Connection of Bike Paths & Schools

Sarah Grego - Final Project

Dr. Jennifer Koch – Spatial Programming & GIS

8-01-2021

Abstract:

Python is an important tool that contributes to building code in a manageable and readable environment. Users rely on python to use geoprocessing tools that can be broken down into small easy steps. By using data from the City of San Angelo GIS Department (COSA) in Python, directs a path for geoprocessing tools such as buffer, clip, select analysis, and finding lengths to be coded. Coding these geoprocessing tools in python provides a maintainable source for testing and exploring the datasets. The purpose of using python in this study is to communicate the users intentions of the chosen dataset through inputs and output codes. The main idea is to use functions that will represent how inputs and outputs can correspond within the program resulting in algorithms that will be used to geoprocess the dataset while capturing the users' intentions. These functions are translated into ArcGIS Pro to give users a visual of the outputs the geoprocessing codes create. This project outlines creating, implementation, and results of IDLE codes and geoprocessing tools for spatial programming research. As with most spatial coding research reports, a programming research paper is highly technical to present information.

Keywords: Spatial Analysis, IDLE Python GUI, Geoprocessing Tools, ArcGIS Pro

Introduction:

To achieve daily tasks, communicating and connecting problems to the proper solution is essential. IDLE (Python GUI) is among other program languages users utilize to communication problems and solutions. IDLE uses two main window types, along with a shell, and editor windows. A main window and a module to stimulate solutions to problems. Two loops and two conditional statements aid in structuring the code in IDLE.

A simple task students must achieve daily is traveling to and from school. Learning how to choose the most efficient route will assist in students to have easeful travels. The target group are students who ride a bike to school in San Angelo, Texas. Provided shapefiles from the City of San Angelo GIS Department of schools and bike paths will assist in solving problems and answering the research question of, "Do bike paths connect students to school?" By selecting inputs that triggers the output in the main window, communicate to a user how the two shapefiles connect within each other. Incorporating different geoprocessing tools: buffer, clip, selecting attributes, and finding lengths. These geoprocessing tools performs spatial analysis to the bike and school shapefiles. Finally, comparing and analyzing the codes from the selected datasets contribute to the solution of the overall question of this research if students are well connected with bike routes around school.

Materials:

The informative materials used in this study consisted of two programs for coding and mapping the data. IDLE Python GUI performed scripts to use the data in a correct manner. ArcGIS Pro is used to cartographically display results of the code from IDLE. IDLE is integrated with a python shell window that interacts with codes input and outputs. This shell is a great resource to utilize if the code had any syntax errors to correct. Jupyter Notebook was used in the beginning steps of creating the format of the codes in IDLE because it is most familiar. Lucid is used to display all codes defined in IDLE and Jupyter Notebook in a flowchart format. The City of San Angelo's GIS Department provided datasets of bike paths and schools for the area.

The main shapefiles provided by COSA are named schools.shp and bikeroute.shp. The bikeroute.shp consisted of polylines that included all bike routes within San Angelo that was later queried out around the schools' points. The attributes in this shapefile had the following: length, name, street names. The schools.shp contained all elementary, middle, and high school in San Angelo with points. The attributes used inside the schools.shp for the queries are type of school and coordinates.

Methods:

In preparing the code to find solutions in this dataset, module.py and main.py is created in IDLE. The module.py file defined all following functions: workspace, dataset, high school, attribute, tool, function, and execute tool (Figure 1). The workspace function allows the user to enter a file they have shapefiles to code. The next function, dataset, allows the users to select which shapefile they want to geoprocess. The high school function queries out all the elementary and middle schools and only selects the high schools from the original schools.shp. The defined functions called tool and function allows a user to select what they want to do with the shapefile, clip, buffer, or find length. Then, the code will ask the user how they want to use the geoprocessing tool. For example, if the user selects buffer, it will prompt them to enter how many miles they want the buffer to surround the school. The user can choose from 1, 2, or 3 miles with the execute function.

