

Sandeep Singh

S2093748

Chapter 1

Software Architecture Design

1.1 Overview

This section would explain the reasoning for all the classes that I used.

1.2 App

App.java is basically the main class of the program and it basically starts everything off. In App.java we have command line arguments which are the date going to be used, and the URL to get data from the REST server. In App.java we would also print out the 3 different json files for each date.

1.3 Direction

Direction.java basically just has all of the 16 directions for the drone to move, and it is used in the other classes

1.4 InvalidPizzaCombinationException

InvalidPizzaCombinationException basically is used in Order.java in the function getDeliveryCost when there's an exception for the pizza's ordered

1.5 Map

Map.java basically generates the geojson for the flightpath of the drone, it takes the coordinates moved by the drone and stores it

1.6 NoFlyZones

NoFlyZones.java gets all the no-fly zones coordinates from the REST server

1.7 OrderOutcome

OrderOutcome.java basically has all of the outcome of the orders possible and is used in the other classes

1.8 LngLat

LngLat.java basically deals with all of the latitude and longitude of the drone, the noflyzones and the restaurants

1.9 Order

Order.java basically deals everything related to the Orders for the drone, it gets the orders from the restaurants on a specific day, checks if the orders are valid and moves the drone to specific restaurants due to the orders.

2.0 Drone

Drone.java does everything when flying the drone it basically moves the drone to the restaurant and back and it also checks on the noflyzones while flying

2.1 Restaurant

Restaurant.java basically deals everything with the restaurants so it checks on the name of the restaurant and also the positions of the restaurant for the drone to fly

2.2 Menu

Menu.java basically deals with the pizza menu in each of the restaurant so it saves the name of the pizza as well as the price of the pizza.

Chapter 2

Class Documentation

2.1 App

App.java basically runs the whole program and starts everything off.

2.1.1 Method: public static void main(String[] args)

The program will first get an input date from the first parameter of args, and it will also get data from the REST server for that date using the second parameter of args, and the data would have all of the orders for that date and all of the restaurants . I also have multiple for loops in the file with a HashMap inside. I'm going to explain the first for loop, in the first for loop we are making a JSON file for deliveries in a specific day so I put a HashMap in the for loop so it will contain all of the data of the orderNo, costInPence and the outcome for each order and in the for loop I also initialise a JSON object so the program would know to make a new JSON object after each iteration and I store it all in a JSON array List.

At the end of the file, we use FileWriter, the FileWriter function basically makes a json and geojson file for all the JSON values stored in the array list

The algorithm of the drone to do everything basically works by:

First, we have Order.java, and in the function order Outcome, this would first get data from the REST server and it will check if the order is valid or is there some mistakes with the card number and so on. If there is a mistake in the program, then the value for orderOutcome would change, but if there isn't I then stored all the valid orders in another arrayList called second_yo. After that, I then checked for the orders inside the arrayList called second_yo what kind of orders it has. This is because I wanted to maximise the number of deliveries I can do in a day, so I put the orders with a very far restaurant one of the last few elements in the arrayList called third_yo and the orders with the closest restaurant would be the first few orders in the arrayList called third_yo

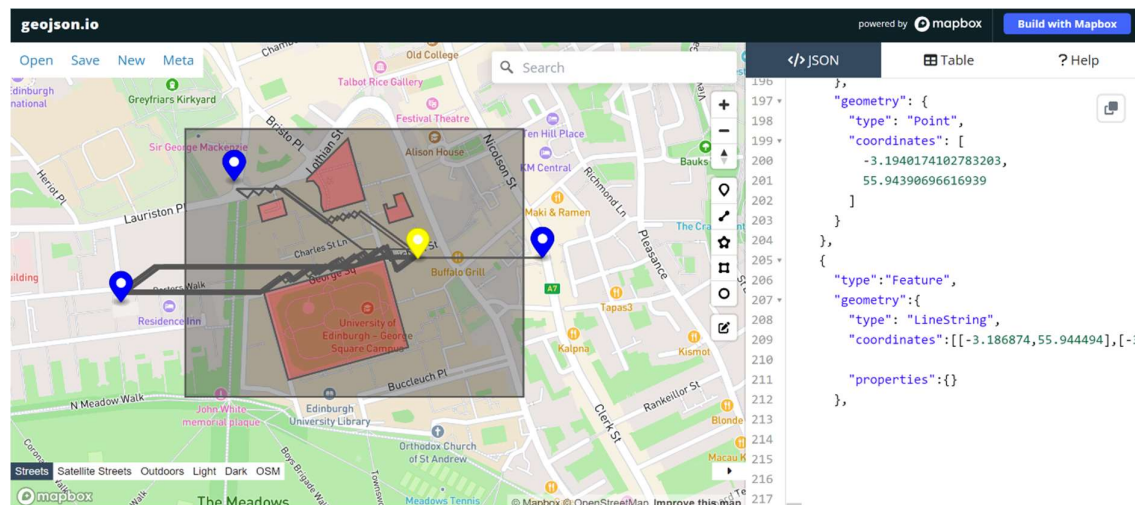
In the function hi() in Order.java I looped all of the orders in third_yo arrayList and it would move the drone with the orders which are closet to Appleton Tower first and I'm counting each move by the variable called drone.moves which is basically from the class Drone and it increments every time, the drone makes a move, and there's a variable called nams which increments every time an order has been done

In Drone.java in the function called direction (LngLat destination, LngLat current) it makes a while loop and it only breaks the while loop when the drone is close to the destination

which means it is a distance less than 0.00015. In that class I basically have an arrayList called anglez, which stores all of the drone angles and I also have an arrayList called positions as well as an arrayList called new_positions and these 2 arrayList basically stores the drones positions and I also have a variable called times which basically calculate the time it takes for each drone movement and I made the drone movement to become a line and the line would be able to tell me when the drone movement is about to intersect the no Fly zone and I have another function which tells the program what to do if it is about to intersect the noFlyZone. In the function direction it works by moving the drone to a specific longitude and latitude depending on the drone's position now and the final destination and it moves the drone by using the function called nextPosition in LngLat.java

I have a function called as helper() which basically stores all of the coordinates of the noflyzones before the program runs in Drone.java. I also have another function called getNoFly() which gets called in the function direction when the drone is about to intersect a noflyzone and the function basically just return false when the distance of the drone to the no fly zone is close and in the function direction, I would then change the position of the drone to avoid the noFlyZone and in the function called direction there's a variable called moves which basically increments after each movement of the drone and that links to Order class in the function called hi() which checks if there is still enough movements to make it to and from the restaurant with some extra buffer due to the noFlyzone if the 2000 moves is exceeded, the drone would break the loop and stop delivering.

Here are some pictures of the drone movement on 2023-01-01:



Here's a picture from 2023-01-02:

geojson.io

powered by mapbox

Build with Mapbox

Open Save New Meta

Search

</> JSON Table ? Help

Map of Edinburgh, Scotland, showing the University of Edinburgh George Square Campus and surrounding streets. A red polygon highlights the campus area. A black line connects four blue location pins: one on the left, one on the top left, one on the top right, and one on the right. A yellow pin is also visible on the right side of the campus. The map includes labels for streets like Lauriston Pl, George St, Buccleuch Pl, and landmarks like the University of Edinburgh and Edinburgh University Library.

196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217

[-3.187157, 55.9445132], [-3.187007, 55.9445132]] },