it to move buttons to another side for their own preferences
6) Hard Mode.
This mode provides a challenge for player, if they wanted to. This mode replace letters on map with a question mark, stops player from spelling low score words, and it bans several letter randomly when player tries to complete words. The game will prompt player a warning when player tries to switch to it.

Grabble Application Flowchart

The application flowchart is relatively small for easier use.



The four menu activities may be merged into one activity using ViewPager.

Software Architecture

Dependencies

The Grabble game uses following dependencies:

- 'com.android.support:support-v4:23.4.0'
- 'com.android.support:recyclerview-v7:23.4.0'
- 'com.google.android.gms:play-services:8.4.0'
- 'com.google.android.gms:play-services-location:8.4.0'

Every activity of the application uses support.v4 to implement, except for the achievement activity, which uses a recycler view to hold achievements. The map activity uses play service 8.4.0, to reduce the size of dex file, while supporting device with earlier versions of play services. Further details about imported libraries can be found at the end of the document (as it’s a really long list). More dependencies may be used to optimize the application design.

Data Structure

The Grabble uses 3 SQLite database to store required data.

1) Markers

The markers on the map will be stored with the schema:
UUID, LETTER, LATITUDE, LONGITUDE, PICKED
The content will be dropped and updated daily using the downloaded kml files, and will be updated in background immediately after a letter is picked.

2) Dictionary

For the storage of the Grabble Words, using following schema:
UUID, WORD, USED
This will be constructed on the first run, and will update only when finish a word, incrementing the “USED” field by one.

3) Achievements

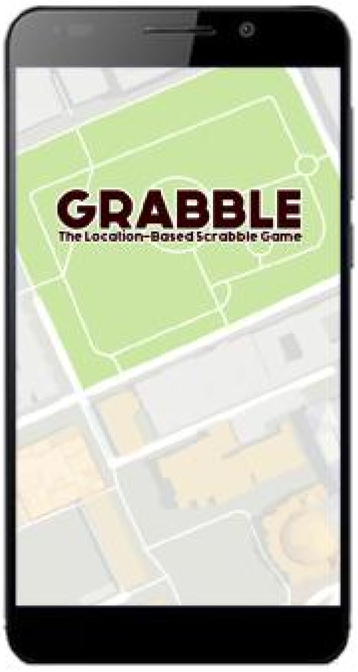
For storing the completion status for achievements, using following schema:
UUID, NAME, DESCRIPTION, HINT, IMAGE, TIME
The achievements are usually updated when certain requirements are met, then a method of achievement class will be called to update the achievements.
Other data like grabble status, letter and item storage, statistics will be stored using SharedPreferences (may be changed to store in JSON file)

Current Activity List

Currently, the application have the following activities:
MapActivity, SingleFragmentActivity, and MainActivity
With following activities extending the Single Fragment Activity
ActivityScrabble, ActivityAchievement, ActivityStatistics, ActivitySettings, and ActivityAbout
The Main Activity is currently used for displaying the intro screen only, and may be deleted on further develop.

Activity Design

All of the activities except map activity only have an empty frame layout, the following design are from the fragment of each activity.

