

EEE 598 Fall 2016 Assignment-1

Checkpoint-1 Report

AugGraffiti -

Augmented Reality Art Game

Project Members:

Omkar Salvi (1209536104)

Abhishek Patil (1209417518)

Goals

High Level Objectives:

- Getting hands on experience of android application development
- Understand architecture of android system, different components and interfaces of android
- Use of different sensors in android
- Web communication in android device

Assignment Description:

AugGraffiti is an augmented reality art application. In this user has to obtain as many points as he can to win. User artist can place 3 graffiti tags at different locations. User can collect other artists' tags. When user is nearby the location of the tag placed by an artist, he will be able to see the location of tag placed by artists on the map in the application. User will also be able to know the distance and direction in which nearby tag is located. Then by clicking on that location, user can collect the tag. Collecting a tag earns user +100 points and earns artist +100 points. User will be able to see the points earned by him on the application. Also he will be able to see all the tags gathered by him in the gallery.

Design

Application Components:

- 1) MainActivity : This is the 1st activity of the application. This activity will allow user to sign-in. When user clicks on the the sign-in button in this activity , application will ask for user email id. Once user gives the email id, application will sign in the user on google. After google sign in is successful application will register the user on the server using volley module to send the request to the 'login.php' service on server.
- 2) Activity2 : This is the map activity. Once google sign-in is successful, this activity will be created and run. In this activity, we display the google map. Application will also get the current location of user and place a marker on the current location of user. Here Application will send the current location to 'neartag.php' on server. If the server sends response with nearby tags' location application will display these locations also on the map. Also we have a button to view gallery, a button for google sign-out, a textview to display current score of the user.
- 3) Place : This is the 3rd activity of the app. When user clicks on the current location marker on the google map in the Activity2, this activity will be created and run. Here we are just allowing the user to open camera and take a photo as of now. This photo will be displayed in this activity using ImageView.
- 4) NearTagService: One of features of the application is displaying all placed tags for the current user. To achieve this functionality we have created the nearTagService request. This request takes in three parameters which are user's email, current user's location parameters i.e longitude and latitude. The response is obtained from near-tag URL.

The response is in the form of comma-delimited triplets of below format:

tag_id, loc_long, loc_lat, ..., tag_id, loc_long, loc_lat

Tag_id : Defines the ID of the Tag

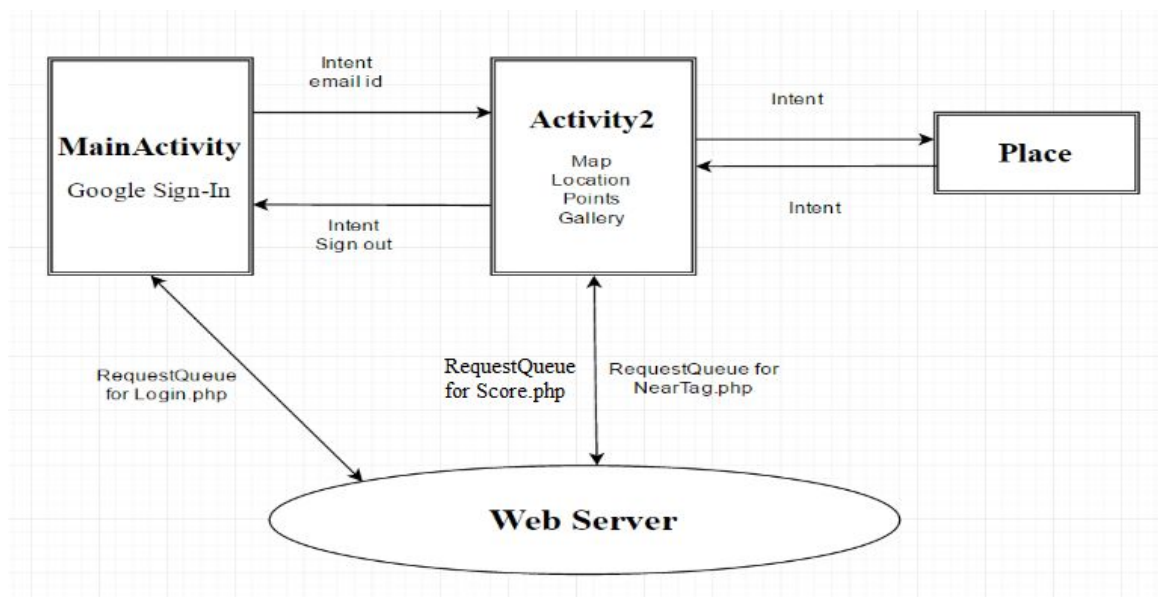
Loc_long and **Loc_Lat**: Defines location coordinates of tags placed by different users.

