Peter Flanders, Grant Linford, Wesley Motlow, George Owen, Seamus Sehres

Sheng

GEOG 181C

June 6, 2022

Student Activities Locator Project

Overview of project and tool:

The goal of this project was to create a student-oriented Activities Locator, curated with activities and destinations for students at UCLA. What makes UCLA a truly unique school to attend, besides the world-class academics, are the near-endless opportunities for students to eat, drink, and explore the greater Los Angeles area. Some colleges are in places with far fewer opportunities, so this group wanted to highlight what is available to UCLA students. Another motivation for this project was that it is often hard for students to figure out what destinations among the many available are actually worth their time. After all, gas is the most expensive it has ever been, and nobody wants to drive somewhere that is going to disappoint.

The main locations within this guide are areas close to UCLA or in the greater Los Angeles area, including theme parks, restaurants, museums, and other sought-after destinations that a student might want to visit. In order to develop this data set, we first started by collaborating on a list of locations that we deemed to be relevant to the goals of our project, then created a spreadsheet with X and Y columns that corresponded to the latitude and longitude locations of the destinations so that we could incorporate that information into the code once the appropriate tools were found. The latitude and longitude points for these locations were obtained through google maps. From these sources, we added locations we found to be worth going to.

After this dataset was completed we later moved on to finding appropriate arcpy scripts that would be suitable for this project. We used the Find Routes documentation on ESRI as a guide for creating the code structure.

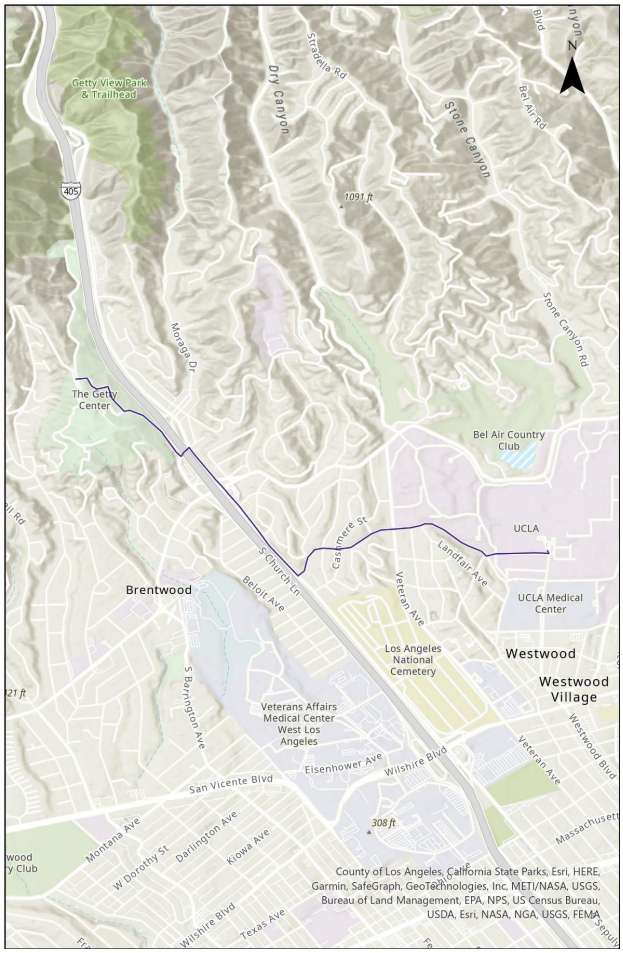
Users manual:

Using the Find Routes Tool and Understanding its Parameters:

The input parameters necessary for running the Find Routes tool are essentially two sets of coordinates, the start location, and the end location. The start coordinates in our script will default to be the Bruin Bear statue and the stop coordinates are stored in a tuple containing all of the coordinates of the restaurants and activities we compiled.

Route To: The Getty

0: Start at Location 1
1: Go southeast on Westwood Plz toward Strathmore Pl
2: Restriction: Through Traffic Prohibited
3: Make a sharp right on Strathmore Pl
4: At the traffic light, turn right on Gayley Ave
5: At the traffic light, continue forward on Montana Ave
6: At the traffic light, turn right on S Sepulveda Blvd
7: At the traffic light, turn left on N Church Ln
8: Turn right on Beverly Park Dr
9: Restriction: Avoid Gates
10: Restriction: Avoid Private Roads
11: Restriction: Through Traffic Prohibited
12: Restriction: Avoid Gates
13: Bear left
14: Restriction: Avoid Gates
15: Turn right
16: Finish at Location 2, on the left



However, the code also allows for the user to enter their own coordinates, such as the coordinates for the public affairs building on UCLA's campus. The find routes tool is used to generate some outputs between these locations such as ‘Output_routes’ and ‘Output_directions’ the two main outputs that are used to append to the mapbook and give the layout its substance. The layout

comes with the title of the destination you seek, a map of the route you will be taking to get there, and turn-by-turn directions next to the route to guide you to the destination.

