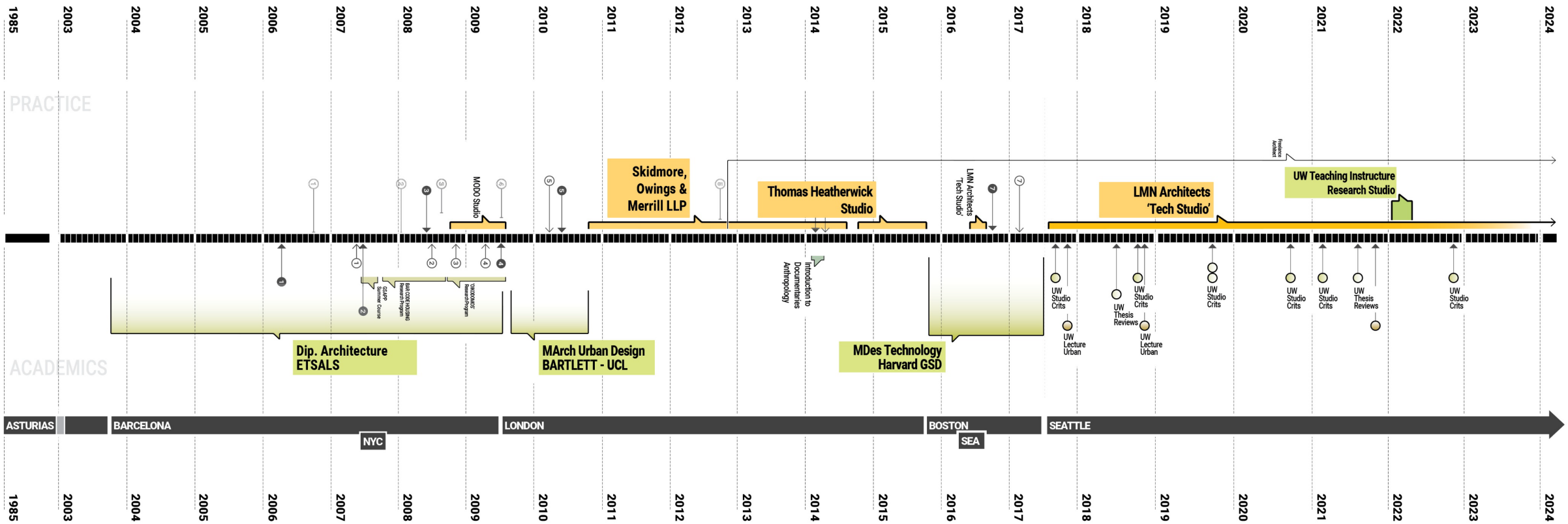
URBANLENSES

Unveiling the Unseen through Data-Driven Urban Exploration

LMNts
ENOL VALLINA | 06.23.23

¡hola!


1985
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
PRACTICE
ACADEMICS
MODO Studio
Skidmore, Owings & Merrill LLP
Thomas Heatherwick Studio
LMN Architects 'Tech Studio'
Freelance Architect
UW Teaching Instructure Research Studio
GSAPP Summer Course
BARCODE HOUSING Research Program
DMODOMUS Research Program
Introduction to Documentaries Anthropology
UW Studio Crits
UW Lecture Urban
UW Thesis Reviews
Dip. Architecture ETSALS
MArch Urban Design BARTLETT - UCL
MDes Technology Harvard GSD
ASTURIAS
BARCELONA
NYC
LONDON
BOSTON
SEA
SEATTLE

WHAT'S THE PLAN?

01 **Part One: "Understanding Urban Complexity" (1h30min)**

+ **Context:** "Decoding Urban Systems" (45min)
+ **Showcase:** "Uncovering the City" (45min)
    a. Workflow and Tools
    b. Leveraging APIs

02 **Part Two: "Visualizing Urban Narratives" (1h)**

+ **Context:** "Data Visualization and Urban Intervention" (30min)

(Break 10-15min)

+ **Showcase: Visualizing Data with QGIS (30 min)**
    a. Activity Point Analysis
    b. Route Analysis

03 **Part Three: "Translating Urban Data into Design" (45min)**

+ **Showcase:** Visualization and Interaction with Rhino and Grasshopper
    a. Activity Evolution
    b. Custom Shortest Path Analysis

