ECM2433 – Report

Design Decisions

Queue as a doubly-linked-list

For this coursework, I decided to implement the queue as a doubly linked list. This was so that in the case of a timed-out customer, they can be removed with relative ease (as when this node is accessed it already has reference to its previous and next node).

Person as a node

Originally for this coursework I implemented two separate linked lists, one for my service points (which is still being used) and the other to store people. Both were implemented with their own node structs – the person/service point being they key. However, I quickly encountered segmentation faults with this method when used with people. This was most likely since when I freed a node it also freed the person acting as its key – behaviour which I did not want. To solve this, I gave the person struct next and previous attributes which removed the risk of this occurring.

Poisson distribution for people arriving per tick

I decided to use the Poisson distribution to model the arrival of people at the post office per tick. This is because the Poisson will only return an integer value but can take a non-integer mean. Also, because Poisson has the “forgetfulness property” – the number of people to arrive in the next tick is independent of the people who have arrived in previous ticks. I believe this property simulated new arrivals best as people are likely to arrive at the post office independent of each other.

Bernoulli distribution for the chance of a customer timing out

To model the probability of a customer becoming bored and leaving the post office prematurely I decided on the Bernoulli distribution, as the Bernoulli distribution only return successes or failures (in this case a success is the customer leaving and a failure is them leaving). As the longer a customer waits in the queue, the more likely they are to leave, I have included a variable (called leaveChance) held within the parameters text document which specifies by how much the customers chance of leaving will increase per tick they spend in the queue.

Flat/Uniform distribution for time taken at a service point

The uniform distribution provides a suitable distribution for the time taken at a service point as it will produce a random number between a minimum and a given maximum. In most post offices, they tend to guarantee that you will be served within a maximum value. The uniform distribution allows for an input of this maximum, so it can be altered to suit different post office set ups.

Assumptions

- The shortest amount of time to be spent at a service point is 1.
- All times spent at a service point have the same probability.
- Each service point has the same probability distribution.
- Each service point has the same minimum and maximum service time.
- The distribution of people arriving throughout the day does not change.

Experiment

With this simulation I ran a series of tests, first with only one simulation and the following parameters:

```
maxQueueLength        20
numServicePoints      3
closingTime           100
maxServiceTime        5
mean                  5.0
leaveChance           0.1
boredomChanceIncrease 0.05
```

Here was the output (I have only included the last few intervals for conciseness):

```
Current time interval: 99
There are 3 customers currently being served
There are 20 customers currenlty in the queue
There are 232 unfulfilled customers, 57 fulfilled customers and 70 timed-out customers
There are 0 customers currently being served
There are 20 customers currenlty in the queue
There are 235 unfulfilled customers, 60 fulfilled customers and 71 timed-out customers
The post office has now closed
Current time interval: 101
There are 3 customers currently being served
There are 20 customers currenlty in the queue
There are 235 unfulfilled customers, 60 fulfilled customers and 71 timed-out customers
It took 0 intervals after closing to serve the remaining customers
Current time interval: 101
There are 3 customers currently being served
There are 0 customers currenlty in the queue
There are 235 unfulfilled customers, 72 fulfilled customers and 79 timed-out customers
The average waiting time was 3.333333
```

Using the same parameters and running 1000 simulations yields:

```
On average, there were 104.027000 fulfilled customers
On average, there were 293.389000 unfulfilled customers
On average, there were 86.843000 timed-out customers
On average, the average wait time was 2.165928
```

Increasing the number of service points to 5 (1/4 of the queue length) yields:

```
Current time interval: 99
There are 5 customers currently being served
There are 20 customers currenlty in the queue
There are 193 unfulfilled customers, 174 fulfilled customers and 69 timed-out customers
There are 0 customers currently being served
There are 18 customers currenlty in the queue
There are 193 unfulfilled customers, 179 fulfilled customers and 70 timed-out customers
The post office has now closed
Current time interval: 101
There are 5 customers currently being served
There are 18 customers currenlty in the queue
There are 193 unfulfilled customers, 179 fulfilled customers and 70 timed-out customers
It took 0 intervals after closing to serve the remaining customers
Current time interval: 101
There are 5 customers currently being served
There are -1 customers currenlty in the queue
There are 193 unfulfilled customers, 194 fulfilled customers and 76 timed-out customers
The average waiting time was 1.664948
```

And for 1000 simulations it yields:

```
On average, there were 124.005000 fulfilled customers
On average, there were 82.683000 unfulfilled customers
On average, there were 65.246000 timed-out customers
On average, the average wait time was 1.641595
```

Increasing the number of service points to 10 (1/2 of the queue length) yields:

1.

```
Current time interval: 99
There are 6 customers currently being served
There are 2 customers currenlty in the queue
There are 0 unfulfilled customers, 187 fulfilled customers and 0 timed-out customers
There are 6 customers currently being served
There are 2 customers currenlty in the queue
There are 0 unfulfilled customers, 189 fulfilled customers and 0 timed-out customers
The post office has now closed
Current time interval: 101
There are 10 customers currently being served
There are 2 customers currenlty in the queue
There are 0 unfulfilled customers, 189 fulfilled customers and 0 timed-out customers
It took 0 intervals after closing to serve the remaining customers
Current time interval: 101
There are 10 customers currently being served
There are 0 customers currenlty in the queue
There are 0 unfulfilled customers, 191 fulfilled customers and 0 timed-out customers
The average waiting time was 3.000000
```

2.

```
Current time interval: 99
There are 4 customers currently being served
There are 19 customers currenlty in the queue
There are 175 unfulfilled customers, 211 fulfilled customers and 20 timed-out customers
There are 6 customers currently being served
There are 20 customers currenlty in the queue
There are 175 unfulfilled customers, 215 fulfilled customers and 21 timed-out customers
The post office has now closed
Current time interval: 101
There are 10 customers currently being served
There are 20 customers currenlty in the queue
There are 175 unfulfilled customers, 215 fulfilled customers and 21 timed-out customers
It took 0 intervals after closing to serve the remaining customers
Current time interval: 101
There are 10 customers currently being served
There are -1 customers currenlty in the queue
There are 175 unfulfilled customers, 235 fulfilled customers and 25 timed-out customers
The average waiting time was 3.272340
```

As you can see from these results, simply increasing the number of service points available – and keeping everything else the same – will have a positive correlation with wait time reduction, up to a point. Increasing the number of services point up to 5 (basically doubling it from 3) halved wait time but repeating the process and doubling it to 10 had the opposite effect. It seemed to *increase* wait time.

While this is only a preliminary experiment and does not go into any great depth it does appear to suggest a “sweet spot” for a queue length to service point ration. That is, there may exist a ratio that decreases the wait time the most.