

Project Description

Restaurant Delivery

Problem Description

The system will receive a list of orders as input. This list represents the scenario to be simulated. For each order, the system will receive the following information:

- Arrival Time Stamp: When the order was made.
- Order Type: There are 5 types of orders: Charity orders, VIP Frozen orders, VIP orders, Frozen orders and Normal orders.
 - Charity orders are serviced first using available normal motorcycles only
 - VIP Frozen orders are serviced before normal VIP orders using the available Frozen Motorcycles.
 - VIP orders must be serviced before frozen and normal orders.
 - Frozen orders are the orders that needs to be delivered using motorcycles with refrigerators (Frozen Motorcycles) to keep its temperature low.
 - Normal orders are the orders that neither VIP nor Charity and needn't to be serviced by Frozen Motorcycles.
- Order Region: the restaurant has 4 branches. Each branch is in a different region. The Order Region indicates the region of the order and hence the branch that should deliver the order to the customer.
- Order Distance: the distance between the restaurant and the order location (in meters).
- Order Money: the total order money the customer should pay. At startup, the system loads (from a file) information about the available motorcycles. For each motorcycle, the system will load the following information:
- Motorcycle Type: There are 3 types: Fast motorcycles, Frozen motorcycles and Normal motorcycles. • Fast motorcycles are motorcycles with higher speed level. • Frozen motorcycles are the motorcycles that have small refrigerator to save food in. • Normal motorcycles are the motorcycles that neither Fast nor Frozen.
- Motorcycle Region: the branch or the region of this motorcycle (from which the motorcycle starts its delivery and to which it returns after delivery).
- Note: The Motorcycle speed (the number of meters it can move in one timestep) is the same for all motorcycles of the same type.

Program Interface

The program can run in one of three modes: interactive, step-by-step or silent mode. When the program runs, it should ask the user to select the program mode. Interactive mode allow user to monitor the orders waiting in each region. VIP orders are printed in red while frozen ones are printed in violet and normal are printed in dark blue. At each time step, program should provide output similar to that in the following figure. In this mode, program pauses for a user mouse click to display the output of the next time step. At the bottom of the screen, the following information should be printed:

- Simulation Timestep Number
- For each region, print: [Note that the following information is for each region.] ▪ Number of active(waiting) orders of each order type ▪ Number of available motorcycles of each type ▪ Type & ID for ALL motorcycles and orders that were assigned in the last timestep. e.g. N6(V3) → normal motorcycle#6 assigned VIP order#3 ▪ Total number of orders served so far of each order type

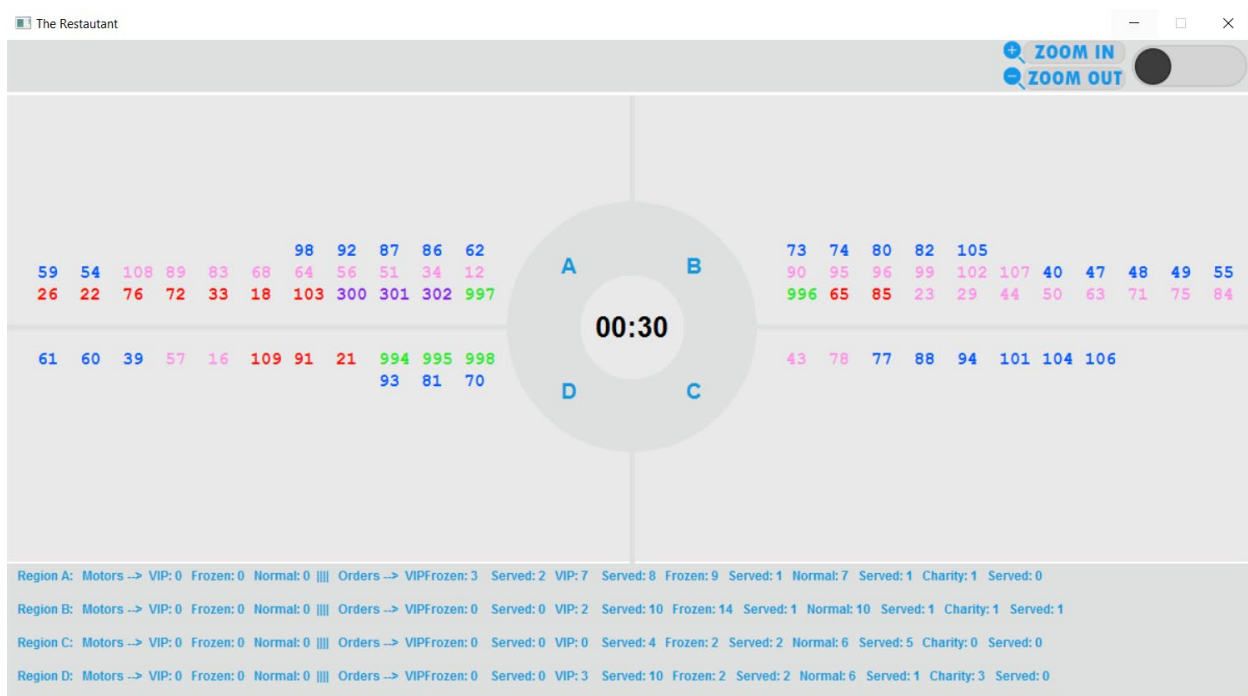


Figure 1: Light Mode

Step-by-step mode is identical to interactive mode except that each time step, the program waits for one second (not for mouse click) then resumes automatically.