04 **Part Four: Future Directions and Open Discussion (30min)**

+ **Next Steps**
+ **Open Discussion**

GITHUB REPOSITORY

https://github.com/lmnts/urbanlenses

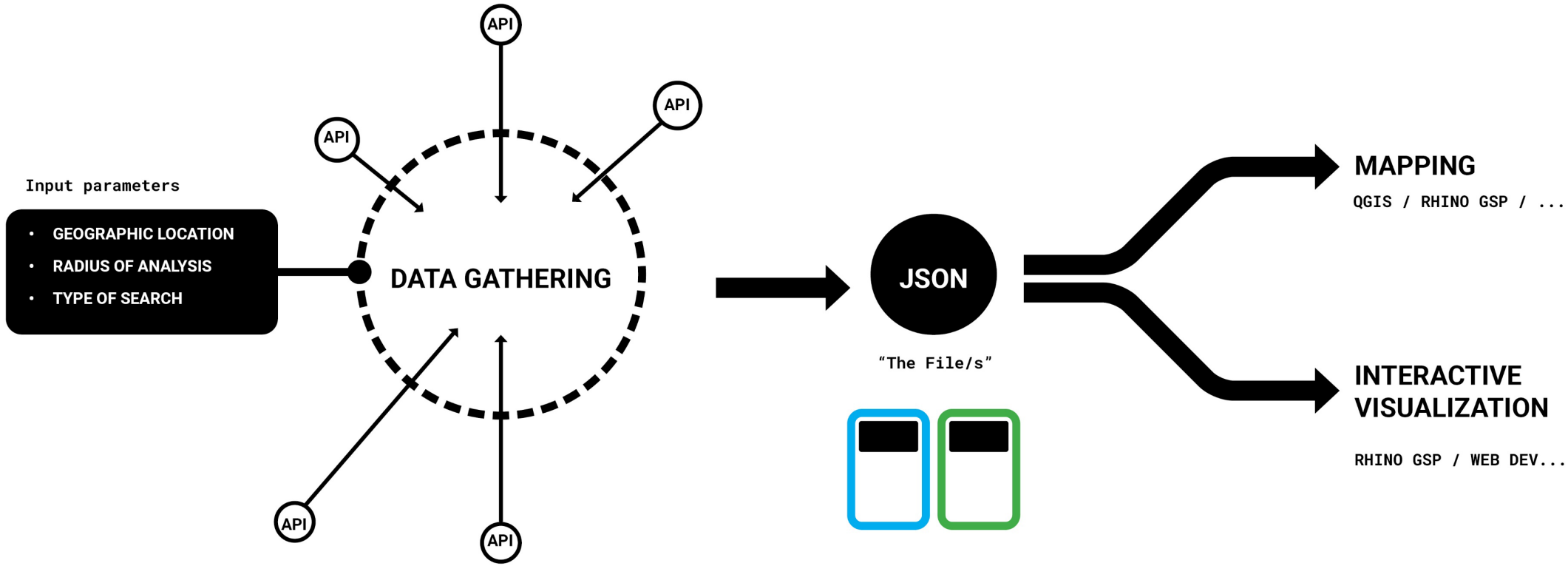
WORKFLOWS

06/23/23

-

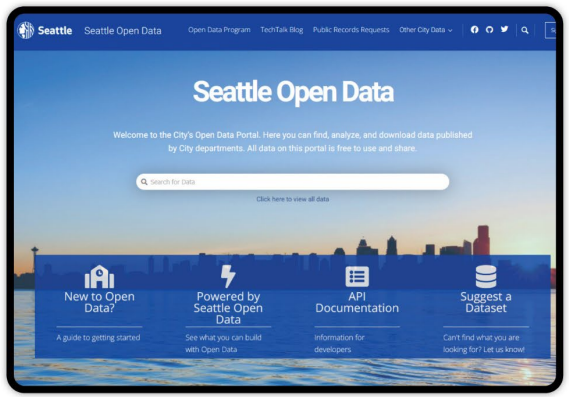
ENOL VALLINA

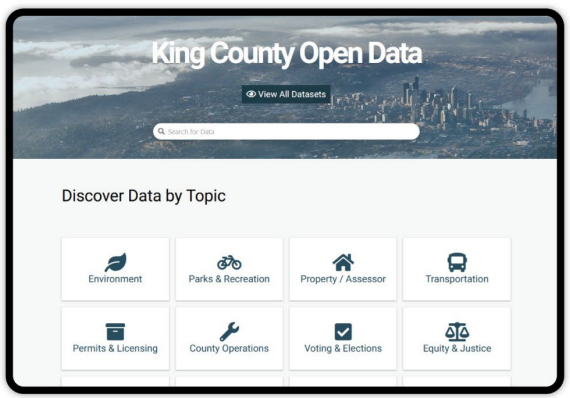
THE BASIC WORKFLOW



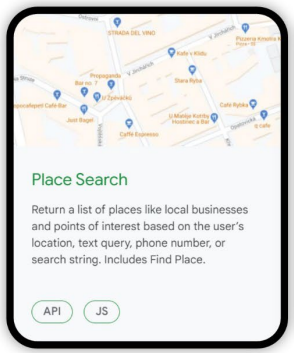
WHERE DOES THE DATA COME FROM?

