CX 4230 Project Checkpoint

Members:

Fei Ding, Xiaoran Zhu, Zhengyang Qi

Abstract:
This project aims to construct an event-driven system that simulates checkout lines in grocery stores. Grocery stores usually have significantly different volumes of customers during different time intervals in a day, which makes it inefficient to keep a fixed number of checkout lines open for the whole day, since it is expensive to keep all lanes open when there are few customers, while opening insufficient number of lines during peak hours may annoy customers significantly. Our project will simplify such situations into an event-driven simulation system and will explore the relationship between the number of open lanes and the rate of arrival of customers to the checkout stations.

Description of the system:
Grocery stores have a steady demand and large volume of customers everyday, so it is common for grocery stores to have multiple checkout lanes and various types of lanes, such as human service lanes, auto-checkout lanes and express checkout lanes. However, the number of customers in a grocery store varies a lot during different times of day. According to google maps, the Publix at Atlantic Station has rush hours that starts from around 12 PM and reaches its peak at 7 PM, while the peak number of visitors per hour could be 4 - 5 times of the number of visitors at 7 AM.

Thus, it is inefficient to keep a fixed number of checkout lanes open regardless of the number of customers, because each open line adds to the total operation cost, such as the hourly pay for cashiers, so it is expensive to keep all lanes open when there are few customers. However, if there are few open lines during peak hours, the customers may get dissatisfied and may not continue to visit the store as frequently, which also leads to a potential loss in revenue.

Literature review:
Retail store operation has always been an interest for researchers, including staffing schedules, shop layouts, and customer flows etc. Opara-Nadi (2005) studied the difference between express checkouts and regular checkouts and suggested that in terms of time efficiency, express checkouts were better. However, the shortcoming of increased possibilities for having errors in express checkouts is also obvious compared with regular checkouts. There are many researchers taking simulation to solve mathematical problems. Schimmel (2013) studied the effects of express checkouts to waiting lines in a supermarket by exploring two different queueing models using base case and worst case and comparing the result. The conclusion shows that regular and express checkouts experience larger queues even though express checkout customers experience shorter waiting time. In addition, Yamane (2012) investigated the best checkout layout for a local retail store in Japan and managed to eliminate congestion

issues which could also be extended to a combination between congestion and time in the queue. In this proposal, we will examine two factors, checkout line picking preferences and different checkout station layout, to discover their effects on the customer's total checkout time in the system.

In *SIMULATION MODELING OF CUSTOMER CHECKOUT CONFIGURATIONS*, Rossetti and Pham (2015) constructed a discrete event simulation model through Java Simulation Library (JSL) for both the shopping and checking-out behaviors based on the First Annual IIE/Arena Contest problem: The SM Superstore (Kelton et al. 2007). In their proposed models, customers will first start shopping in the shopping aisles for several items upon arrival and then go to the checkout area, where they may choose either a regular checkout lane, an express checkout lane, or a self-checkout lane, while a self-checkout lane may also be a regular lane or an express lane. During checkout, a cashier will scan the items, a bagger will help bagging the items, and eventually the customer will pay. The arrival of a customer is generated by a non-homogeneous Poisson process, while other parameters such as time for scanning, bagging, and paying are given. Our project is inspired by Rossetti and Pham's work with a focus on the checking out behavior.

Data Source:

Due to the difficulty in obtaining real data from grocery stores in practice and the potentials of privacy violation, we have created a data generator for generating random virtual data to feed into the simulator.

We take in a list of tuples reflecting the number of customers arriving at the checkout zone in the following way: (start_timestamp, end_timestamp, number_of_customers, distribution_type, param_1, param_2). We may either take in a single tuple to generate a time uniformly-distributed virtual customer data or a list of parameterized tuples to generate data which approximates to a time-varying data distribution of customer arrival events.

Field	**Meaning**
`start_timestamp`	The *start* timestamp of the interval in *seconds*
`end_timestamp`	The *end* timestamp of the interval in *seconds*
`number_of_customers`	Number of customers arriving in this interval
`distribution_type`	The *distribution type* for **number of grocery items**
	Can be `"uniform"` or `"gaussian"`
`param_1`	The 1^{st} distribution parameter
	min for uniform , *mean* for gaussian
`param_2`	The 2^{nd} distribution parameter
	max for uniform , *standard deviation* for gaussian

