**CSE 6331.002, Spring 2018**

**Designing University of Texas at Arlington’s map on a GIS and Spatial database systems**

(Section H)

By

Nikhar Patel - 1001371707

Ridham Kothari- 1001446086

**Introduction:**

In this project, we have created a section of detailed University map for part of the UTA campus. The map includes many of the departments, apartments, residential halls, different playgrounds and administrative offices. For all these, we have generated KML files using Google Earth and then converted them to shapefiles. We have used these shapefiles in QGIS.

In map creation process, we created spatial objects like lines, points and polygons using google maps (google earth pro). In each area, we have marked all points for entrances to buildings, lines for connecting adjacent paths or roads – the focus was on walking paths only including sidewalks of roads, not roads for vehicles, and polygons. (buildings, lawns, parking lots, etc.)

Once the entire map was created, the next objective was to find the shortest path between any two points. We have used QGIS plugin name as ‘Road Graph’ to find shortest path from marked path in shapefile.

To use core feature of the QGIS and Python we have extracted all the features of each layer and applied few functions like length of line, area of polygon, total length of all the lines, attributes of features and displayed results. Python file can be found named as “area-length-point.py.”

To achieve the objective of this project, we have we played with several spatial queries applied on spatial database name “spatialdb.sqlite”. On of the query is calculate distance between given tow buildings in meters implemented in “databaseconnection.py”

It will also identify points of interests (POIs) on the map based on the category of the POI. Regions can be divided into sub-categories like academic buildings; administrative buildings; athletic fields/playgrounds; parking lots; etc. To site an example, we have shown parking lots available on UTA campus code can be found in file name type of “building.py”

**Software and Tools used:**

* QGIS
* Spatial lite database
* Python console
* Road Graph Plugin

**Procedure for getting started:**

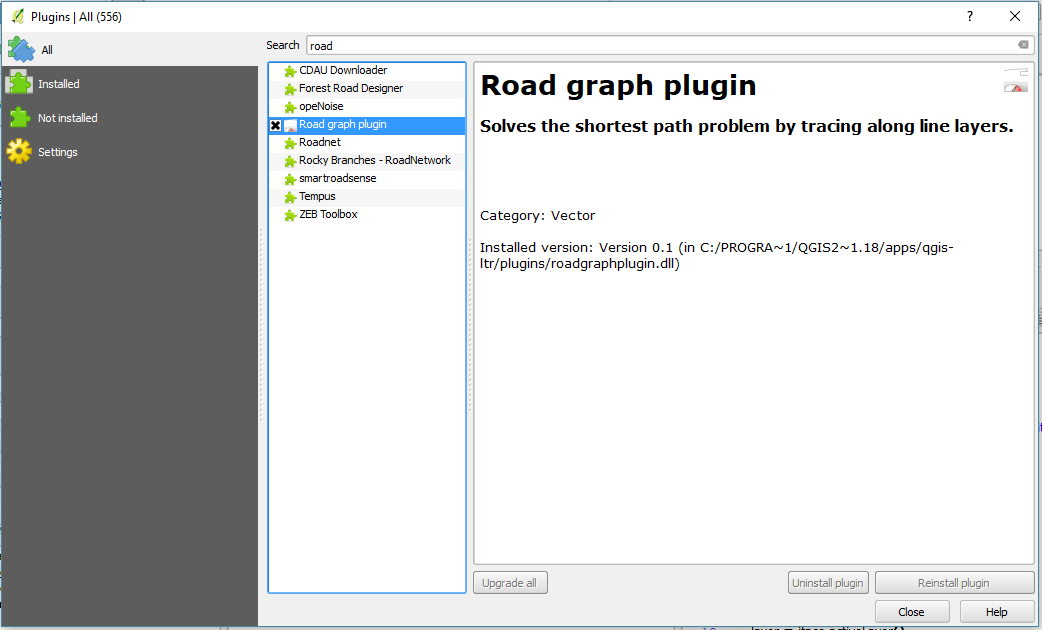
* First, we have generated KML files of all the apartments, roads, residential halls, administrative offices etc. using “Google Earth Pro”.
* Then we have loaded every KML files in QGIS.
* Using QGIS, we have generated individual shape files for each KML.
* After generating individual shape files, we merged them all to create a single detailed shape file which covers each of the buildings, their entrances and paths in between them
* Furthermore, we imported those shapefiles into spatial-lite database as one table for each layer
* We have implemented number of functionality using it.

**Functionality:**

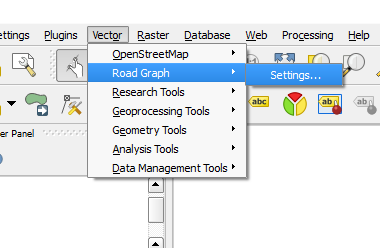
**Shortest Path:**

* We can find the shortest path between any two points.
* For this feature, we have used **Road Graph Plugin** of QGIS.
* It shows path to a vector layer and calculates shortest path as well as travel time.
* We have implemented in following steps.