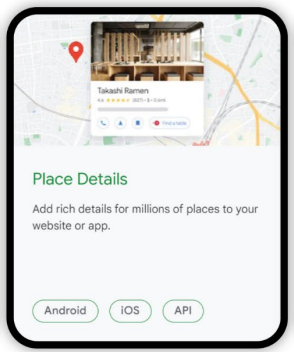
CITY'S OPEN DATA

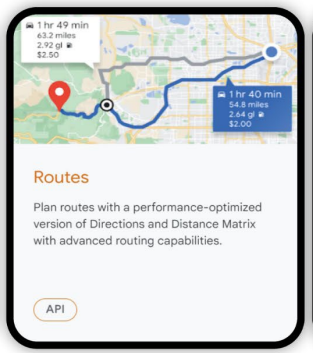


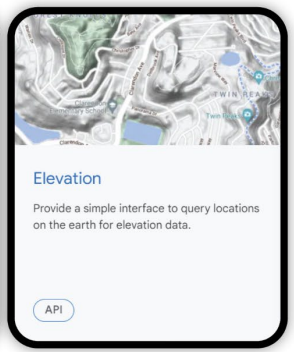


GOOGLE MAPS API



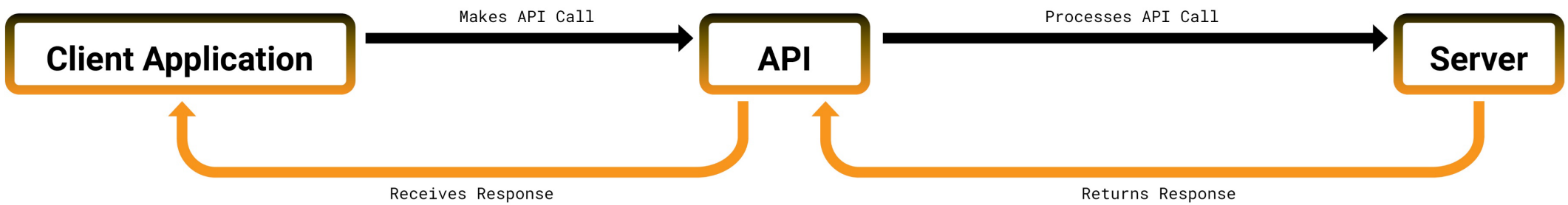






Application Programming Interface

An API is a “bridge for software communication”, enabling different software applications to interact and share data with each other.



LEVERAGING THE GOOGLE API

01. Google Cloud Project: Create or select a project in the Google Cloud Console.

02. Enable Billing: Enable billing for your project. Google's APIs are not entirely free, and they require billing information even if the usage falls within the free tier.

03. Enable APIs: Enable the Google Maps APIs that you want to use.

04. Get API Key: Create credentials for your project to get the API Key.

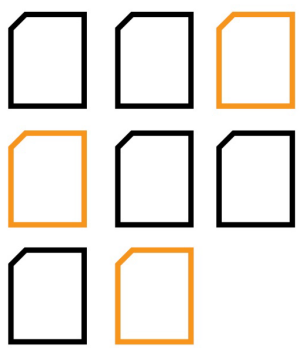
05. Restrict API Key: Restrict the API Key to prevent unauthorized usage and quota theft.

06. Use API Key: Use the API Key in your application.

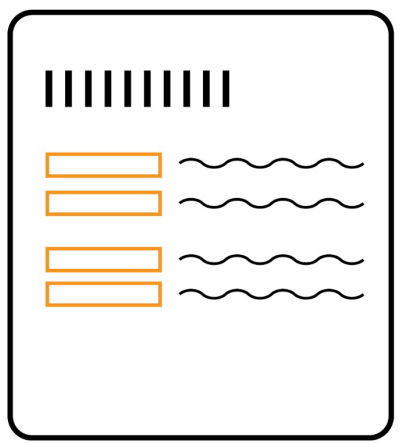
LEVERAGING THE GOOGLE API

SET UP → INTERFACE → LOCATIONS → ROUTES → OUTPUT

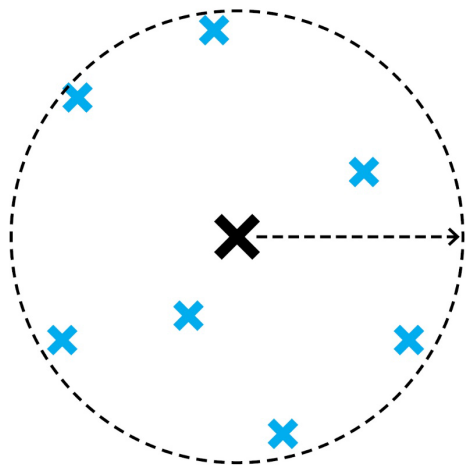
Python libraries and API prerequisites neccesary to run the tool



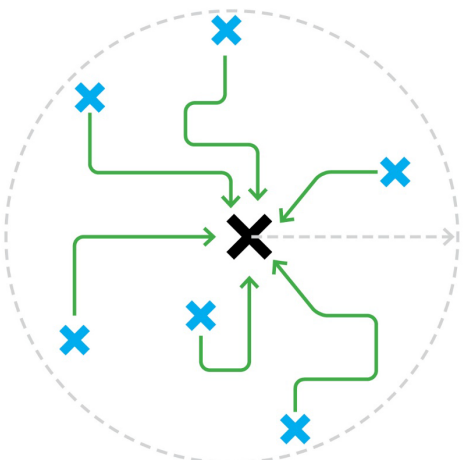
User Interface where to set up the type of search.



API requests that gather information about activities around an urban area of analysis.



