Project 4

Drive-Thru Window

Introduction

Drive-thru banks, grocery stores, and other merchants typically have several check-out lanes available, but often many of them are closed because of a low volume of customers checking out. Because the customer flow varies, sometimes additional lanes are opened and sometimes opened ones are closed.

In this project, we will create a simulation of the customer flow through several check-out lanes. The customers will arrive at random and depart at random. If at any time all the lanes are backed up too much, then we will open a new lane. On the other hand, if one lane sits idle for too long, then we will close that lane.

The Arrival of New Customers

One simulation parameter will be the arrival rate a of new customers. Its value represents the average time, in seconds, between arrivals. For example, if a new customer arrives, on average, every two minutes, then the parameter's value will be 120 (seconds).

Simulating such arrivals is quite simple. Our model will update the simulation once each second. On each update, it will check to see whether there is a new customer arriving. On each update, there is probability $1/a$ that there will be an arrival. To decide whether there was an arrival, we will generate a random number between 0 and 1. If the random number is between 0 and $1/a$ (which has probability $1/a$ of occurring), then we will create a new customer. Otherwise, there was no new arrival on that update.

Typically, lines of waiting customers will develop in the different lanes. When a new customer arrives, he will join the shortest line available. (If all lanes are "full," then a new lane will opened. More on that later.)

The Departure of Current Customers

Another simulation parameter will be the service time s of a customer. That is the time that the customer spends at the check-out counter having his bill tallied and paying it. That,

too, will be random. On each simulation update and for each customer current at a check-out counter being served, the probability that that customer will finish service is $1/s$ and a random number will be used to determine whether that customer did finish service.

The Lines of Customers

Often when a new customer arrives, every check-out counter will be busy with a current customer. Thus, we need to put the customer in a queue. Thus, associated with each check-out lane will be a queue of customers. New arrivals will go to the end of the queue. When the current customer at a counter finishes service, he will depart and the next customer will be dequeued, if there is one. That customer then becomes the current customer.

Opening New Lanes

Another simulation parameter will be the maximum line length, limiting the length of any queue. If a customer arrives when all of the queue lengths are at the maximum allowed, then a new lane will be opened and the new arrival will, naturally, go to that lane. (To keep the simulation as simple as possible, no customer will ever switch from the lane he is in to a shorter lane.) Thus, we will need to maintain a list of lanes. When a new lane is opened, we will append it to the list of currently open lanes. There will be no built-in limit to the number of lanes possible, although in practice the number of lanes should never be very large.

Closing Idle Lanes

A fourth simulation parameter will be the maximum idle time for any lane. If a lane has been idle, that is, no customer being served, for at least the maximum idle time allowed, then that lane will be closed. When a lane is closed, it should be removed from the list of lanes. (That will result in a "renumbering" of the lanes, but that is ok.)

Time

The last two simulation parameters will be the start time and the ending time. These will use the `Time` class. See the document `Time Class.pdf` for details. Times are input and output in the format `hh:mm:ss`. Internally, they are created using a `Time` constructor: `Time(hh, mm, ss)`. The maximum idle time will be stored as a Time object. For example, if the maximum idle time is 5 minutes, then its value will be `Time(0, 5, 0)`.

The Simulation

I suggest that, after a few necessary initializations, you create a `for` loop with index variable `time`, a `Time` object. The `for` loop should run from the start time to the ending time, in one-second increments. On each iteration of the loop, perform an update of the situation:

- Check for a newly arriving customer.
- If there is one, then
  - Check to see whether all the customer queues are full. If they are, then create a new check-out lane.
  - Place the newly arriving customer in the shortest queue.
- For each check-out lane,
  - Check for a departure.
  - If there is one, then dequeue the next customer from the queue for that lane and make him the current customer.
  - If there is not one, then designate that lane as "idle" and note the time.
- Check the idle time of all lanes. If any idle time exceeds the maximum idle time, then remove that check-out lane.

Input

Your program should read the six simulation parameters in the following order: arrival rate, average service time, maximum line length, maximum idle time, start time, ending time.

Events and Output

The events that will trigger output are

- A new customer arrives, in which a new lane may or may not be opened.
- A customer finishes service and departs.
- A check-out lane is closed.

If any of these events occurs on an iteration of the for loop, then the current state of the simulation should be displayed. Otherwise, do not display it, as it will be no different from the previous display.

The display should include:

- The current time.
- The announcement of the opening, if any, of a new check-out lane.
- Announce the arrival of a new customer and which lane he chose.
- Announce the departure (end of service) for any customers and from which lane they departed.

- Announce any customers who were dequeued and began service and in which lane it occurred.
- The announcement of any closures of check-out lanes.
- The current status of each check-out lane, including the current customer and the queue. If there is no current customer, then report "idle." If there is a current customer, but the queue is empty, then report "no one in line." If all lanes are closed, then report "All lanes are closed."

Name the program `Drive_Thru_Window.cpp` and place it and all related files in a folder named `Project_4` and drop it in the dropbox by Sunday, April 16.

Sample Output

The following is a segment of a much longer output. This segment shows an acceptable form in which to present the output.

```
TIME =  4:28:32
Customer 26 departs from lane 1
Customer 28 begins service at lane 1
Lane 1: customer 28 at lane, customers in line {32}
Lane 2: customer 30 at lane, customers in line {31}

TIME =  4:29:03
Customer 33 arrives at lane 1
Lane 1: customer 28 at lane, customers in line {32, 33}
Lane 2: customer 30 at lane, customers in line {31}

TIME =  4:29:11
Customer 34 arrives at lane 2
Lane 1: customer 28 at lane, customers in line {32, 33}
Lane 2: customer 30 at lane, customers in line {31, 34}

TIME =  4:29:19
Customer 28 departs from lane 1
Customer 32 begins service at lane 1
Lane 1: customer 32 at lane, customers in line {33}
Lane 2: customer 30 at lane, customers in line {31, 34}

TIME =  4:29:40
Customer 30 departs from lane 2
Customer 31 begins service at lane 2
Lane 1: customer 32 at lane, customers in line {33}
Lane 2: customer 31 at lane, customers in line {34}
```