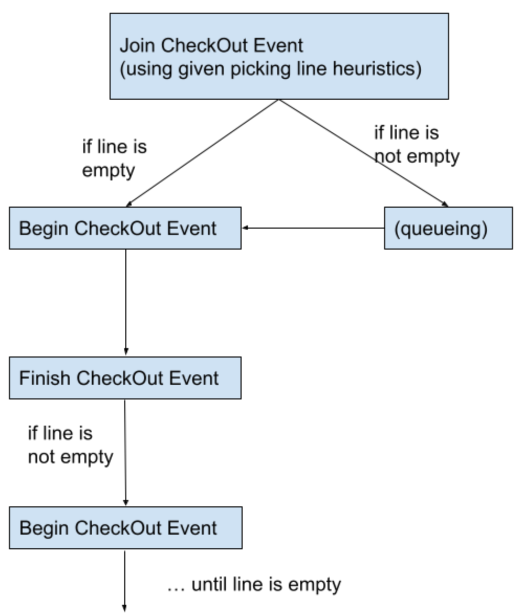
-

Conceptual model:
Even though the customer may have a few number of activities in the grocery store including finding items, selecting items, making payment etc, we will only focus on the checkout process. We assume a customer will first join the checkout line and pick one based on the picked algorithm. He will wait in line if checkout lines are busy. Otherwise, he will begin the checkout process, including scanning items and making payment.

We assume that the checkout time T for each customer is deterministically dependent on the number of items they purchase N and the type of checkout line they use. There are three types of checkout lines in our simulation model, cashier, express and self-checkout lines. Specifically,

$$T_{cashier} = N + 7$$
$$T_{express} = N + 4$$
$$T_{self} = 2N + 1$$

The intuition behind is that an experienced cashier spends roughly the same amount of time (~1 second) to scan a grocery item regardless of what line (regular/express) they are in, but an express checkout line in general has less overhead, so it has a smaller constant term. A self-checkout line is not operated by a cashier, but an inexperienced customer will likely spend twice as much time for a single item. We will continue the process until the line is empty.



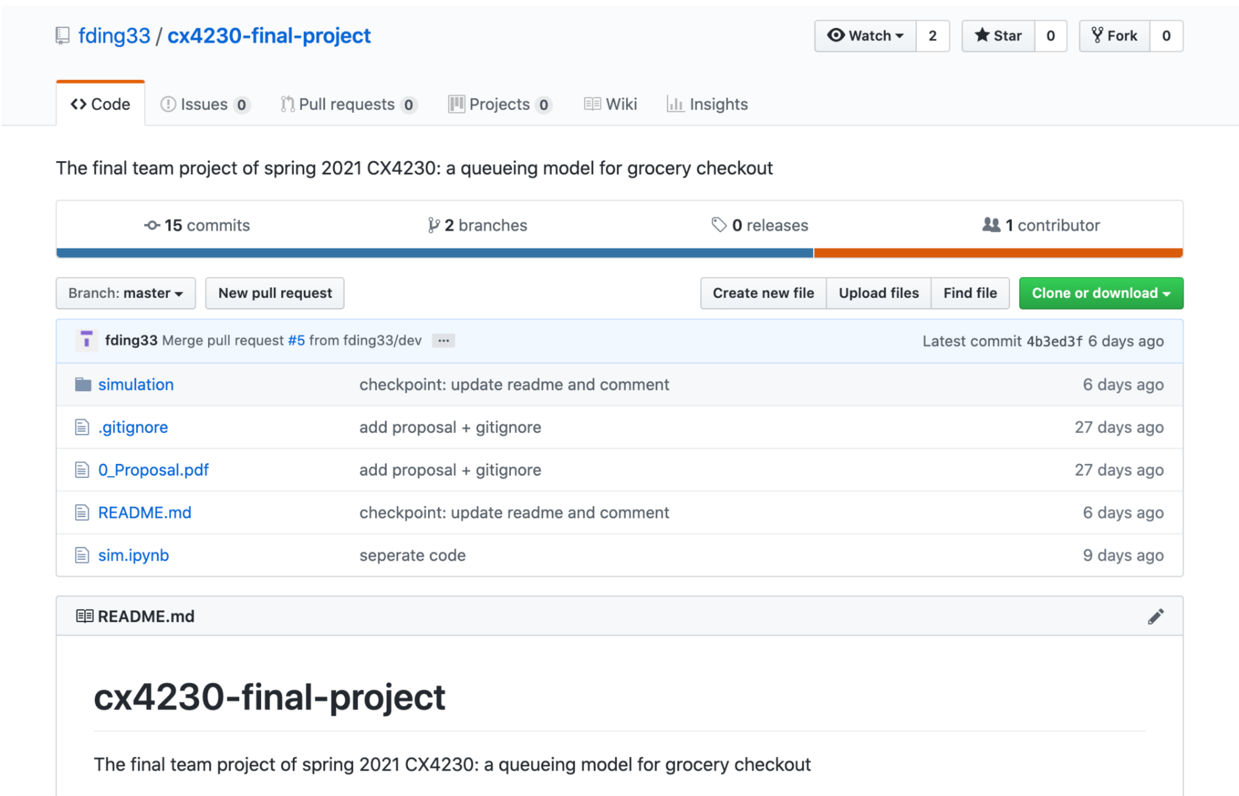
In terms of the customer line selection algorithms, we come up with several heuristics how a customer selects a checkout line and we will explore the effect of these different customer queueing behaviors.