API requests that gather information about all the most likely route from every location found earlier to the location of the analysis



LEVERAGING THE GOOGLE API

SET UP → INTERFACE → LOCATIONS → ROUTES → OUTPUT

PYTHON VO

PYTHON 3.0

PYTHON LIBS

```
import requests
import json

import csv
import pandas as pd

import time
import sys
import os
import random
from datetime import datetime
from datetime import timedelta
from time import sleep


import tkinter
import PySimpleGUI
```

AUTHENTICATION

Google API Key

LEVERAGING THE GOOGLE API

SET UP → INTERFACE → LOCATIONS → ROUTES → OUTPUT

PYTHON VO

PYTHON 3.0

PYTHON LIBS

```
import requests
import json

import csv
import pandas as pd

import time
import sys
import os
import random
from datetime import datetime
from datetime import timedelta
from time import sleep


import tkinter
import PySimpleGUI
```

AUTHENTICATION

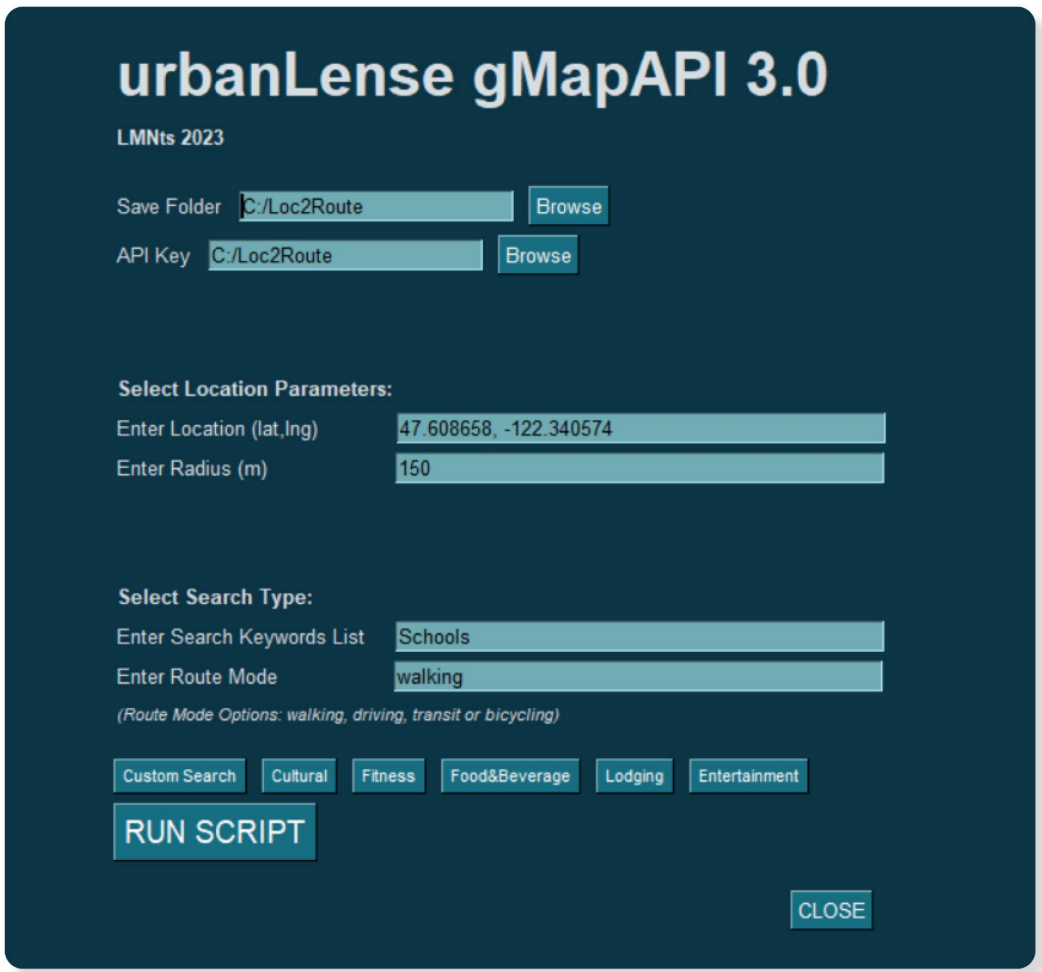
Google API Key

01. INTERFACE SET UP

02. LOCATE THE API KEY

03. DEFINING THE INPUT PARAMS.

- LOCATION & RADIUS
- PRESET ACTIVITY SEARCHES /CUSTOM
- ROUTE MODE



LEVERAGING THE GOOGLE API

SET UP → INTERFACE → LOCATIONS → ROUTES → OUTPUT

PYTHON VO

PYTHON 3.0

PYTHON LIBS

```
import requests
import json

import csv
import pandas as pd

import time
import sys
import os
import random
from datetime import datetime
from datetime import timedelta
from time import sleep


import tkinter
import PySimpleGUI
```

AUTHENTICATION

Google API Key

01. INTERFACE SET UP

02. LOCATE THE API KEY

03. DEFINING THE INPUT PARAMS.

- LOCATION & RADIUS
- PRESET ACTIVITY SEARCHES /CUSTOM
- ROUTE MODE

01. HANDLING THE DATA RETRIEVED FROM API

- DUPLICATE RESULTS
- DEALING WITH MULTIPLE PAGES

02. MAKING THE REQUEST PER KEYWORD (GOOGLE PLACE SEARCH)

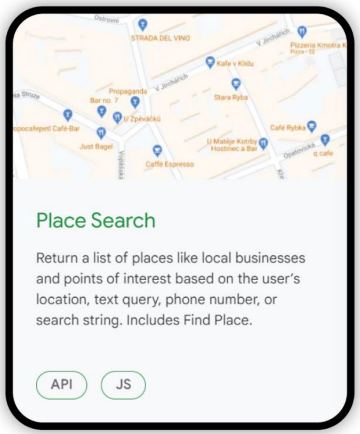
- DEFINING PRE-STABLISHED PARAMS.

