Ethan Shaw
s3600532

Major Project

Geospatial Programming

GitHub: https://github.com/LordRaini/CartographicPleasures

Project.

For my project, I created a script that could generate a series of points for use in R Studio to recreate Joy Divisions' Unknown Pleasures album artwork. The artwork is famous for its data visualisation (Figure 1).

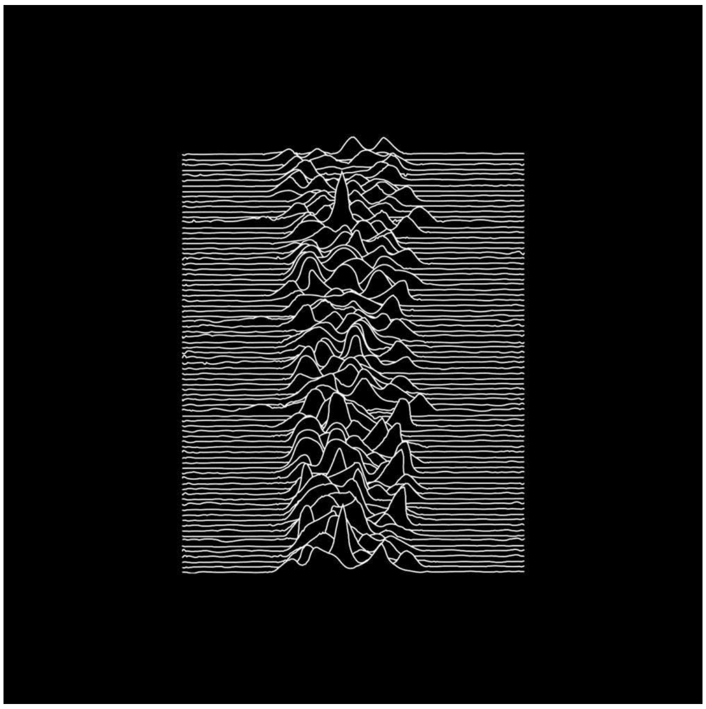

Figure 1. Joy Division's Unknown Pleasures

Travis M. White developed a methodology to recreate this artwork using a digital elevation model (DEM) and vector layer. The DEM is used to generate heights for vector layer while the vector layer is used to generate to generate lines then points which will hold the XYZ of each point. This point layer will then be converted into a comma separated values (.csv) to be used in R Studio. (https://cartographicperspectives.org/index.php/journal/article/view/1536/1726)

For my project, I used his process as a guide to understand the steps he took, then developed a script to automate the QGIS section of his guide. The original process when first designed was as follows (Figure 2):

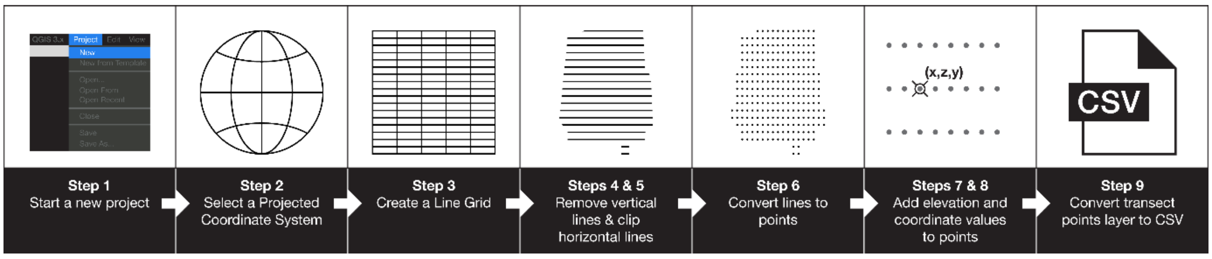
Step 1
Start a new project
Step 2
Select a Projected Coordinate System
Step 3
Create a Line Grid
Steps 4 & 5
Remove vertical lines & clip horizontal lines
Step 6
Convert lines to points
(x,z,y)
Steps 7 & 8
Add elevation and coordinate values to points
CSV
Step 9
Convert transect points layer to CSV

Figure 2. Cartographic Pleasures Process

Flow Chart.

Figure 3 shows the code flow chart I developed for my automated process.

First libraries are imported.

After is a section of variables to edit allowing the code to run.

After the directory is changed to the chosen one to keep all files in a single spot.

After the two specified layers were added as Qgs Objects.