To handle the response received from the near tag service we have created handleNearTag Function. This function basically splits the current string response into list of parameters. By using these parameters, we are creating objects of Latlang class which are further used to create tag marker's. These markers are placed on the googlemap object.

- 5) MySingleton: The Auggrafiti app makes constant use of the network, we are setting up only single instance of the request queue that will last the lifetime of our application. To achieve this, we have implemented a singleton class that encapsulates RequestQueue and other volley functionality.

A key concept is that the request queue is instantiated with the activity context. The main purpose behind this is that the RequestQueue will last for the lifetime of your app. To clear all request, we have checked whether the request queue is empty in the onStop method of the activity. We are associating the tag object with each request of the activity. In our application, we have tagged all request in the main activity and activity-2 by using tags "actMainStop" and "act2Stop". After setting up the tag on the requests, we have cancelled all activity requests in the onStop method by calling requestQueue.cancelAll(TAG_NAME).

Interfaces between Application Components :



Strategy

Division of labor:

Since one of the team member had some previous experience with android app development, he started working on setting up the 1st and 2nd activity and user Interface design. During this period the other team member learnt all basics of android and practise them on android studio. After this the second member started working on platform developed by 1st member. From this point onwards both members keep working parallely, discussing the design and solving the issues.

Internal Checkpoints:

CheckPoint	Deadline	Team Member
User Interface	08/26/2016	Omkar
Google Sign-In/ Sign-out	08/30/2016	Omkar
Google Map	09/03/2016	Omkar
Location	09/07/2016	Omkar
Register user on the server	09/10/2016	Abhishek
Place Activity setup	09/14/2016	Abhishek
Points	09/17/2016	Abhishek
NearTags	09/17/2016	Abhishek
Documentation	09/19/2016	Omkar and Abhishek

Challenges Encountered:

- 1) Google Sign-In problem: We registered our application on google console to use google sign in functionality in the application. For this one team member registered the application using his own device ssh key and generated the API key. This API key is added to the Manifest file of application. Due to this Application works fine from the device which is registered on google console. But when we try to run the application from device of another team member, application.
- 2) Request Queue problem: This is problem encountered during the creation of different services created for login, Near-tag requests and Get-score requests. Each time, when user's location

gets changed, the new queue was getting formed. This was leading to number of requests to the server.

How you overcame the challenges:

1) For Google sign-in problem we have found following solution:

- Generate the SHA-1 from your team member's device
- Go to Google Console
- Open your application's credentials page
- Open the Android Key list item
- Add the new SHA-1 and save the changes

2) Solution for request queue problem:

Using the Mysinleton class file, we are creating only single queue for a given activity. So, every time when location coordinates are changed, if the queue is already formed then further requests are added to same instance of queue. Otherwise, If there is no existing queue for that activity then the new queue is created. Also, when current activity is stopped then the request queue is cleared to avoid unnecessary processing of requests.

Ways to Improve the Assignment:

Assignment Level:

Assignment is appropriate level of challenging for the given time period. There can be number of improvements/features to current description of an application.

User Interface Improvement: Whenever any user collects tags, tag owner should get the automated notification regarding his score increase. Also, if the tag owner is available, we can add animation to enhance the user experience.

More breadth-More depth:

We think this assignment have appropriate breadth and depth, to understand the android app development and get hands on different components of android.

Any fundamental flaws system description:

At the end of phase-1, we feel that the description of goals for phase-1 has given us basic understanding on components of application. However, we feel that little bit broad description in activity communication could have made it better. This is not actually a flaw but just a secondary improvement in description.

Privacy/Security Issues:

- **Authentication Issue:** To use the application, User must sign in with his google's gmail account. When user is successfully logged in then the application fetches his email and creates login request. If the user is already registered on the server and then user's current score is displayed on the next screen. The main flaw in this process is that once hacker(server listener)

sign in with his google account, he/she can post any user's email and get his information such as score, tags collected and gallery information.