Figure 1:

```python
# define workspace
def userWorkspace():
    URL = input("Enter file path")

# define shapefile to use
def userDataset():
    userData = input("Enter name of file 'schools.shp', bikeroute.shp'")

# define High Schools in San Angelo by selecting the correct attribute and create a new shapefile
def highSchools():
    arcpy.env.workspace = "C:/Users/sgreg/OneDrive/Documents/gis5653/Final"
    arcpy.Select_analysis("schools.shp", 'SUBSET_SELECTION', '"EDU_TYPE" = \'High\'')

# define what tool the user wants to use
def userFunctionTool():
    functionTool = input("Enter geoprocessing tool 'clip', 'buffer', or 'length'")
    if functionTool == 'clip' or functionTool == 'buffer' or functionTool == 'length':
        print("Tool", functionTool)
    else:
        print("invalid")

# Ask user what to do with tool
def userFunction():
    userTask = input("Enter data to find 'buffer around school', 'a clip of bike paths', or 'length of path'")
    if userTask == 'buffer around school' or userTask == 'a clip of bike paths' or userTask == 'length of path':
        print("Task", userTask)
    else:
        print("invalid")
```

The functions called ‘userFunctionTool’ and ‘userFunction’ use a conditional statement and loops to create the code (Figure 2). These imbedded tools allow the code to output a value.

Figure 2:

```python
# define what tool the user wants to use
def userFunctionTool():
    functionTool = input("Enter geoprocessing tool 'clip', 'buffer', or 'length'")
    if functionTool == 'clip' or functionTool == 'buffer' or functionTool == 'length':
        print("Tool", functionTool)
    else:
        print("invalid")

# Ask user what to do with tool
def userFunction():
    userTask = input("Enter data to find 'buffer around school', 'a clip of bike paths', or 'length of path'")
    if userTask == 'buffer around school' or userTask == 'a clip of bike paths' or userTask == 'length of path':
        print("Task", userTask)
    else:
        print("invalid")
```

The geoprocessing tools, buffer, clip, and length are defined in the module. These tools are designed in the code to utilize the users’ recommendations they input from the functions and for the programmer to insert correctly where they should be in code to be geoprocessed. These tools are defined below the functions in the module.py (Figure 3). The buffer tool creates a polygon around the high schools select function. A new shapefile called buffer.shp is generated. The clip tool is designed to overlays any bike routes inside the buffered polygon. Like the buffer tool, the clip will also generate a new shapefile of the clipped features called clip.shp. The length tool uses the clipped bike paths attribute called ‘SHAPE_Leng”’ to find the miles of all the paths.

Figure 3:

```
def buffer():
    arcpy.env.workspace = "C:/Users/sgreg/OneDrive/Documents/gis5653/Final/schools.shp"

    # set varibles to users needs
    inFeat = "High_Schools.shp"
    outFeat = "High_Schools", "C:/Users/sgreg/OneDrive/Documents/gis5653/Final/buffer"
    distance = "1 mile"

    # execute buffer
    arcpy.Buffer_analysis(inFeat, outFeat, distance)

# create a clip

def clip():
    arcpy.env.workspace = "C:/Users/sgreg/OneDrive/Documents/gis5653/Final"

    # Set local variables to users needs
    in_features = "bikeroute.shp"
    clip_features = "buffer.shp"
    out_feature_class =  "C:/Users/sgreg/OneDrive/Documents/gis5653/Final/clip.shp"

    # Execute Clip
    arcpy.Clip_analysis(in_features, clip_features, out_feature_class)

# Define length of clip.shp bike paths

def length():
    arcpy.env.workspace = "C:/Users/sgreg/OneDrive/Documents/gis5653/Final"

    # set local varibles to users needs
    inFeatLen = "clip.shp"
    field = "SHAPE_Leng"
    lengthUnit = "Miles US"
    len = "1 Mile"

    arcpy.polyline.getLength(inFeatLen, field, lengthUnit, len)
```

The module.py is used to develop code for the main.py to call the code in. The main.py imports arcpy for the geoprocessing tools to work. Then the module.py is imported in IDLE to set where the user is calling the defined function from. After, the functions are put in the main.py to be applied and ran (Figure 4).