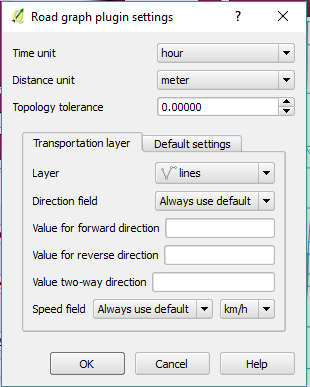
1. Install plugin ‘Road Graph’ using official instruction manual



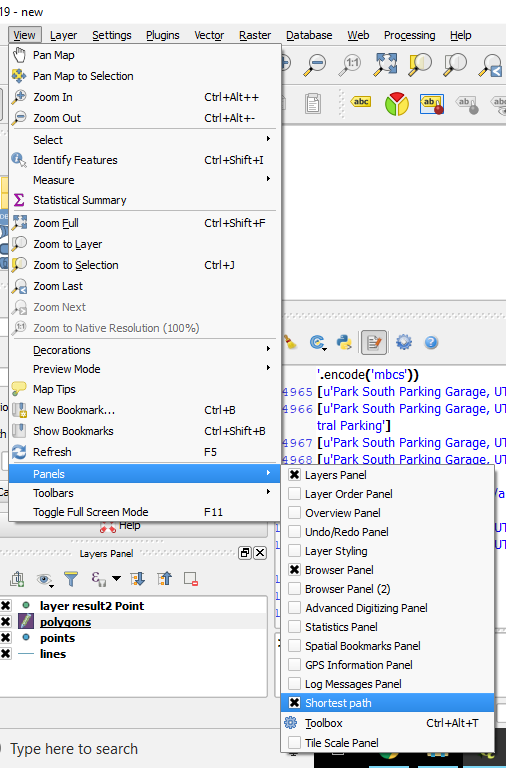
1. Go to Vector -> Road Graph - > Settings



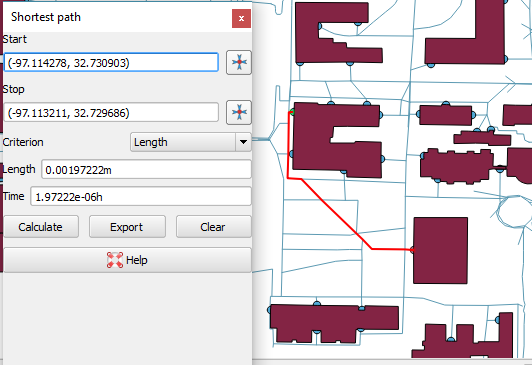
1. Select layer for the path (*uta path*) and close



1. Go to ***View*** - > ***Panel*** -> ***Shortest Path***



1. Select Start and Stop point from the canvas and click on ***Calculate***.
2. Results are displayed on canvas.



**Basic features of QGIS & Python:**

* It shows use of basic functionality of some of the pyqgis functions like iface, geometry, QgsWKBTypes, etc.
* It shows points, length of lines, area of polygon, total length of all the paths for selected layers.
* Following is the procedure we have used
  1. Python console plugin is already downloaded with the QGIS. So, we don’t need to load externally
  2. Go to ***Plugins*** - > ***Python Console***. It will open python console in QGIS.
  3. Click on ***Open Script*** icon to load python file into console.
  4. Once file is open in console select layer from layer panel to run the script.
  5. After selecting a layer run ***area-length-point.py*** file from console using Run.
  6. It displays the results in console.

**Using spatial query:**

* This shows use of spatial query to retrieve information from database using spatial query.
* In spatial query we can use some specific functions like distance, intersect, union, overlap, buffer, etc. In our example we have used distance function.
* This program shows distance between two given buildings from the map with their name. Query can be modified to find distance from one building to all other buildings.
* Following is the procedure we have used
  1. Python console plugin is already downloaded with the QGIS. So, we don’t need to load externally
  2. Go to ***Plugins*** - > ***Python Console***. It will open python console in QGIS.
  3. Click on Open Script icon to load python file into console.
  4. Make sure you have connected to spatial-lite database named ***spatialdb.sqlite***. Some libraries are required to run the script mentioned in python file
  5. Then run ***databaseconnection.py*** file from console using Run.
  6. It displays the results in console.

**Points of Interest:**

* To show results according to the category of different points of interest such as regions, type of building, etc.
* In our example we have displayed results according to type of building POI.
* Following is the procedure we have used
  1. Connect to spatial-lite database
  2. Load database tables in layers
  3. Select polygon layer from all the layer and select ***Toggle Editing*** option by the ***right-click*** on layer
  4. Then again ***right-click*** on layer and select ***Open Attribute Table*** option.
  5. Select ***New Field*** option from the top of the window.
  6. Set name as ***‘BuildingType’*** and datatype as ***‘Text’***
  7. Then add type of building in each feature
  8. Open python console using procedure mention above
  9. Open python file name type of ***building.py***
  10. Run the script and results will be displayed in the console