Pure Random	As the name suggests, it's not based on any rules. The customer randomly selects an available line and goes for it. (It serves as a bottom line in terms of measuring simulation performance.)
Least Person	The customer selects the line with the least number of people waiting in that line. (This is closest to reality.)
Least Item	The customer selects the line with the least number of items held by the people waiting in that line. (Though this is not realistic as customers do not count this quantity during checkout, it "seems" more efficient, so we will do it from a simulation perspective.)
Least Time	The customer selects the line with least *expected* queueing time determined by customers waiting ahead. (Though this is not realistic as customers won't bother calculating this quantity during checkout, it "seems" more efficient, so we will do it from a simulation perspective.)
Round-Robin	Each checkout line takes turns to accept new customers. (Though this is not realistic as customers cannot be forcely assigned a line to use, it is a "fair" algorithm, though. The fact that it avoids selfishness of customers may increase efficiency from the least person heuristic.)

Platform of development:
Python3.7 is used for developing the simulator code in “simulation” folder.
The *Jupyter notebook* “sim.py” contains documented experiments and results.

Show of Progress:
By this point, we have finished implementing the base simulator under the “simulation” directory including the following classes shown in the table.

Entities	customer.py	checkout_line.py	grocery_store.py		
Events	JoinJoinCheckoutEvent	BeginCheckoutEvent	FinishCheckoutEvent	LineClose Event	LineOpenE vent
Simulator	utility.py	data_generator.py	simulator.py		



We have conducted some test runs as a quick verification that we have implemented the system correctly. Below is the raw output and some statistics of one simulator run experiment.

```
Line ID: 0
Type: cashier
Closed: False
Customers:
11
ID      #items  t_join  t_begin t_finish
0       20      3       3       30
3       18      13      30      55
8       19      31      55      81
11      12      56      81      100
19      10      87      100     117
21      9       90      117     133
25      2       102     133     142
28      17      122     142     166
31      20      134     166     193
37      20      153     193     220
42      20      181     220     247
```

```
Line ID: 1
Type: cashier
Closed: False
Customers:
13
ID      #items  t_join  t_begin t_finish
1       20      5       5       32
5       1       18      32      40
9       7       44      44      58
10      13      52      58      78
13      16      65      78      101
18      6       80      101     114
22      6       91      114     127
26      3       116     127     137
29      11      124     137     155
30      16      127     155     178
34      15      140     178     200
39      4       162     200     211
44      11      193     211     229
```

```
Line ID: 2
Type: express
Closed: False
Customers:
13
ID      #items  t_join  t_begin t_finish
4       1       17      17      22
7       9       26      26      39
12      9       61      61      74
14      8       70      74      86
16      10      75      86      100
20      7       88      100     111
23      7       94      111     122
27      10      117     122     136
32      9       136     136     149
35      9       143     149     162
36      3       152     162     169
41      10      180     180     194
43      2       189     194     200
```

```
Line ID: 3
Type: self
Closed: False
Customers:
8
ID      #items  t_join  t_begin t_finish
2       21      9       9       52
6       11      23      52      75
15      14      73      75      104
17      15      77      104     135
24      9       95      135     154
33      11      138     154     177
38      21      157     177     220
40      16      165     220     253
```

```
Min Wait: 0
Max Wait: 55
Mean Wait: 16.67
Std Wait: 13.87
```

Division of Labor:

Xiaoran Zhu	- Seek for model dataset (POS dataset) - experimentation on checkout lines composition - possible to extend the model like separation of checkout/scanning and payment area
Zhengyang Qi	- Seek for model dataset (find more detailed data for checking out behaviors) - experimentation on different line selection algorithms - take cost of hiring cashier and buying machinery into consideration to improve efficiency - Explore relation between waiting time & satisfaction as
Fei Ding	- Experimental verification of some famous queueing theory theorems including the Little's Law. - Seek for model dataset (POS dataset) - Seek better data generation process that more accurately reflect the reality (even through real observation)

Git Repository:
https://github.gatech.edu/fding33/cx4230-final-project

Reference:

Opara-Nadi, G. 2005. “Electronic Self-Checkout System Vs Cashier Operated System-a Performance Based Comparative Analysis.”Ph.D. thesis, Capella University.

Manuel D. Rossetti and Anh T. Pham. 2015. Simulation modeling of customer checkout configurations. In <i>Proceedings of the 2015 Winter Simulation Conference</i> (<i>WSC '15</i>). IEEE Press, 1151–1162.

Schimmel, M. 2013. “Deployment of Express Checkout Lines at Supermarkets.” https://www.few.vu.nl/en/Images/werkstuk-schimmel_tcm39-335266.pdf.

R. N. Rechtschaffen, "Queuing simulation using a random number generator," in IBM Systems Journal, vol. 11, no. 3, pp. 255-271, 1972. doi: 10.1147/sj.113.0255 http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5388236&isnumber=5388231