Figure 4:

greg0058_main.py - C:\Users\sgreg\greg0058_main.py (2.7.18)

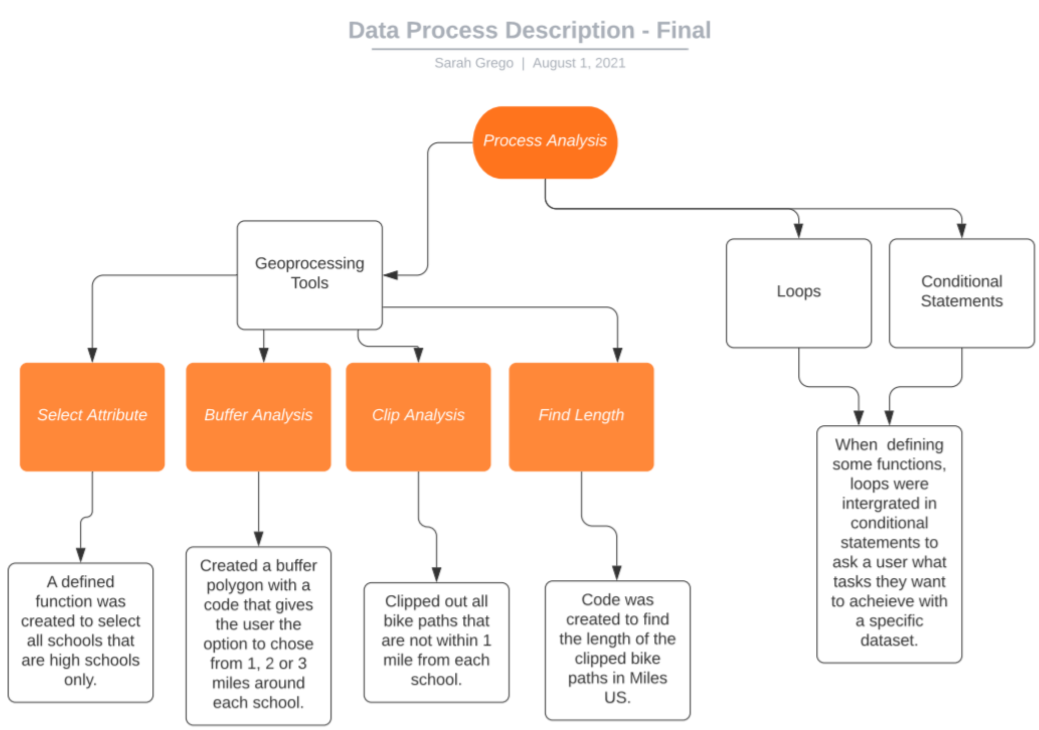
File Edit Format Run Options Window Help

```
import greg0058_module

import arcpy
greg0058_module.userWorkspace()
greg0058_module.userDataset()
greg0058_module.highSchools()
greg0058_module.userFunctionTool()
greg0058_module.userFunction()
greg0058_module.executeTool()
greg0058_module.buffer()
greg0058_module.clip()
greg0058_module.length()
```

Using the the application Lucid, a flowchart was created to display the codes generated in the module.py (Figure 5).

Figure 5:



Results:

The geoprocessed data reveled that there were bike paths around high schools in San Angelo, TX. There was a total of seven high schools in the area with only three that had bike routes one mile around the school. The other four high schools were spatial in rural areas where the bike paths are not located. The three high schools, Lake View High School, Central High School, and Central Freshman Campus are in the central of the city which would explain why they have bike paths. Lake View High School had a total of three bikes paths selected one mile around the campus. The total of all bike path around high schools estimates to 25.8 miles. The bike routes around this high school estimate to 4.6 miles of path within the one-mile buffer. Central High School had a total of 17 bike paths within the one-mile buffer. The total estimate miles within the buffer of this campus are 12.6 miles. Central Freshman Campus had 11 total bike paths inside the one-mile buffer with a total of 8.6 miles of bike path. This shows that Central High School has the most bike routes to choose from than the other two schools. The figures below cartographically display the geoprocessed selection.

