CMPT270 Intersession 2017

Assignment #3

Queue Analysis of the Tim Horton's Lineup for Coffee

As mentioned in class, there are usually two ways in which customers are lined up for access to a number of service windows. Typically in a bank, there is one queue line for everyone and a number of windows. When a window becomes free, the next person at the head of the queue leaves the queue and proceeds to the window. People coming into the bank simply join the single queue at the tail of the queue.

At Tim's in Geology, we have the other approach to providing service. Every service window has its own queue. Typically when a person comes up to the lines, they choose the shortest line to join and then stay in that line. It can frustrate some people to see the longer line actually move faster and sometimes people will line hop (i.e. when the other line is shorter than the queue in front of me).

So which is the best approach? In this assignment we are going to try and determine that. Here are the parameters for the simulation.

For both simulations you are to do the following:

- Run the simulation for 6 hours
- A new customer group arrives every 1 minute ± 58 seconds (i.e. as little as 2 seconds apart and as much as 1 min 50 seconds apart
- A group will consist of 1 to 6 customers following this distribution:
    - 1 person – 30%
    - 2 people – 40%
    - 3 people – 20%
    - 4 people – 5%
    - 5 people – 3%
    - 6 people – 2%

  Each person in the group joins the queue single file but we can use the same join time for each member of the group.
- Average serving time for a customer is 2 minutes ± 65 seconds.

For each simulation below, report on the following:

- Maximum number of customers served
- Average number of customers served per window per hour
- Average wait time in the queue for each customer
- Maximum wait time in the queue
- Average queue length for each queue
- Maximum queue length for each queue

Simulation #1

Simulate a scenario with two server windows and two queues. When a group of customers arrive each individual in the group will choose the same queue, but the first individual will choose the shortest queue.

Simulation #2

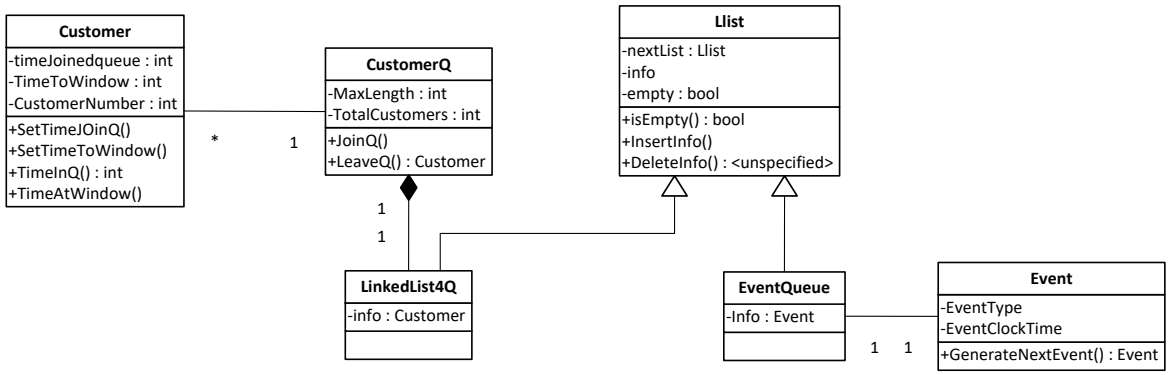
Simulate a scenario with two server windows and one queue. Report on the same statistics as scenario #1.

Based on your results of these two simulations, which queue configuration is the best, or is there no difference??

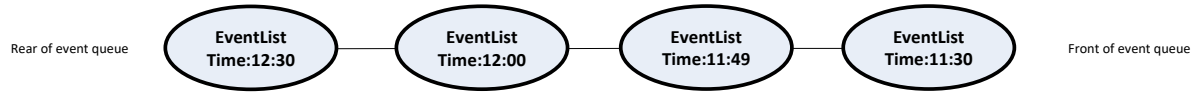
Considerations for designing your simulation:

- You can use an ordered linked list for the event chain rather than actually running a clock
  - List is ordered by increasing clock time
  - Each list element has the event type (customer arrives and joins queue, customer leaves queue and is at server window, customer leaves window), the clock time when the event is to occur
- Use a queue for each lineup. The info at each element in the queue is a customer. Along with the customer you need to record things like time when they joined the queue, time when they approached the server window, amount of time in the queue, amount of time at the server window. Then before destroying the customer object, update your statistics counts and totals for final analysis

Consider the following UML to begin your development. It may be incomplete. Document fully any changes or additions you may make to this design:



Note: The linked list that supports the CustomerQ must behave as a queue. The linked list that is the EventQueue must be in order of event time, i.e. as you work from the rear to the front of the queue the event time gets smaller. Consider:



When you generate the next event of the type pulled off the event queue front and it should occur at 11:35 it must be inserted between the event at 11:30 and the event at 11:49.

Submit all source code files and your data results which include the simulation raw data and the calculated results in a Word doc or pdf.

This assignment is to be accomplished using extreme programming – pair up with someone in the class and using pilot/copilot share the programming/overseeing duties as we discussed in class. It works best if you share the duties even if one of you is a much more adept programmer.

Marking scheme:
50 marks – correct implementation of the classes and execution of the code.
30 marks – results and documentation.