Code design:

Main For-loop:

The basic structure of the code works by first looping through all of our data points then generating a route from a chosen start location to a specific destination point and then loading it into ArcGIS Pro, where it is formatted into a layout using the route and driving directions, and exported into the mapbook. There are also turn-by-turn directions added to the layout from the code design.

```
## ===== Main Loop =====
with open(activity_csv, 'r') as read_obj:
    csv_reader = reader(read_obj)
    header = next(csv_reader)
    # Check file as empty
    if header is not None:
        # Iterate over each row after the header in the csv
        for entry_row in csv_reader:
            # row variable is a list that represents a row in csv
            print(f"  Location Name is {entry_row[0]}, its type is {entry_row[3]}, and its coords are ({entry_row[1]}, {entry_row[2]})")
            tmp_coord = (float(entry_row[2]), float(entry_row[1]))

            #generate directions to this entry, and add the route to the map
            directions = add_route_to_map(stop_coords=tmp_coord)

            # add directions into the layout
            thisLayout.listElements("TEXT_ELEMENT")[0].text = directions
            thisLayout.listElements("TEXT_ELEMENT")[1].text = "Route To: " + entry_row[0]

            # export layout to pdf
            thisLayout.exportToPDF(tmp_PDF_path)

            # append pdf to final_PDF
            final_PDF.appendPages(tmp_PDF_path)
            print(f"Added {entry_row[0]} to mapbook.")
```

Earlier in the script the starting point (Bruin Bear statue) is defined as the default start point. . Then for each point in our file (the activities destinations), a feature class is created with this new destination as the endpoint, and the Bruin Bear as the start point. After the feature class is

created, we run Find Routes to get the path and directions. Then we extract directions as a workable object and save them into a list. Then, the code loads the path data into ArcGIS Pro and zooms the map extent to each specific feature class iteration. Finally, we export each map layout iteration as a temporary PDF to be added to the final mapbook PDF.

Mapbook production:

We decided the best way to visualize the locations that we would include in this project was through a mapbook like the one we created during Week 7. We drew from that week's code to start appending the outputs from the Find Routes tool like the directions and map routes into a temporary layout which was cleaned through each iteration to make space for the next destination and then appended those layouts after each iteration to a final mapbook PDF. This final mapbook PDF was compiled on top of a .aprx layout created and customized in ArcGIS Pro to give these outputs an intelligible user interface.

Problems Associated with the Code:

The initial challenge of this project was to develop the data set of locations. Each group member added locations to a common spreadsheet that they knew to be of interest to students and worth their time. Group members also made sure to filter out more common destinations like Mcdonald's or Taco Bell from their contributions to the list, as these locations do not fit with the goal of creating a unique and desirable dataset of places for students to visit. Once the locations were determined and latitude and longitude coordinates were added to correspond to the X and Y columns on the spreadsheet, the next challenge was to find a way to code this data into arcpy.

When writing the code to be used in the find routes tool, one challenge was removing and overwriting the code to draw each map. This was overcome by parameterizing each route into a function that allows for modular looping to take place and creates a new map for each route.

This was done by locating several tools available in ArcGIS Pro and consulting the code parameters available on the web page to see if they would fit with the needs of the project. While this took some effort since ArcGIS Pro is full of tools that can be used to display spatial information, the group was able to find the ideal tool called “Find Routes”. The tool gives turn-by-turn directions from one location to another, which was ideal for the purposes of this project.

Once the appropriate tool was located another challenge arose; where to begin each route? Picking a central location was necessary because the data set contained over 100 locations, and creating routes from one within the list location to another location on the list would be needlessly complicated, and unhelpful to students. To be the most applicable to student needs the group decided to make the origin of each route to be the Bruin Bear Statue that is located on campus be the default start point. The location for the Bruin Bear can be seen in line 61 of the code.

The most significant problem that this project presented was actually solved just before this group's presentation to the class. This problem was in the "parsing the output of FindRoutes_agolservices." First, the documentation of this function from ESRI was poor, so the group was confused about exactly how to use it. Second, the function outputted objects called “record_sets,” which were nowhere to be found on the ESRI Find Routes documentation. A “record_set” is similar to a feature layer, but the API prefers working with feature layers, so we had to convert the record set data into feature layers. We were able to accomplish this by experimenting with different configurations that the API has until a suitable one was found. This can be seen in lines 202-204 in the code.

This group also wanted to add changeable parameters to the code but faced challenges doing this. This is traditionally done through the use of arcpy parameters but this forces the script to be run within ArcGIS Pro. The group elected not to go down this path as it would prevent the user from continuing to use their computer while the script was building the mapbook. To get around this, the group used the argparse module from the standard python library to provide command-line arguments in our script.

Another problem associated with our project is that the find routes tool is an agol service that takes up server space at ESRI as the code computations are done for the user, which means that every time the user runs this mapbook they are subtracting credits from the 20,000 that ESRI gives you for free. After enough uses the user would be unable to run the program without any credits unless they paid for more, which can be quite costly. To work around this limitation we could implement the use of the Network Analysis toolbox which runs this function quite similarly with the shortest path functions that output distance, travel time, and turn-by-turn directions.

Conclusion:

This group sought to create a locator and mapbook that directs UCLA students from the Bruin Bear or a location of their choosing to one of many destinations in the Los Angeles area that it determined would be relevant to student interests. It began by gathering data on locations and finding their geographic coordinates, then found appropriate tools on ArgGIS Pro to process and map the data, and finally wrote python scripts to run the tool and create a series of maps based on the tool's output. There were several challenges during this process, including determining which locations to use and the appropriate tool to use to create the mapbook.

While the group was satisfied with the results of their script. The time it took to run the script and build the map book was very long, usually over twenty minutes. In the future, the group would elect to have fewer points in the data set or a quicker way to generate route pages.

The overall goal of this group was achieved and a comprehensive map book has been created which directs users from UCLA's campus to a wide variety of destinations in the Los Angeles area. Importantly, the creation of this map book involved the use of numerous skills learned and developed in Geography 181C. This group is confident that any UCLA student would benefit from using their map book, especially if they are looking to celebrate finishing their finals.