From the Vector Layer Qgs Object, a Coordinate References System (CRS) is generated using Proj4.

A new project is generated with the new CRS allowing the CRS to be consistent within QGIS.

The layers are reprojected using the new CRS.

The number of grid lines to generate is calculated based on the code input and vector layer extent.

Grid lines are generated using the previously calculated values.

Vertical Lines are deleted.

The grid is clipped based on the vector layer boundary.

Lines are converted to points using the specified spacing.

A new column is added to the points layer attribute table.

The new column is filled with heights from the DEM.

Coordinates are generated for the points.

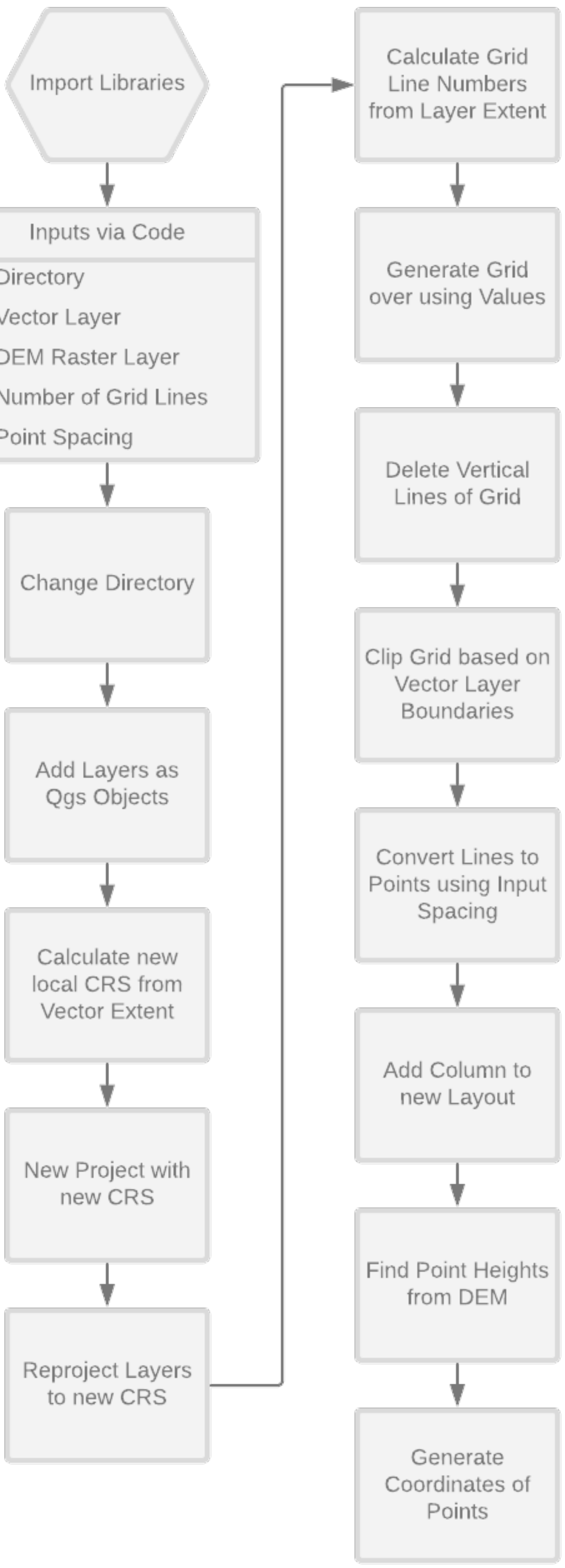
Import Libraries
Inputs via Code
[1] Directory
[2] Vector Layer
[3] DEM Raster Layer
[4] Number of Grid Lines
[5] Point Spacing
Change Directory
Add Layers as Qgs Objects
Calculate new local CRS from Vector Extent
New Project with new CRS
Reproject Layers to new CRS
Calculate Grid Line Numbers from Layer Extent
Generate Grid over using Values
Delete Vertical Lines of Grid
Clip Grid based on Vector Layer Boundaries
Convert Lines to Points using Input Spacing
Add Column to new Layout
Find Point Heights from DEM
Generate Coordinates of Points

Figure 3. Program Flowchart

Development.

The outline of the project was simple having been given a website with a full step-by-step process. To start off development I imported the layers Travis used and followed his guide on building the Joy Division artwork in QGIS. Once I followed his steps, I had a good guide I where I needed to go with my code.