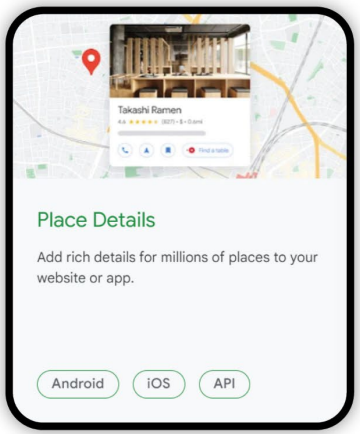
03. SELECTING THE RELEVANT RESULTS

- DIRECT RESULTS
- MANAGING OTHER RESULTS: USING '**PLACE DETAILS**' TO GET ADDITIONAL INFORMATION ABOUT OPENING TIMES.

04. HANDLING DATA

- TRANSFORMING RESULTS INTO AN OUTPUT JSON FILE.
- SAVING FILE WITH UNIQUE NAMING SYSTEM.





LEVERAGING THE GOOGLE API

SET UP

PYTHON VO

PYTHON 3.0

PYTHON LIBS

INTERFACE

01. INTERFACE SET UP

02. LOCATE THE API KEY

03. DEFINING THE INPUT PARAMS.

- LOCATION & RADIUS
- PRESET ACTIVITY SEARCHES /CUSTOM
- ROUTE MODE

LOCATIONS

01. HANDLING THE DATA RETRIEVED FROM API

- DUPLICATE RESULTS
- DEALING WITH MULTIPLE PAGES

02. MAKING THE REQUEST PER KEYWORD (GOOGLE PLACE SEARCH)

- DEFINING PRE-STABLISHED PARAMS.

03. SELECTING THE RELEVANT RESULTS

- DIRECT RESULTS
- MANAGING OTHER RESULTS: USING '**PLACE DETAILS**' TO GET ADDITIONAL INFORMATION ABOUT OPENING TIMES.

04. HANDLING DATA

- TRANSFORMING RESULTS INTO AN OUTPUT JSON FILE.
- SAVING FILE WITH UNIQUE NAMING SYSTEM.

ROUTES

01. MAKING THE REQUEST PER LOCATION FOUND IN PREVIOUS STEP.

- DEFINING PRE-STABLISHED PARAMS.

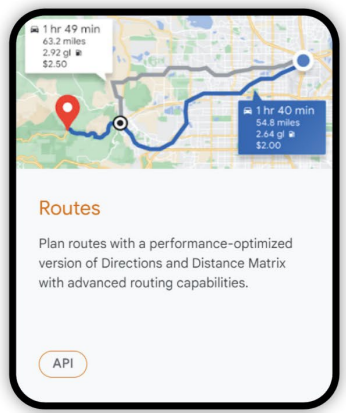
02. SELECTING THE RELEVANT RESULTS

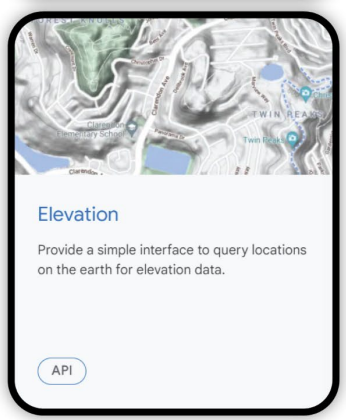
- DIRECT RESULTS: COLLECTION OF POINTS THAT REPRESENT EVERY STEP OF THE ROUTE GIVEN THE PARAMETERS.
- MANAGING OTHER RESULTS: USING '**ELEVATIONS API**' TO ADD THAT INFORMATION ON EVERY STEP OF THE ROUTE.

04. HANDLING DATA

- TRANSFORMING RESULTS INTO AN OUTPUT JSON FILE.
- SAVING FILE WITH UNIQUE NAMING SYSTEM.

