

Project 2 – Simulating a Store

Learning Objectives

- Implement Linked Lists
- Implement Queues
- Implement Priority Queues
- Perform a Code Analysis

Overview

A very common use of computers is to run simulations. A simulation is a program designed to model a real system of interest. By setting parameters and selecting appropriate input values, results are produced that mimic the behavior of the real system. If the simulation is a valid representation of the real system, it can even predict the future performance of the real system.

In this project, you will construct a program to simulate a grocery store. Your goal is to use your simulation to answer a specific question: Should the owner of the store remodel her checkout lanes to include express self-checkout lanes and, if so, how many?

Model

Customer Data

To prepare for your analysis the owner has collected the following data for you:

- Arrival time for customers
- Order size for customers
- Average time it takes them to select an item

All times are in minutes, and arrival times are set in reference to store opening time of zero minutes. This data is in the file `customer_data.txt`. Each line has 3 values: <Arrival Time> <Order Size> <Average Time To Pick Up An Item>

Discrete Event Simulation

We will use a discrete event simulation to simulate the grocery store. This means time is managed by using a simulation clock that is advanced whenever an interesting event occurs. For example, if the simulation time is 0.0 minutes and a customer enters the store at a time 2.0 minutes, we mark the passage of time by advancing the simulation clock to 2.0 minutes ($\text{current time} = 0.0 + \text{time of arrival event} = 2.0$).

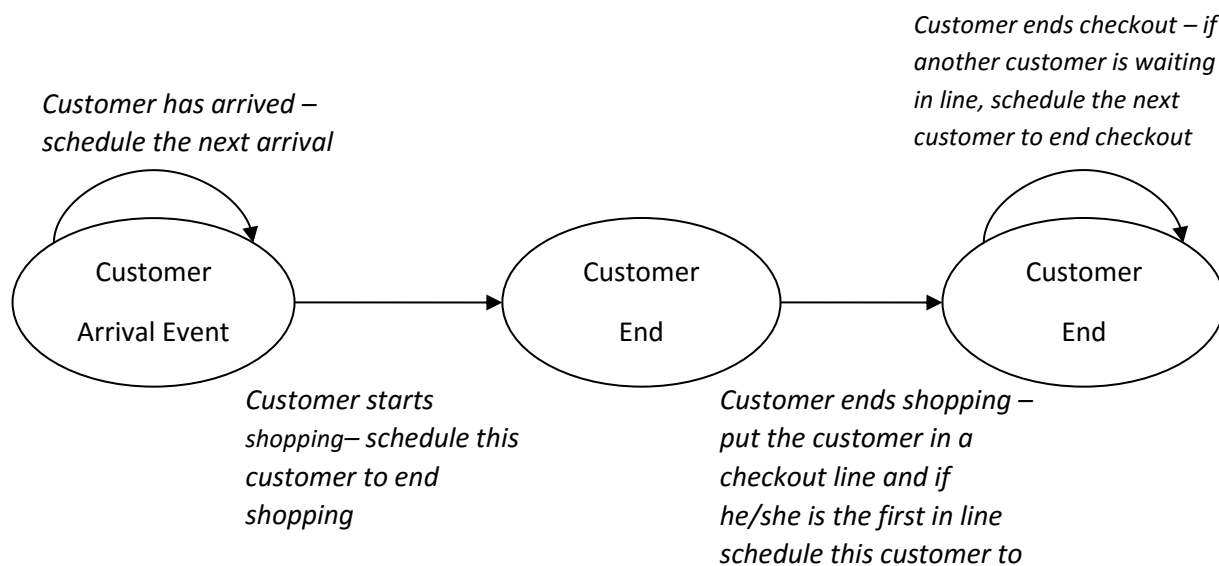
To model this system, we will need the following events:

- customerArrival
- customerEndShopping
- customerEndCheckout

An event consists of three pieces of information:

- event type (enumeration)
- event time (double)
- event customer (Customer)

The diagram shown below outlines the flow of events in our simulation:



Checkout Information

The owner has 8 checkout lines. She can remove 2 checkout lines to install 4 express self-checkout lanes.

Cashier takes .01 minutes per item to scan purchases and 1.5 minutes to pay. Customers using self-scan checkouts take .04 minutes per item and 3 minutes to pay.

Data Collection

Collect the following statistics:

- average waiting time per customer
- total customers passing through each line
- maximum length of each line

Use these statistics to write a proposal about how many self-checkout lanes the store owner should install, if any.

Implementation Details

-You must build your own priority queue on a linked list.

-You must build a register queue using a linked list.

-You must perform a code analysis (aka provide a Big Θ) for each method in your priority queue and register queue classes.

Grading*

A – Simulation runs and collects data for 8 Regular & 0 Self-scan Checkouts, 6 Regular & 4 Self-scan Checkouts, and 4 Regular & 8 Self-scan Checkouts. Code analysis completed for your linked list, priority queue, and register queue classes. Paper proposing a remodel plan based on supporting statistics is included.

B – Simulation runs and collects data for 6 Regular & 4 Self-scan Checkouts. Code analysis completed for your linked list, priority queue, and register queue classes. Paper explains statistics that were collected about the simulation.

C – Simulation runs and collects data for 6 Regular & 4 Self-scan Checkouts. Code analysis completed for your linked list, priority queue, and register queue classes.

D – Simulation runs and collects data for 6 Regular & 4 Self-scan Checkouts. Code analysis completed for Linked List Class.

*Up to one and a half letter grades (15% of total points) could be removed for bad style and/or poor testing. Projects must compile to receive a letter grade.