AE3MDP COURSEWORK 2 REPORT

Student name: Killian Davis

Student number: 6523204

In this report I will detail my design and implementation decisions that I have made while completing coursework 2. The problem we were tasked with was to create a mobile application that functions as a basic location-enabled game. The basic design of my code is architecture of my code is that I use my Main Activity to launch a Maps Activity and a Service. The map activity generates 3 random markers within the range that is inputted by the user via a Slider. The map activity then uses a Local Broadcast Manager to send a message to my service which then logs the number of goals that have been reached for this run of the app. From the user perspective there are two buttons and a slider in the main activity. One of the buttons starts the game, the other stops the service that counts number of markers reached and the slider allows the user to choose the range of the markers. When they click start game they are taking to the maps activity where they are given their current latitude and longitude in the form of a toast and they can see their current position as a pink marker, and the goal positions as blue markers. Every time the user's current location changed they are notified via Toast of their new longitude and latitude. When the user reaches a goal there is a toast that says "GOAL REACHED!" and the marker disappears.

I will now discuss how I designed my app in relation to the requirements. The four requirements for this game as specified by the professor are as follows:

- Select a specific range in which targets are defined, e.g. 1km, 3km, 5km from the current location of the device.
- Randomly define 3 location targets to be reached within the selected range
- Allowing the user to view its location and the locations of the targets
- Allowing the user to have statistics such as the number of reached targets per day, the distance covered...etc

I will go through each of these requirements and discuss how I implemented each one.

1. **Specific range**

   I achieved this task by allowing users to pick which range they would like to have their goals appear in from a slider. The value of this slider was then checked and then the integer range was assigned the value corresponding to the user's choice in kilometres. This range was then converted to longitude and latitude values using a formula I found online:

```
Longitude: 1 deg = 111.320*cos(latitude)km
1km = 0.00898311175
2km = 0.01796622349
5km = 0.04491555874

/Latitude: 1 deg = 110.574
1km = 0.00904371733
2km = 0.0180874347
5km = 0.0452185866
```

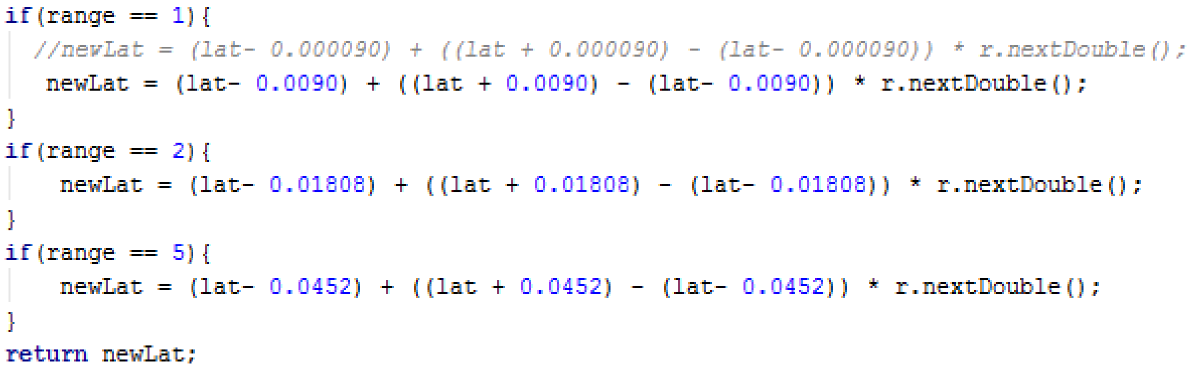
These formulae can also be seen in comments in my map activity.

2. Randomly define 3 locations

For this task I used the formulae stated above to get the maximum and minimum longitude and latitude that would be allowed by the range. Below I have shown my code for how this was achieved for latitude where lat is our current latitude. The general formula for the code below is as follows:

```
Random r = new Random();
double randomValue = rangeMin + (rangeMax - rangeMin) * r.nextDouble();
```

This formula was taken from http://stackoverflow.com/questions/3680637/generate-a-random-double-in-a-range