Figure 6:

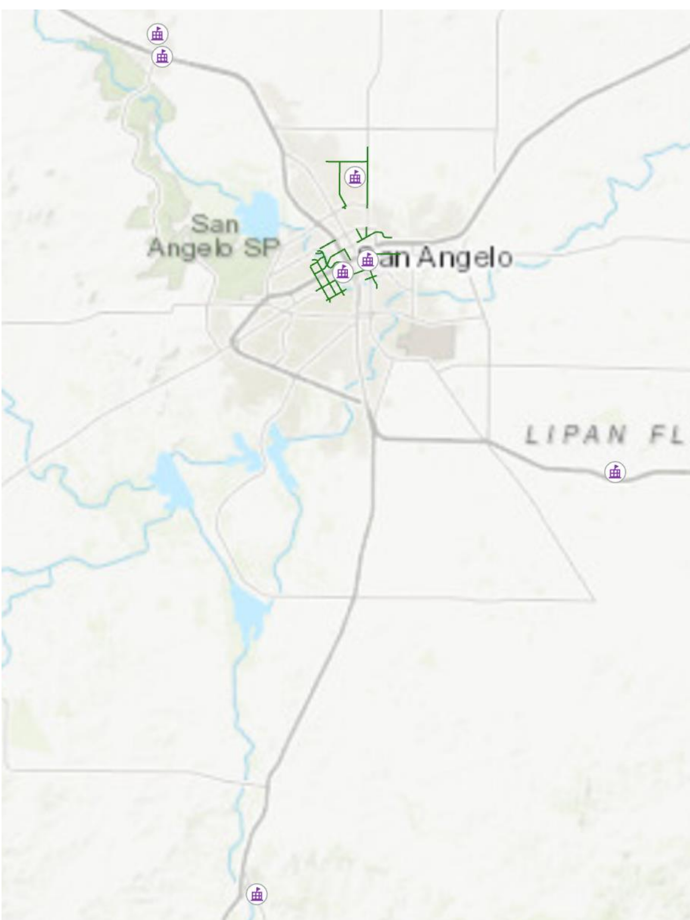


Figure 6 shows the selections of all the high schools in the area that were queried from the select attribute function.

Figure 7:

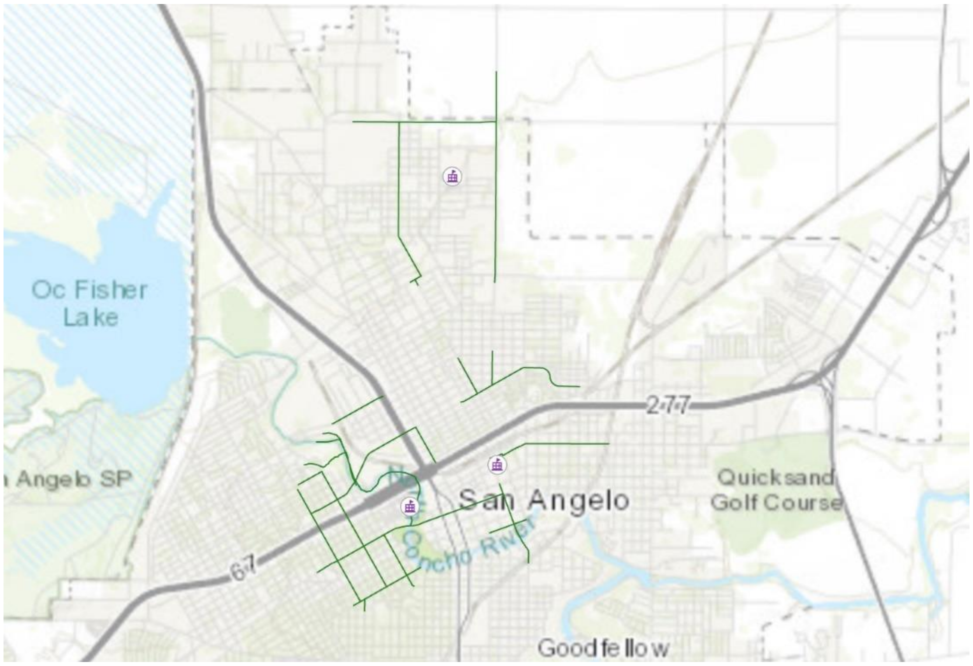


Figure 7 shows the three high schools located around the central of the city. The green polylines are the clipped bike paths from the buffer.