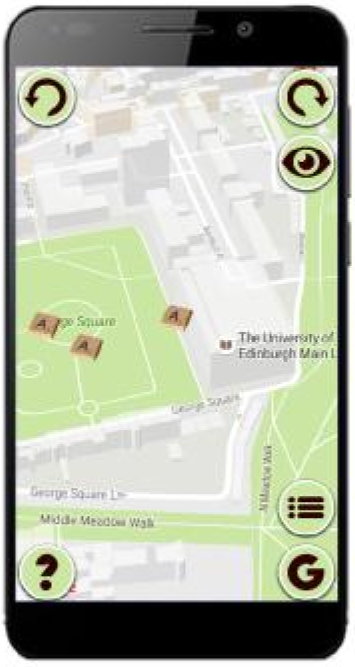


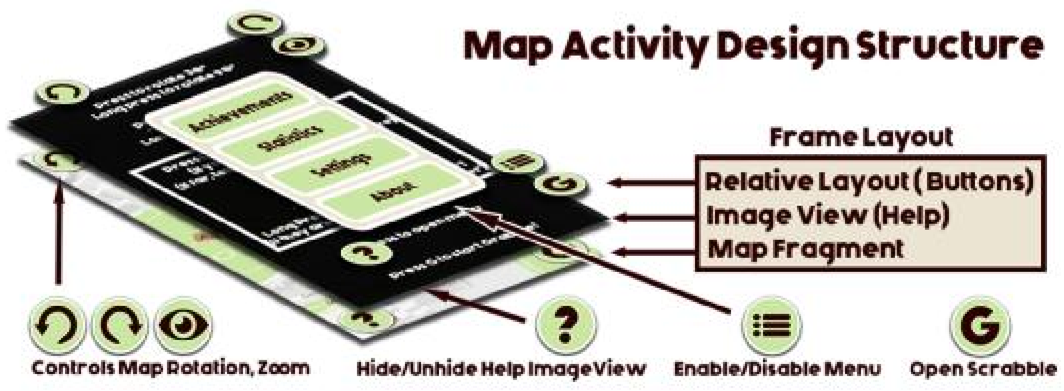
Main Activity

The starting screen, it opens the map activity shortly after initialized, and closes itself. It only contains an ImageView of the game title.

Map Activity

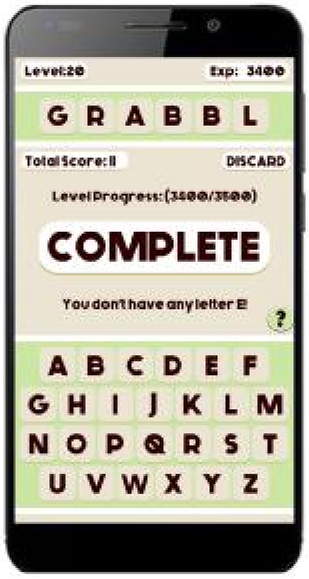
The Map Activity are the activity the player starts, they collect letter on the map, and it also have buttons to open all the other activities, the basic control, xml structure and a sample screenshot from emulator are in the images below.

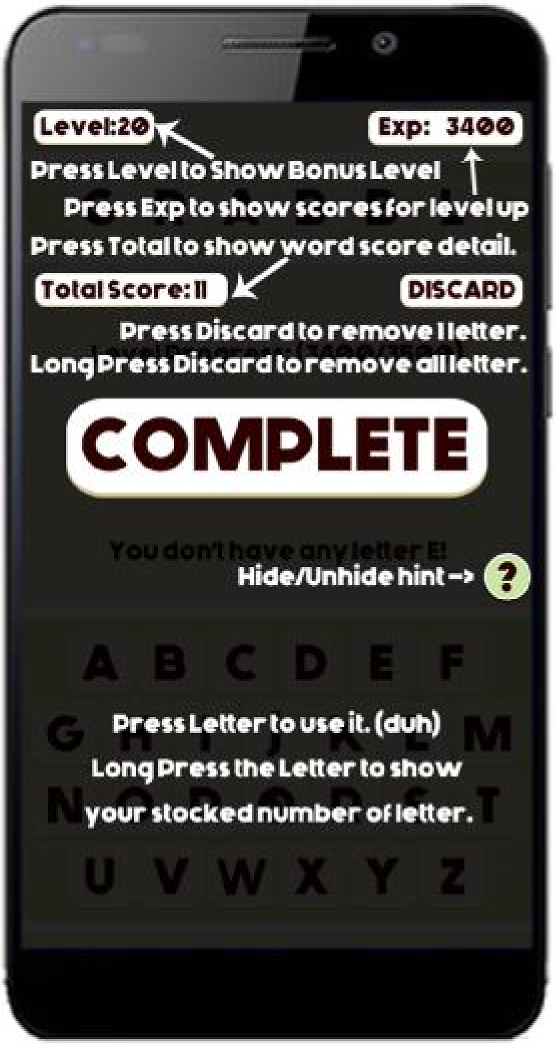




Scrabble Activity

The Activity for players to complete their 7-letter words. The controls are shown in the help layout, check the screenshot for detailed button use. The finished help screen will be different, as it will show how many letters the player have.





Press Level to Show Bonus Level

Press Exp to show scores for level up

Press Total to show word score detail.

Press Discard to remove 1 letter.