In silent mode, the program produces only an output file. It does not draw anything graphically. No matter what mode of operation your program is running in, the output file should be produced.

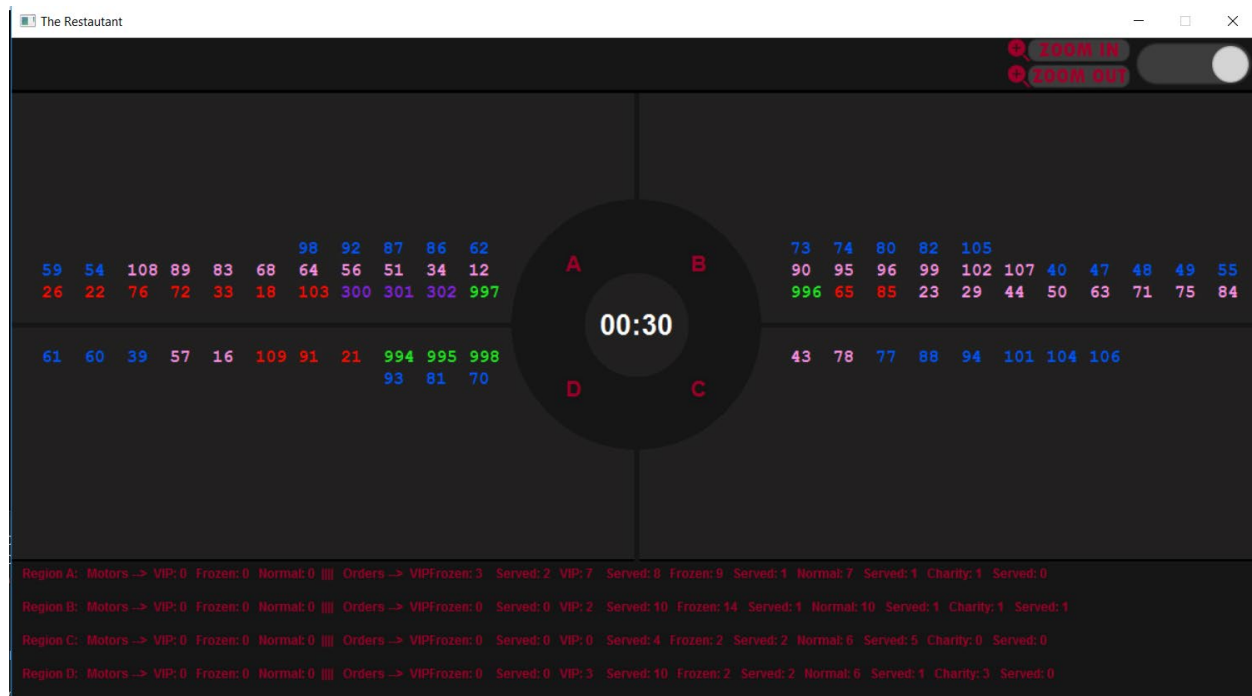


Figure 2: Dark Mode

Data Structures:

Priority Queue: Implemented as a Heap. Used for getting max Priority elements in lists used.

List: Implemented as Linked List. Used to manipulate the elements freely without changing their order.

Queue: Implemented as Linked List. Used for data entering and exiting with their respective order.

Important Classes:

Restaurant: The Main Class of the Project. It controls the flow of other classes and coordinates their functionality.

Region: It has all the important Lists of Orders and Motorcycles. It is linked with the restaurant class by Composition as each Restaurant has four Regions.

Order: It stores the information of the objects of orders.

Event: It stores the information of the objects of Events that are read from the input file.

Motorcycle: It stores the information of the objects of Motorcycles.

Important Functions:

Run Simulation (Member function of Class Restaurant): It prompts the user for the input file that the simulation will be done on, and lets him choose one of the three Modes for Simulation as well as turning on/off the music and the music file of their choice, and the Interface Mode (Dark & Light).

Simulation (Member function of Class Restaurant): It is the Organizer of the Whole simulation as it starts calling the other functions that in turn start doing their functionality in the Simulation.

Assign Orders: Assigns the orders to the Motorcycles and sets their finish time.

Serving orders: Moves the Orders to the finished List to be printed on the time step that they finished on.

Arrived Motors: Makes the motors that finished Delivering their orders free again for use.

Draw: The main function that Organizes all the Interface drawn.