Mobile App Design Project
GPS Location Display with Map and Location Plotter

Description:

This assignment has you take an existing App and create the code to do the following:

- Read GPS Location Values from the Android GPS sensor and display them in TextView objects on the screen.
- Deploy these GPS Values to draw a yellow dot on a Marist Campus Map representing your location.
- Provide EditText fields for you to input a Target GPS Location.
- Display the Target GPS Location on the Marist Map as a red dot.
- Explore other Location class methods to provide data such as altitude, bearing, distance, speed . . .

Directions:

Phase 1: Setup

1. Download and unzip the GPSMapAssignment package at: http://nebomusic.net/androidlessons/GPSMapAssignment.zip.
2. In Eclipse – Select File->Import and import the GPSMapAssignment Package
3. Run and test the App on the Nexus Phone

Phase 2: Program the Main.java to display Location Data (75 Points)

1. Using the GPS Display Directions at http://nebomusic.net/androidlessons/GPS_Data_Display_Project.pdf as a model, complete the code in the Main.java section to display the Longitude and Latitude in the TextView fields.
2. Test with the Nexus Phone.

(Additional GPS Sensor Description available at: http://nebomusic.net/androidlessons/gps_sensor.pdf)

Phase 3: Write the code to pass GPS data to the MapView class (5 Points)

1. Consult the MapView.java code and examine the functions you need to complete to location data from Main.java. These function are:

   public void setMyLat(double lat)
   public void setMyLon(double lon)
   public void setDestLat(double lat)
   public void setDestLon(double lon)

2. Modify the code in Main.java to use these functions to pass along the location latitude and longitude data.

Phase 4: (Where the computation comes in!) There are two functions in MapView.java that perform the calculations that take the location data and plot the yellow dot on the map. (10 Points)

1. Finish these functions using the location data and the constants provided in the fields of MapView.java.

2. (Hint . . . Think about the proportions of the map. You need a ratio of the position in space versus position on the graphic of the map. How does longitude relate to X and latitude relate to Y?)

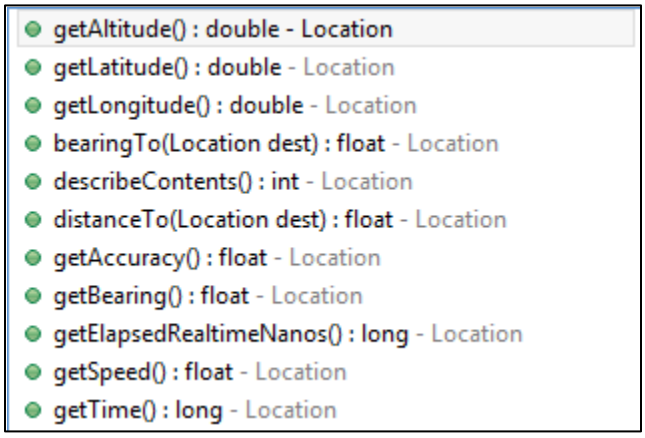
3. Test the app – the yellow dot should appear approximately in your location.

Phase 5: Target Destination (5 Points)

1. Modify the Main.java code to take user input in the EditText fields for latitude and longitude.
2. Pass these user inputs into the MapView.java. (Remember the functions you wrote earlier?)
3. Program the red dot to appear where on the Target Destination.
4. Test the App with some obvious fixed references on Marist Campus.

Phase 6: Extras (Up to 10 Points) (2 Points per data field)

1. Explore and experiment to add more data to the display. Several options are shown below.



Grading: Project Maximum: 100 Points

Phase 2: 75 Points
Phase 3: 5 Points
Phase 4: 10 Points
Phase 5: 5 Points
Phase 6: Up to 10 Points