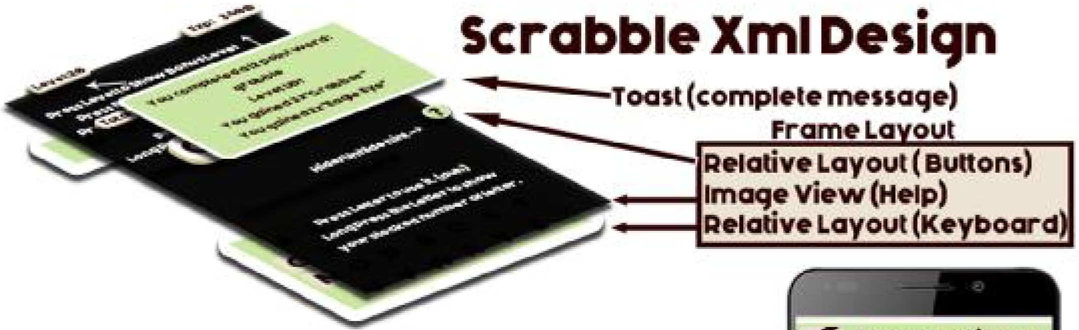
Long Press Discard to remove all letter.

You don't have any letter E!

Hide/Unhide hint ->

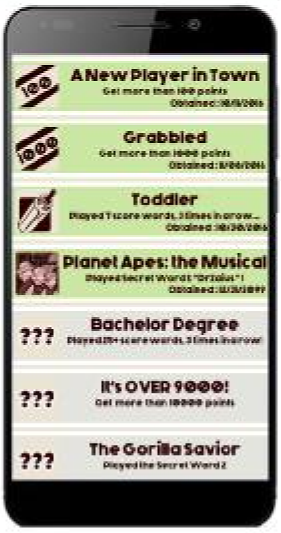
Press Letter to use it. (duh)

Long Press the Letter to show your stocked number of letter.



Achievement Activity

The Achievement activity's fragment uses recycler view, as each relative view it holds stands for one achievement. Unlocked achievements will have green background, having "completion time" and its hint will be replaced with the achievement description.
Player can only scroll through the achievement, as they would be unlocked at other activities.


A New Player in Town
Get more than 100 points
Obtained: [illegible]

Grabbled
Get more than 1000 points
Obtained: [illegible]

Toddler
Played 7 score words, 3 times in a row...
Obtained: [illegible]

Planet Apes: the Musical
Played Secret Word: "Ortolus" 1
Obtained: [illegible]

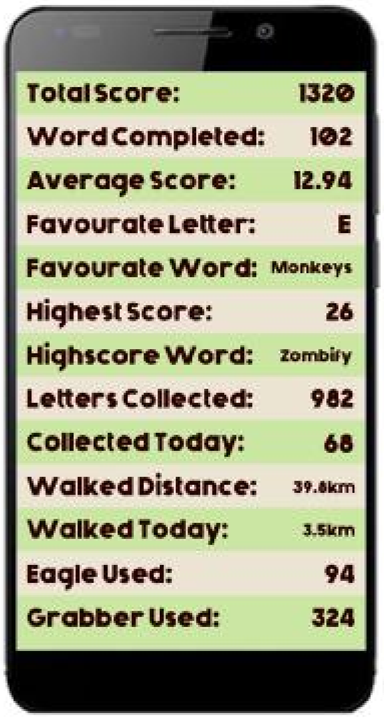
??? Bachelor Degree
Played 20+ score words, 3 times in a row!

??? It's OVER 9000!
Get more than 10000 points

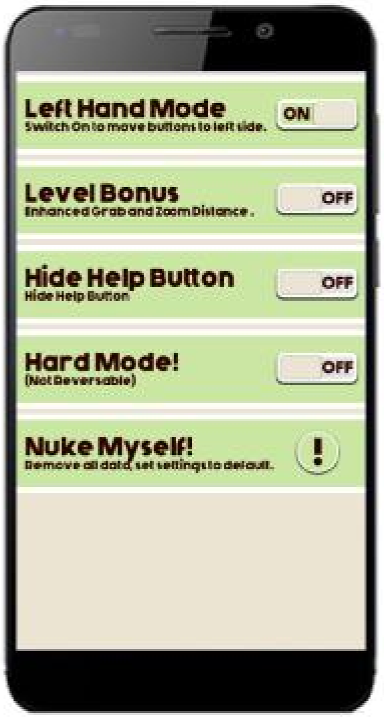
??? The Gorilla Savior
Played the Secret Word 2

Statistics Activity

This activity have a linear view filled with relative layouts. Each relative layout is a line of the screenshot, same as Achievement activity, this activity only provides information, player do not have controls on it.