Firstly, I imported the necessary libraries and referred to the layers and directory I would be using. Then I had to create a new CRS from the layers. This was my first hurdle. Understanding how the CRS works within QGIS was confusing, first I tried building a CRS from the layer and setting the CRS of the project and layer to the new CRS, but this did not work fully. The first time the code ran it did not work but the second time it did. I'm still unsure why but decided to go some research and found a solution of creating a new project instance and setting that CRS to the new one worked perfectly.

From there I attempted to calculate the extent of the vector layer so I could calculate the number of grid lines required for the process. This was unsuccessful because the grid extent value wasn't in metres even though I set it as such in the Proj4 CRS. I hard coded this into the process so I could move on and generated the grid using Travis' values. From there some attribute table editing was used with a FOR loop to assess whether a line was horizontal or vertical. The attribute table had 5 columns: ID, Left, Right, Top, Bottom. By pulling the ID, Left and Right I could check whether a line was horizontal by comparing if there was a different in Left and Right values. If not, I appended the ID to a list which would be used to delete the fields after the FOR loop was completed. The list was then inputted to a delete field line and those lines were deleted but it should be noted it does not delete the first vertical line and I am still unsure why.

After the Grid was clipped using the original vector boundary before points were generated using the inputted spacing at the top. From there the point Z values where found referring to the DEM but I had three issues before this could work. The first being that I needed a new column to put these values into, the second was that these layers needed to be on the same CRS for this to work and lastly the process was pushing an error whenever I wanted to look at the inputs form the processing toolbox. The first issue required me to use the Add attribute field to a table process before I could run the DEM. Once these were done, I had another issue. While for the second issue they looked to be on the same layer in the QGIS renderer, they were indeed not. I had to add two processes in earlier that would reproject the layers using the processing toolbox which worked perfectly to put everything into the same CRS. Finally, for the last issue I was able to fix this by running the .exe labelled "QGIS 3.4.13 with GRASS 7.6.1, so my thoughts are that something either wasn't working within my QGIS or QGIS does not automatically open with GRASS running. This is confusing but an easy work around. The GRASS7:v.what.rast process used in this step to obtain the DEM Z values was chosen to specifically avoid the idea of having the user need to download a plugin before using this script, which was achieved, but not perfectly. While this script can run without needing to download a new plugin, you may also need to switch .exes or enable GRASS in the plugins.

After this a simply generate coordinates of the points was used to have a final output which was a point layer the XYZ information.

Ethan Shaw
s3600532

Discussion.

The submission for this project holds three scripts.

The first labelled MajorProject_Save.py or LordRaini-patch-1 within GitHub is the most up-to-date version which works if you set the inputs within the code. I could have maybe created proper input requests but decided to focus on functionality before I attempt to make the program user-proof. This program makes the processes save each output then the output is brought into QGIS before moving to the next process.

The MajorProject.py in the master branch is the work in progress code before I made corrections such as fixing the CRS and extent issues I was having.

The LordRaini-ProcessingScript or CartographicPleasures.py is the attempt at a processing script using the QGIS template. This would allow me to restrict user input using the tools provided by QGIS, but I was unable to get this function. I am currently having an issue with a dictionary INPUT within the code. This may be just a spelling error or line I'm missing but I cannot find the issue so had to halt this development.

The final branch I have is the LordRaini-MemoryOnlyAttempt-WIP. This was an attempt to remove the cluttered design of saving every output and reinputting it into QGIS using the 'memory' output. This is an odd development. I had the issue with how to refer to the memory layer and not having them override each other. I was able to solve this referring to some other code that using this setup:

$$Variable = Process['OUTPUT']$$

Along with instead of using 'memory' I used 'memory : name' which seemed to work, and no errors were presented when the code was ran. I ran into a couple of issues with GRASS7 and SAGA not recognising these outputs, I converted the outputs to Qgs Objects as a new variable which solved this problem. But when I saved the output Point XYZ file and viewed it the file was empty. This means I am having a logic error in the code as the program works but the output is incorrect. Finding resources online on how to handle chained processes was scarce.

Finally, an attempt was made to export the point XYZ file to CSV, but I also had issues with this. The CSV output has been modified multiple times recently so current guides on how to use it are scarce as well as people not exporting to CSV often. I decided in the end to leave this out as exporting to CSV using the GUI is easy within QGIS.