

Implementation Document

Algorithms and Data Structures

Download and parsing the daily Grabble letter map:

The download process starts at the onMapReady. When it is called, the game first check the String of weekDay stored in SharedPreferences, if it is not stored (first time playing), the value of it will be set to “noDay”, then the readKml() function is called.

In readKml(), the current day is first calculated and compared with the current stored weekday, and also checks the internet connection. Then it splits to the following cases:

- 1) No Internet: The game will pop a toast to warn player that they don't have internet connectivity, and it will only load letter if there are previously loaded letter.
- 2) First Time Running: The game will tell player to wait for letters to download, then the async DownloadTask is called, while setting the weekday to current day.
- 3) Change of Day: It's basically same as first time running, but the database for markers will be dropped before new letters are added.

Inside the DownloadTask, it will check the connection with the provided URL, if the resources are moved, it replace the URL and continues the check. It will only proceed if the connection have respond code of 200. When available, it downloads the required kml files to external storage folder [1]. After downloaded, it will call the MarkerLab's [2] static get method within the DownloadTask, to parse the kml to database records.

In the MarkerLab, if get method cannot find a static MarkerLab, it will call the constructor to read today's kml file and parsing it as many Strings of data, it takes out the description, name, and coordinates (when it find tag like: <name> in each line), and put these into the database with following schema:

UUID, Name, Description, Lat, Lng, Collected.

After get MarkerLab is called, the async part ends and it sends a message to a handler in MapActivity to add the markers from the database to the map.

Word Lookup in dictionary

The letters entered is stored as a 7-character array, with a pointer integer, it's usually casted to a string before looking up in the dictionary.

The look up of the word is moved to FragmentScrabble, and no longer use a database to store the words. The game uses **binary search** in method isWord() to find the existence of a word in the dictionary.

The binary search start with 2 indexes: start and end with values of 0 and 23868, it calculates the half and search the specific line in the dictionary. If it matches, it returns a true, it decrease the end to (half – 1) when the half have a higher ascii value than the entered word, and increase the start to (half+1) otherwise. Then, it start again from the calculation of half. If there are no words match until start exceeds end, it returns a false. The non-dictionary word used in Achievement system are compared directly.

[1] The file download code is from: <http://stackoverflow.com/questions/3028306/download-a-file-with-android-and-showing-the-progress-in-a-progressdialog>

[2] The MarkerLab structure are from: <<Android Programming: The Big Nerd Ranch Guide>> 2nd Edition by Bill Phillips, Chris Stewart, Brian Hardy and Kristin Marsicano.

Detection of the collectible letters

The MapActivity have an `isInRange(LatLng, LatLng, double)` method, it calculates the euclidean distance between 2 latlng, and check if the distance is smaller than the double. When a letter is clicked, the game calculates the distance between the marker and actual location of player, compared it with a specified grab distance (increases when level up). And it only collects a letter when in range.

Another case is the “long click map to use a grabber”, the game checks all the letters in a “for loop”, to find out which one player long clicks using `isInRange`.

Achievements

Achievements are stored in a database with schema:

UUID, name, description, hint, unlockTime

The achievement data is read from a csv file in asset folder, the UUID is generated using `randomUUID()`, and unlockTime is a string with value “No Unlocked Yet”.

The game tracks the unlock of achievement in MapActivity and FragmentScrabble, when it satisfy the condition, it passes the name of the achievement into the AchievementLab, setting the unlockTime to current time, and makes a Toast to notify the player the achievement is unlocked.

Statistics

All of the statistics are stored in SharedPreferences, and is updated when changed.

Features changed/removed

Item System: Items are now only given when complete words, upon word completion, two dice number with range (1-20) are generated, player will have a greater chance to get items when completing higher score words.

Score System: Now it uses required score table.

Level System: Level up adds a letter collect distance of 1% for each level, player can disable this change in settings menu.

Achievement/Statistics System: Unchanged, doesn't gives out item now.

Left Hand/Right Hand and Hide Help: Player can swap the buttons to the other side of the screen, and disable the help button if player don't need it.

Hard Mode: Hard mode doesn't hide letter on the marker anymore, it tracks two previous word score played by player, and ban a score range of words for player. For example, if the player completes word “Looking” (70point), and “Coconut” (54point), it bans score range of (60-80) and (44-64).

Clear Data: If player finds their gameplay data broken or wanted to unlock achievement again, they can clean up their data in settings menu.

Version Control System Use

The whole project were archived in a version control system, I used GitHub Desktop to archive the project daily, and a few rollback is used during development.

The project is archived here:

<https://github.com/JianmengYu/Grabble>

Testing Used

The implementation is tested on both emulator and physical device, other test method were not used.

Emulator Detail:

| | |
|-------------------------------|--|
| Skin: | nexus_s |
| Size on Disk: | 1GB |
| CPU/ABI: | Google APIs Intel Atom (x86) |
| Target: | google_apis [Google APIs] (API level 23) |
| Resolution: | 480x800: hdpi |
| Google Play services version: | 8.4.89 (2428711-470) |

Physical Device 1 (with Google Play) Detail:

| | |
|-------------------------------|---|
| Device: | Huawei H60-L01 |
| API Level: | API level 19 (4.4.2) |
| Resolution: | 1080 x 1920 pixels (~445 ppi pixel density) |
| Google Play services version: | 10.0.84(038-137749256) |

Physical Device 2 (no Google Play) Detail:

| | |
|-------------|----------------------|
| Device: | Galaxy S4 GT-I9500 |
| API Level: | API level 19 (4.4.2) |
| Resolution: | 1080×1920 xxhdpi |

S1413557

Jianmeng Yu