Total Score:	1320
Word Completed:	102
Average Score:	12.94
Favourate Letter:	E
Favourate Word:	Monkeys
Highest Score:	26
Highscore Word:	Zombify
Letters Collected:	982
Collected Today:	68
Walked Distance:	39.8km
Walked Today:	3.5km
Eagle Used:	94
Grabber Used:	324



Left Hand Mode
Switch On to move buttons to left side.
ON

Level Bonus
Enhanced Grab and Zoom Distance .
OFF

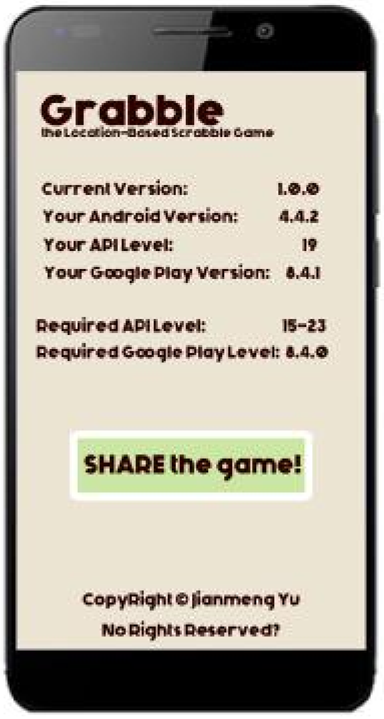
Hide Help Button
Hide Help Button
OFF

Hard Mode!
(Not Reversable)
OFF

Nuke Myself!
Remove all data, set settings to default.
!

Settings Activity

This is the activity player changes settings, it uses same layout as statistics, with switch/button on the right hand side.
Player could toggle hard mode here, and it also allows player to clean their play data, both options will prompt to warn player when pressed. The players need to confirm they want to do so.



Grabble
the Location-Based Scrabble Game

Current Version:	1.0.0
Your Android Version:	4.4.2
Your API Level:	19
Your Google Play Version:	8.4.1
Required API Level:	15-23
Required Google Play Level:	8.4.0

SHARE the game!

CopyRight © Jianmeng Yu
No Rights Reserved?

About Activity

This is simply the activity provides game information such as software requirement and versions.
There is one single button “Share” which sent out an intent to other applications such as text so the player can tell others to get the game, if they want to.

Library used in the activity:

android.content.Intent;
android.content.ContentValues;
android.content.Context;
android.content.SharedPreferences;
android.database.Cursor;
android.database.CursorWrapper;
android.database.sqlite.SQLiteDatabase;
android.database.sqlite.SQLiteOpenHelper;
android.support.v4.app.Fragment;
android.support.v4.app.FragmentManager;
android.support.v4.app.FragmentStatePagerAdapter;
android.support.v4.view.ViewPager;
android.support.v7.widget.LinearLayoutManager;
android.support.v7.widget.RecyclerView;
android.os.Bundle;
android.os.Handler;
android.view.KeyEvent;
android.view.LayoutInflater;
android.view.View;
android.view.ViewGroup;
android.widget.TextView;
android.widget.ImageView;
android.widget.Button
android.widget.Toast;

com.google.android.gms.common.ConnectionResult;
com.google.android.gms.common.api.Api;
com.google.android.gms.common.api.GoogleApiClient;
com.google.android.gms.location.FusedLocationProviderApi;
com.google.android.gms.location.LocationServices;
com.google.android.gms.maps.CameraUpdateFactory;
com.google.android.gms.maps.GoogleMap;
com.google.android.gms.maps.OnMapReadyCallback;
com.google.android.gms.maps.SupportMapFragment;
com.google.android.gms.maps.model.BitmapDescriptorFactory;
com.google.android.gms.maps.model.CameraPosition;
com.google.android.gms.maps.model.LatLng;
com.google.android.gms.maps.model.Marker;
com.google.android.gms.maps.model.MarkerOptions;

java.io.BufferedReader;
java.io.IOException;
java.io.InputStream;
java.io.InputStreamReader;
java.util.ArrayList;
java.util.List;
java.util.UUID;

Document Design by:
Jianmeng Yu
s1413557