Xiyu Fan
CS 5001
Final Project Design Document

I. Basic Design Concept

My project aims at analyzing the park facility density in each park and the number and type of different park facilities in each neighbourhood. It's a simple program that the general public may use to decide which neighbourhood they want to move into depending on their outdoor leisure needs, and it might be helpful for urban planners to assess the current development of park facilities in different parks and neighbourhoods area and to decide how to maintain them to meet the public's recreation needs. I use two databases from the City of Vancouver Public Data, the Parks data, and the Park facilities data. And their QRLs are at the end of this profile.

II. Classes Design

Correspondingly, I increase two classes named Park and Facility in model_1_park_class.py and model_2_facilities_class.py. The Park class is a module that provides functionalities to manage a park model in Vancouver City. It is responsible for reading the name of a park, the name of the neighbourhood that this park is located, and its area. Its main features include expanding the park's area and decreasing the park's area. The Facility class is a module that provides functionalities to manage a park facility model in Vancouver City. The main features of the park facility module include increasing and decreasing the facility count. It is responsible for reading the name of the park that this facility belongs to, the facility type, and its count. Its main features include increasing and decreasing the facility count.

III. Data Fetching

To fetch and clean the data, I design three functions for each data source separately and one for cleaning these data together in fetch_data.py. Firstly, to fetch data from the URLs, I design get_parks_csv() and get_facilities_csv(), using the requests.get() method to fetch data into text. Then, by using get_needed_park_data() and get_needed_facility_data(), I try to get the data I needed from the two data resources that contain various useless data for my program, followed by encapsulating these data into two lists of instances by create_park_instances() and create_facility_instances(). Lastly, while I noticed that some of the park hectare data from the data resource representing as zero, which is unreasonable and would cause problems when I analyze the park facility density later, I design remove_invalid_park_instances() to remove invalid park instances whose hectare is zero and their correspond facility instances from the lists created before.

IV. Data Structure

As for the usage of data structures, I mainly use lists and dictionaries in model_3_parks_and_facilities_analysis_functions.py.

By create_neighbourhoods_park_facilities_dictionary(), I try to create a dictionary whose key represents neighbourhood names and value represents park facility details, representing like {'Neighbourhood1': [['Park1', 'Facility1', 1]], 'Neighbourhood2': [['Park2', 'Facility2', 2]]}. By create_individual_facility_distribution_dictionary() and create_full_facility_distribution_dictiona

ry(), I create a dictionary whose keys are the neighbourhood names and values are the distribution of facility types within the neighbourhoods, representing like {'Neighbourhood1': {'F1': [3, 75.0%], 'F2': [1, 25.0%]}}, 'Neighbourhood2'...}. Lastly, to get the string representation of the facility number and distribution of each neighbourhood in order to display them easier, I design the get_individual_facility_distribution() and get_facility_distribution_sttring_dictionary(), representing like {'Neighbourhood1': '3 F1(75.0%), 1 F2(25.0%)\n'} .

As former functions are used to store data for analyzing the number and type of different park facilities in each neighbourhood, I also design two functions for analyzing the park facility density in each park. By calculate_the_facility_density(), the facility density of a park can be calculated. By create_facility_density_dictionary(), the program can calculate the park facility density for every park with facilities in the park instances list and store them in a dictionary whose keys are park names and values are their facility density.

V. Visualization

As for visualization, I plan to use the histogram to show the park facility density in each park and use the pie chart to show the number and type of different park facilities in each neighbourhood. The user can choose to check the density graph either in an increasing order or a decreasing order and check the number and type of all neighbourhoods by histogram too.

VI. Interaction

The user can impact the data analysis mainly in two ways, changing the park data and changing the park facility data. For the first one, the user can add a new park, delete an existing park, increase the area of a park, and decrease the area of a park. For the second one, the user can add a new facility, delete an existing facility, increase the number of a facility, and decrease the number of a facility. All of them can impact the analysis.

QRLs:

Park:

https://opendata.vancouver.ca/explore/dataset/parks/table/?dataChart=eyJxdWVyaWVzIjpbeyJjaGFydHMiOlt7InR5cGUiOiJjb2x1bW4iLCJmdW5jIjoiU1VNIiwieUF4aXMiOiJoZWN0YXJlIiwic2NpZW50aWZpY0Rpc3BsYXkiOnRydWUsImNvbG9yIjoiIzAyNzlCMSJ9XSwieEF4aXMiOiJuZWlnaGJvdXJob29kbmFtZSIsIm1heHBvaW50cyI6NTAsInNvcnQiOiIiLCJzZXJpZXNCcmVha2Rvd25UaW1lc2NhbGUiOiIiLCJjb25maWciOnsiZGF0YXNldCI6InBhcmtzIiwib3B0aW9ucyI6e319fV0sInRpbWVzY2FsZSI6IiIsImRpc3BsYXlMZWdlbmQiOnRydWUsImFsaWduTW9udGgiOnRydWV9&location=11,49.21234,-123.33475

Park facility:

https://opendata.vancouver.ca/explore/dataset/parks-facilities/table/?disjunctive.facilitytype&dataChart=eyJxdWVyaWVzIjpbeyJjb25maWciOnsiZGF0YXNldCI6InBhcmtzLWZhY2lsaXRpZXMiLCJvcHRpb25zIjp7ImRpc2p1bmN0aXZlLmZhY2lsaXR5dHlwZSI6dHJ1ZX19LCJjaGFydHMiOlt7ImFsaWduTW9udGgiOnRydWUsInR5cGUiOiJjb2x1bW4iLCJmdW5jIjoiQVZHIiwieUF4aXMiOiJwYXJraWQiLCJzY2llbnRpZmljRGlzcGxheSI6dHJ1ZSwiY29sb3IiOiIjMDI3OUIxIn1dLCJ4QXhpcyI6ImZhY2lsaXR5dHlwZSIsIm1heHBvaW50cyI6NTAsInNvcnQiOiIifV0sInRpbWVzY2FsZSI6IiIsImRpc3BsYXlMZWdlbmQiOnRydWUsImFsaWduTW9udGgiOnRydWV9