Figure 8:

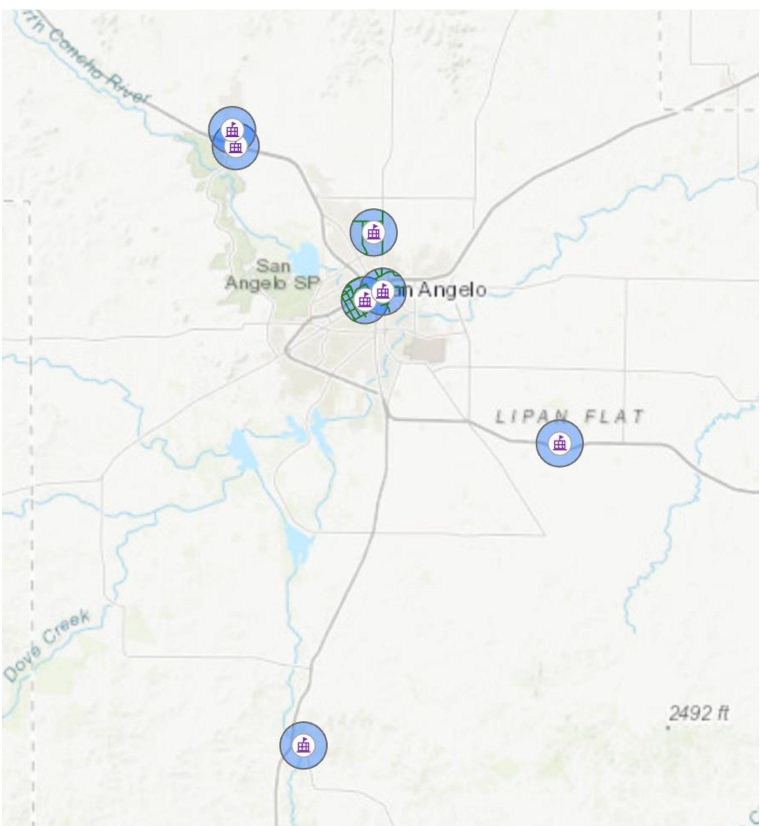


Figure 8 shows blue, circle polygons which is the one-mile buffer that was created by the defined buffer geoprocess tool.

Figure 9:

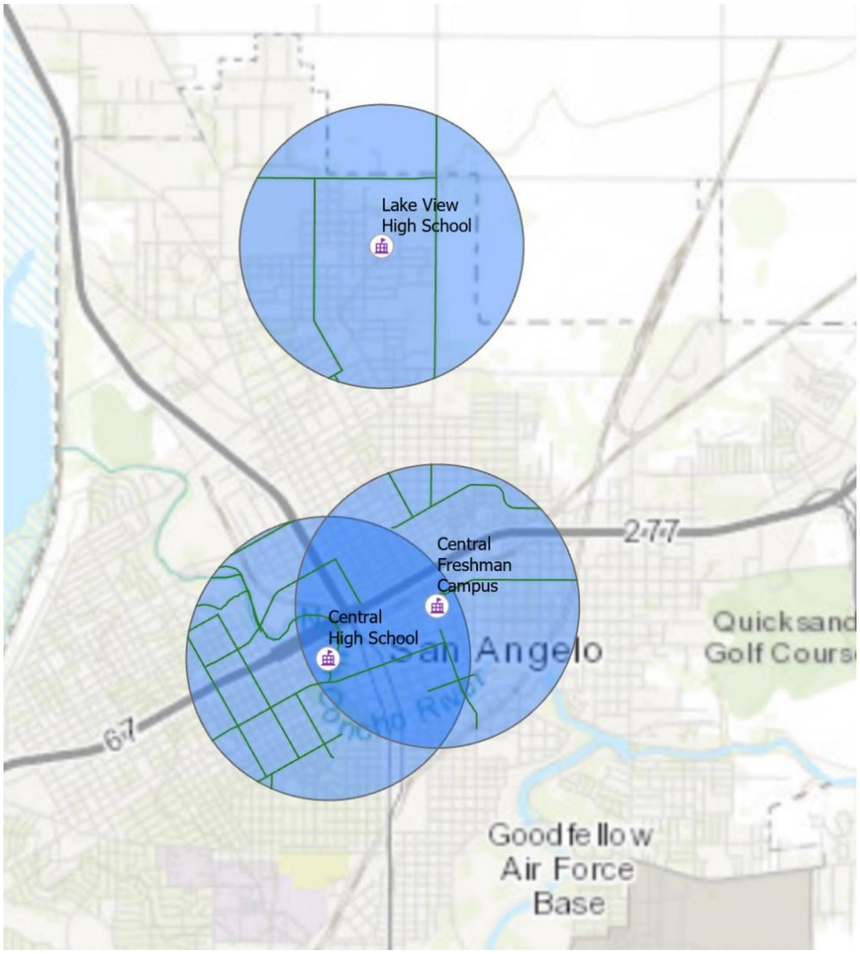


Figure 9 is the zoomed in images of the three high schools with bikes paths. The green polyline is the bike paths. The purple and white point are the high schools. The blue polygon is the buffer.

Figure 10:

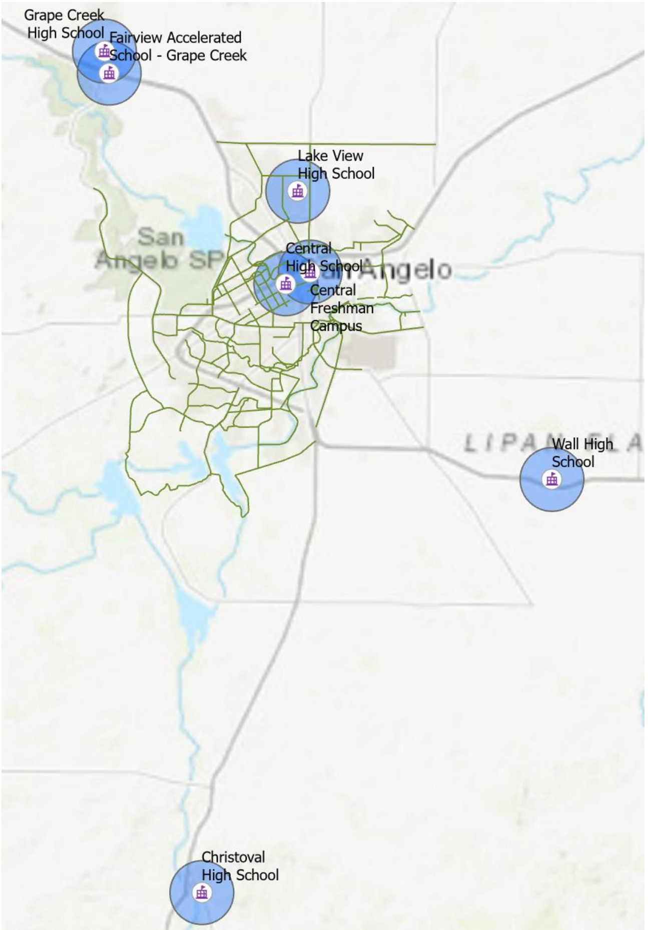


Figure 10 is all high schools and all bike paths with the one-mile buffer at the in the target location.

Discussions and Conclusions:

The purpose of this research effectively utilizes codes in IDLE to measure the lengths of bike routes around each high school. The research question is effectively solved through these methods and returned that the bike paths do connect students to school. The communication of functions between the module.py and the main.py in IDLE created significant codes to find solutions by enabling the user to describe the length of the bike path around high schools. The geoprocessed tools queried down the targeted areas of the shapefile with results that showed 27 bike paths for students around the high schools in the city. The cartographic representations connect an adequate amount bikes paths for students to utilize around campus. The buffers displayed the one-mile radius around the high schools and the clip showed the paths close to campus.

Writing code is intimidating to beginner programmers. Limitations of this work consisted of writing the code correct with zero errors. The preformed codes were small tasks that can be rewritten more efficiently. This project can be continued by incorporating the geoprocessing tools into the functions code.

References:

"Buffer (Analysis)." *Buffer (Analysis)-ArcGIS Pro | Documentation*, pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/buffer.htm.

"Clip (Analysis)." *Clip (Analysis)-ArcGIS Pro | Documentation*, pro.arcgis.com/en/pro-app/latest/tool-reference/analysis/clip.htm.

"Geographic Information SYSTEMS/ONLINE MAPPING." *Geographic Information Systems/Online Mapping | City of San Angelo, TX*, www.cosatx.us/departments-services/geographic-information-systems-online-mapping.