OUTPUT





QGIS ACTIVITIES ANALYSIS

LOCATIONS

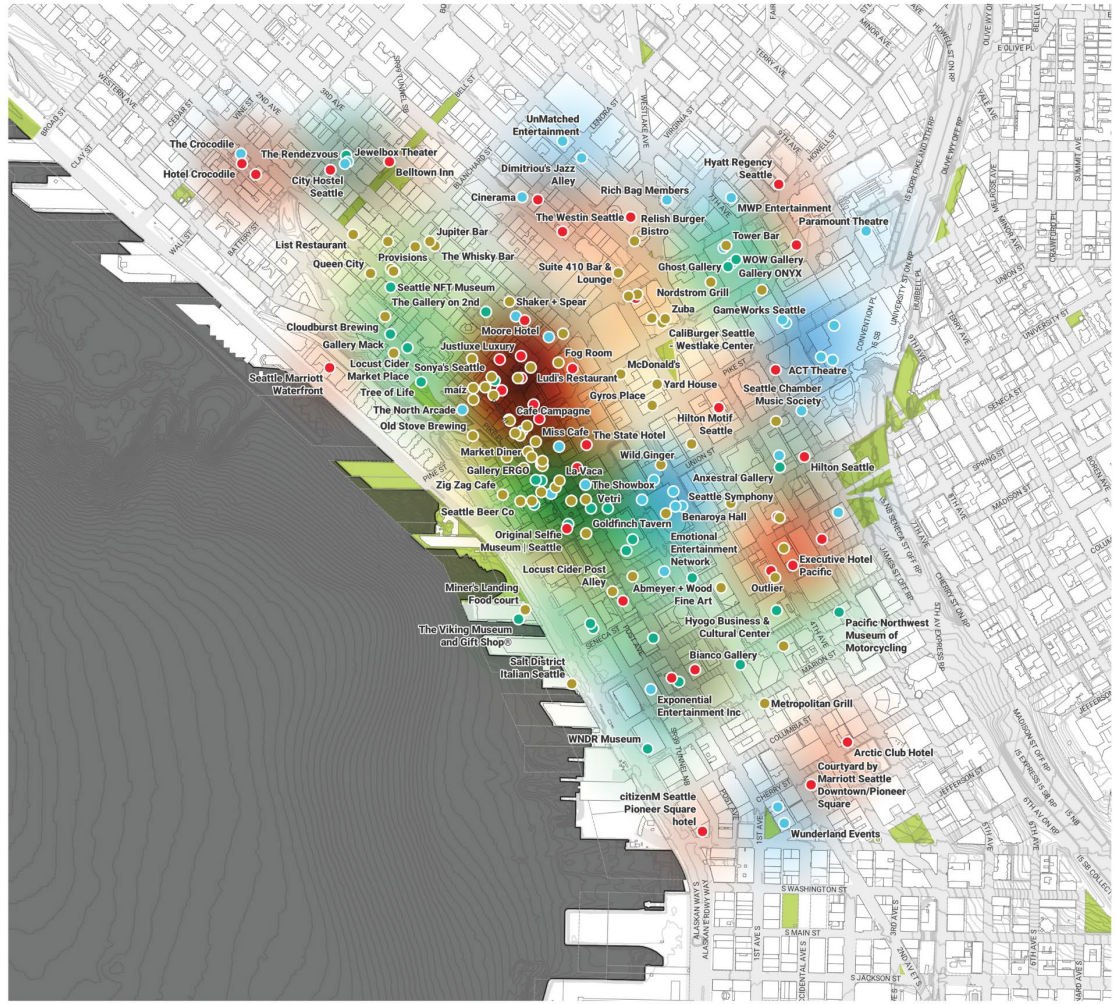
JSON

DATA

+ ACTIVITY PTS
+ NAME
+ TYPE

CALCULATIONS

+ POINT INTERPOLATION
+ HEATMAP BY AGLOMERATION



VISUALIZATION

QGIS ROUTES BY ACTIVITY

ROUTES

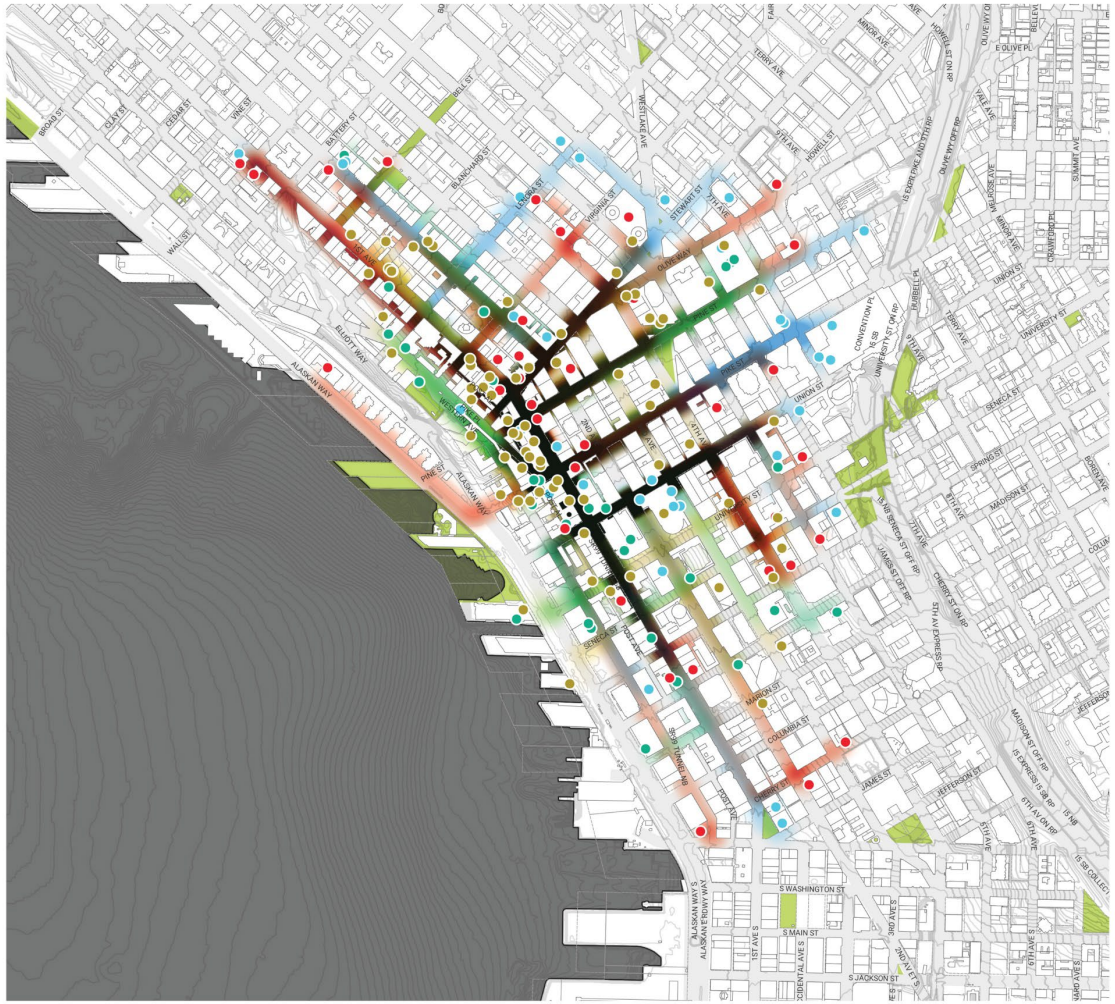
JSON

DATA

+ ROUTE STEP POINTS
+ ACTIVITY

CALCULATIONS

+ POINT INTERPOLATION
+ HEATMAP BY AGLOMERATION



VISUALIZATION

QGIS ROUTES SLOPES

ROUTES

JSON

+ ROUTE STEP POINTS
+ SLOPES AVERAGE
+ SCORES CHANGE

DATA

CALCULATIONS

VISUALIZATION

INTERACTIVITY WITH GRASSHOPPER: PATHFINDERS

LOCATIONS