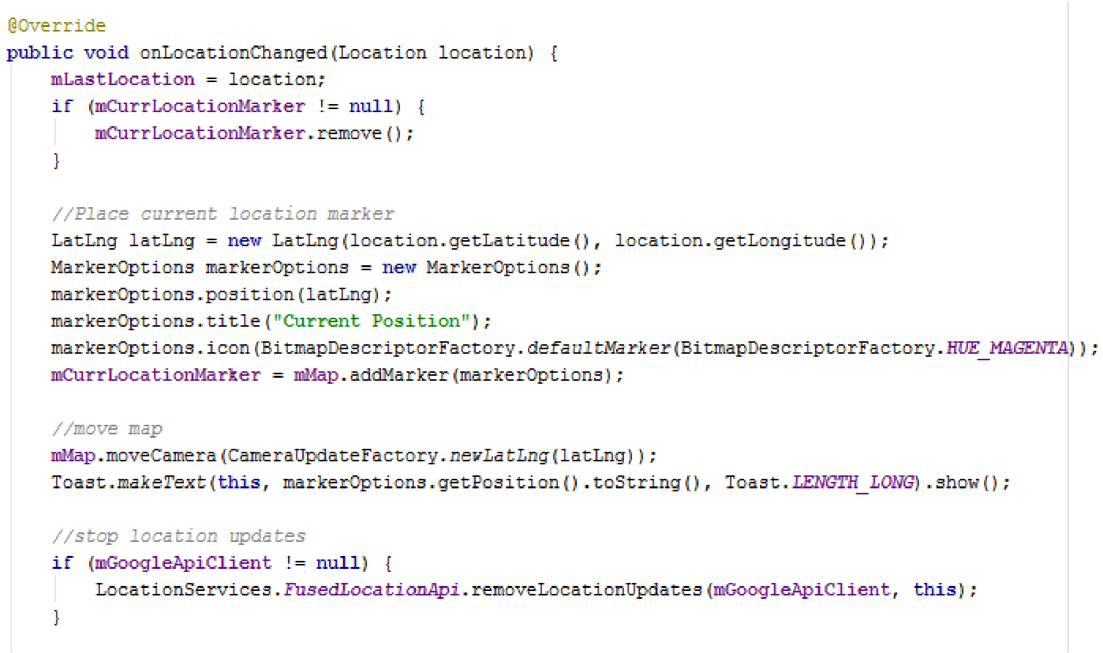


```
if(range == 1){
   //newLat = (lat- 0.000090) + ((lat + 0.000090) - (lat- 0.000090)) * r.nextDouble();
    newLat = (lat- 0.0090) + ((lat + 0.0090) - (lat- 0.0090)) * r.nextDouble();
}
if(range == 2){
    newLat = (lat- 0.01808) + ((lat + 0.01808) - (lat- 0.01808)) * r.nextDouble();
}
if(range == 5){
    newLat = (lat- 0.0452) + ((lat + 0.0452) - (lat- 0.0452)) * r.nextDouble();
}
return newLat;
```

Once I had the longitude and latitude of my new points I simply created a new marker point object and gave these values as inputs to the marker.

3. Allow target to view its location and the location of the targets

This task was helped greatly by the google maps API functions in android studio but there is still a lot of code to write to get it working. First of all we must define in the manifest file that this app has the permission to access our network state, fine location, coarse location and the internet. We then must generate an Android API key and add this to our manifest also. I then used the longitude and latitude of UNNC as our starting points for the game. I then gave those values to a marker point I called origin which will then place a marker at this point as our start location. I used the UNNC values as default longitude and latitude because when I was testing my code, if the origin did not have default values the app would crash. However, once we start moving the current position marker will move with us as shown in my Location changed function:

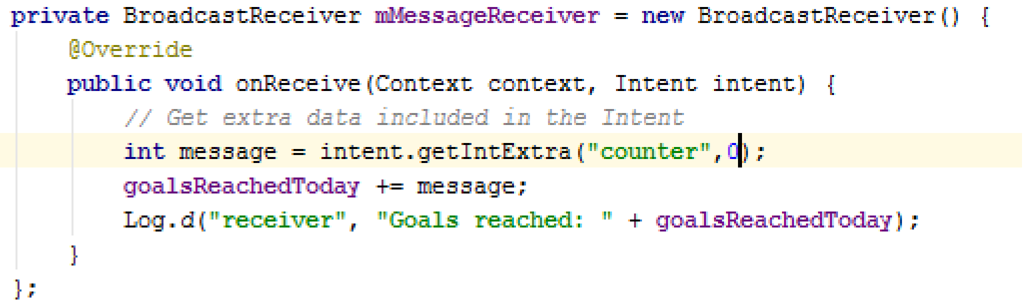


```java
@Override
public void onLocationChanged(Location location) {
    mLastLocation = location;
    if (mCurrLocationMarker != null) {
        mCurrLocationMarker.remove();
    }

    //Place current location marker
    LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions();
    markerOptions.position(latLng);
    markerOptions.title("Current Position");
    markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_MAGENTA));
    mCurrLocationMarker = mMap.addMarker(markerOptions);

    //move map
    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
    Toast.makeText(this, markerOptions.getPosition().toString(), Toast.LENGTH_LONG).show();

    //stop location updates
    if (mGoogleApiClient != null) {
        LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient, this);
    }
```

I found this task to be the most challenging simply due to the fact that my laptop is not powerful enough to use a virtual device to input new longitude and latitude values for current location. As a result of this I had to go and walk around a lot and the signal for mobile data is not very good on campus so testing was erratic at best.

4. Allow the user to have statistics such as number of targets reached and distance covered.

I tackled this task by using a local Broadcast Manager to send a value of 1 to my service (which was acting as my broadcast receiver). This value was then added to my integer goalsReachedToday.



```java
private BroadcastReceiver mMessageReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        // Get extra data included in the Intent
        int message = intent.getIntExtra("counter",0);
        goalsReachedToday += message;
        Log.d("receiver", "Goals reached: " + goalsReachedToday);
    }
};
```

I then simply logged the value of goals reached today with a message that says “Goals reached: …”