

CS417 Programming Assignment 3

Adam Lewis

Contents

1	Painful Airlines Ticketing Terminal	1
1.1	Background	1
1.2	The problem	1
1.2.1	A few rules	2
1.2.2	Customer arrival rates and service times	2
1.2.3	Output	2
2	Submitting your assignment	2

1 Painful Airlines Ticketing Terminal

1.1 Background

Painful Airlines has a problem: they were just given the award as being the second worst airline in the world (only the national airline of North Korea was voted as having a worse reputation with their customers).

One area of concern is the inordinate amount of time that it takes for travelers to check-in for their flight. The operations staff at Painful Airlines (Slogan: "We hate to fly, and we hate you!") has come up with a new arrangement for the check-in counter. They have contracted with you to develop a simulation that will confirm whether or not this scheme will speed up customer check-in.

The new procedure will have three lines: a line for first class passengers, a line for frequent flyers, and a line for economy class passengers. There will be three check-in counters: a first class, frequent-flyer, and economy class. First class travelers arrive randomly at an average rate of 1 customer every 20 time periods. Frequent flyers arrive at an average rate of 1 customer every 5 time periods. Economy flyers arrive at an average rate of 1 customer per time period. It takes an agent a random amount of time between 2 to 10 time periods to service a customer. In this scheme, if no customers are in the first class line, then that agent may service a frequent flyer customer or, if no customers are in the frequent flyer line, an economy class customer. Similarly, if there are no customers in the first class or frequent flyer line, then that either the first class or frequent flyer agent may service a customer in the economy line (frequent flyer agent has to select first, as the 1st class agent has greater seniority and wants to work less).

This is a classic problem in queuing theory and has connections to the computer science that underlies the scheduling functions in your operating systems. This is in a class of problems called "discrete event simulations" where we want to evaluate the performance of a system over a time period but rather than using actual clock time, we simulate the clock using an integer timer. The problem being modeled tells us the mapping between simulated time and actual time. For instance, for Painful Airlines each tick of the simulated clock is the same as one minute (though given their customer service history, each minute feels like an hour to their customers).

1.2 The problem

Write a program in C++ that will simulate this system running for a configurable number of time periods as defined on the program command line. On each time tick, display the number the customers waiting in each line.

At the end of the simulation run, display the following:

- Average and maximum line lengths for each of the three lines.

- The average amount of time spent by a customer in each line.
- The average amount of time a customer service agent spends with a customer.

Try to use the STL as much as possible. Remember: reuse rather than rewrite.

1.2.1 A few rules

- If the line for First Class passengers is empty, then the agent serving the First Class line can assist customers in the Business Class line.
- If the line for First Class passengers is empty and the line for Business Class passengers is empty, then the agent on the First Class line can assist passengers in the Economy line.
- If the line for Business Class passengers is empty, then the agent at the Business Class customer position will first see if there are customers in the First Class line. If that line is empty, then the customer service agent will service customers in the Economy line.
- If the Economy class line is empty, then the customer service agent in that position will assist First Class passengers first, and then Business class passengers.

1.2.2 Customer arrival rates and service times

- The system will keep a simulated clock that starts at zero and counts the simulated number of minutes in the simulation.
- The simulation will run 12 hours (or 720 minutes).
- Customers will randomly arrive in the Economy line at the rate of 1 every 3 simulated minutes
- Customers will randomly arrive in the Business Class line at the rate of 1 every 15 simulated minutes.
- Customers will randomly arrive in the First Class line at the rate of 1 every 30 simulated minutes
- It takes a customer service agent randomly between 5 - 10 minutes to assist an Economy Class customer
- It takes a customer service agent randomly between 6-12 minutes to assist a Business Class customer
- It takes a customer service agent randomly between 5 - 20 minutes to assist an First Class customer

1.2.3 Output

Your simulation needs to run for a simulated 12 hours (720 minutes). Every 10 time ticks, your program needs to display how many customers are in each line and whether or not a customer agent is assisting the customer.

At the end of the simulation, your program needs to display:

- Average and maximum line lengths for each of the three lines.
- The average amount of time spent by a customer in each line.
- The average amount of time a customer service agent spends with a customer.

2 Submitting your assignment

When ready to submit, make certain that all code has been committed and pushed to your class repository on GitHub. Make a notation in the assignment submission on Blackboard where to find the code in your repository.