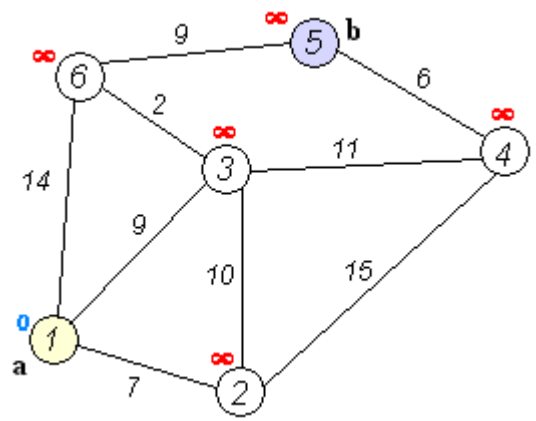
JSON

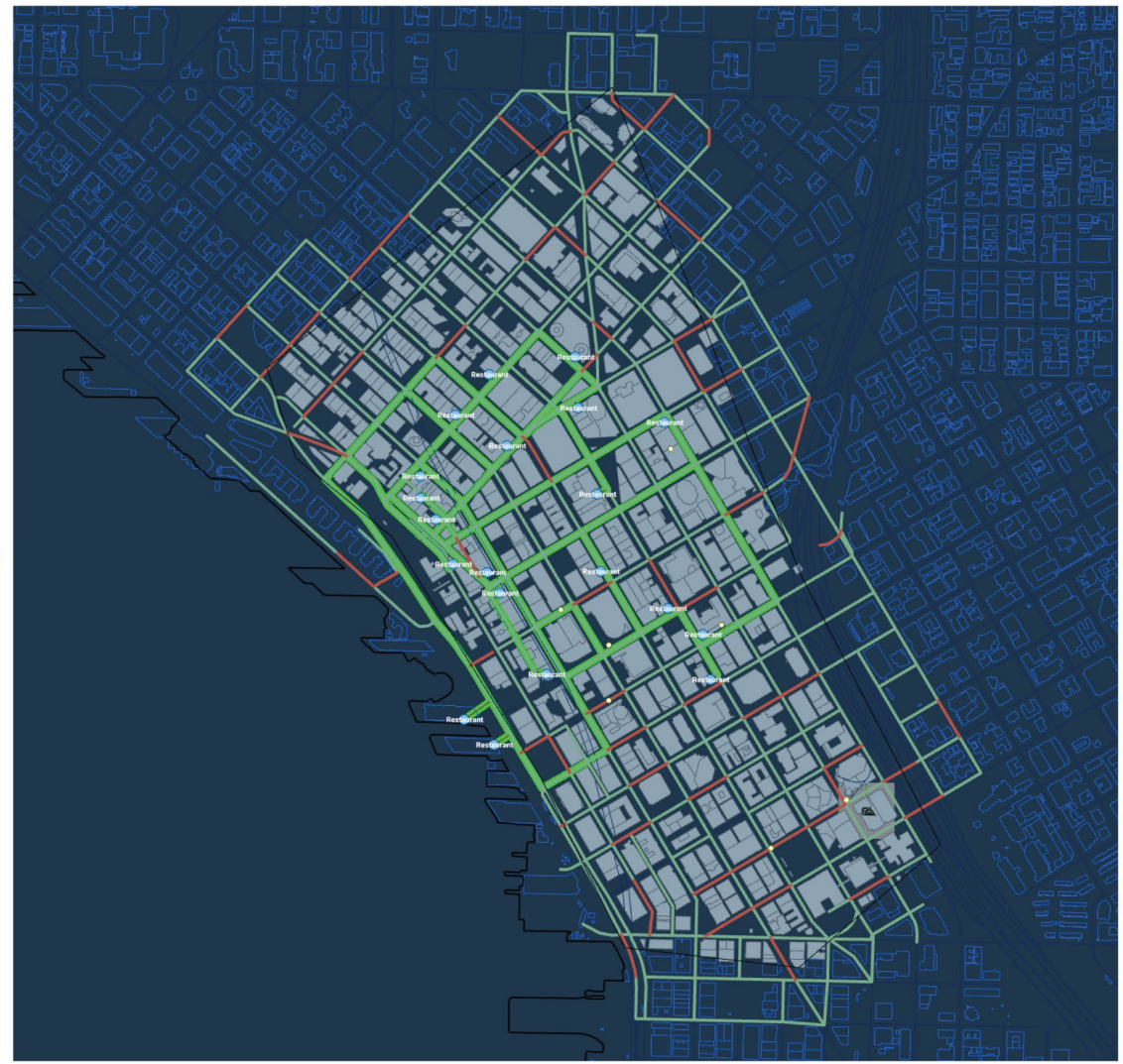
DATA

+ ACTIVITY PTS
+ NAME
+ HOURS OF OPERATION
+ BASE CITY MAP BUILD

CALCULATIONS

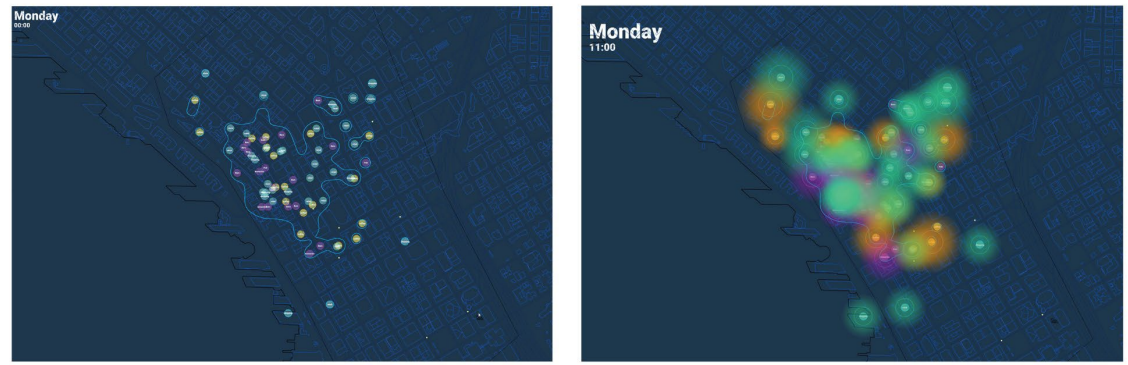
+ SHORTEST PATH USING DIJKSTRA ALGORITHM
+ BUILD POSSIBLE ROUTE COMBINATIONS
+ PARSE THE SLOPE OF STREET NETWORK SECTIONS





VISUALIZATION

INTERACTIVITY WITH GRASSHOPPER: ACTIVITY ANIMATION



LOCATIONS

JSON

+ ACTIVITY PTS
+ NAME
+ HOURS OF OPERATION
+ BASE CITY MAP BUILD

DATA

+ POINT INTERPOLATION
+ HEATMAP BY AGLOMERATION
+ FILTERING BY INPUT DATE/TIME

CALCULATIONS

VISUALIZATION

NEXT STEPS

- INTEGRATION OF SODA API TO DIRECTLY RETRIEVE OPEN DATA FROM CITY

- HOW CAN WE HARVEST DATA ABOUT THE ACTIVITY WITHIN THE CITY?

- GOOGLE API STREET VIEW AND IMAGE ANALYSIS

- INFERENCE OF SOCIAL MEDIA AND OTHER VIRTUAL PLATFORMS IN THE URBAN ENVIRONMENT.

QUESTIONS FOR DEBATE

- THE QUESTION ABOUT DEVELOPMENT USING CHAT GPT AS AN AID. WHAT DOES IT MEAN FOR THE WAY WE DEVELOP OUR TOOLS?

- ARE THERE WAYS TO MEASURE HOW DO WE EXPERIENCE THE STREET? HOW THESE TYPE OF VISUALIZATION OUTCOMES CAN INFLUENCE THAT EXPERIENCE?

Gracias!

https://github.com/lmnts/urbanlenses