Alex Mitro

Fair Simulation Report

In order to accurately depict a fair in the simulation, there were many things that I had to take into consideration as I created it. The hardest part, by far, is finding a way to handle having multiple ticket booths. I ended up taking in a command line argument and using that to determine how many to make. I then created however any cash/express lines as I needed by creating the booth object and storing it in the appropriate array.

For the express booth, I created a line by using a queue object. As a new person enters the queue, I add their expected time to buy the tickets to the total. In the express lines case, that would be 15 seconds. For the cash line, I did the exact same thing, except I predetermined how many tickets the person would buy and calculated how long it would take to finish their transaction. Keeping track of this value will help me determine which line new customers will choose while also letting me keep track of wait times to find the averages.

For the Roller Coaster, I created it as its own object. The roller coaster would accept 60 people maximum per ride. I used this fact to help me determine how long the rollercoaster wait time was. I divide the size of the line by 60 and determine how many rides it would take to clear the line from that point.

For the Merry-Go-Round, I know that it accepts 20 people per ride. In a similar way, I calculated how many rides it would take in order for the line to clear. I used this to help find the wait time. Since the size of the Merry-Go-Round can change, the wait times for the line will change as well. The number of seats in the Merry-Go-Round is accepted as an argument when the program is executed. I manipulate the expenses based off of whether or not I add more or less seats to the ride.

For the Games and Stalls, I accept a number for how many stalls I want to create. The capacity of these stalls is calculated by multiplying the number of stalls by 5. I calculate the cost for adding more or less stalls before the current experiment begins.

For each run of the experiment, it simulates the fair for 100 days. Each day begins at 10:00 and ends at 6:00. I represent the day by using a loop and representing each iteration as a second of the day. Each part of the park has a method that checks to see if a transaction/ride is ready to start. After it has started, each iteration that represents a second will decrement how much longer the current transaction/ride is running. For example, if the setup time for the roller coaster was 30 seconds, each iteration through the loop will decrement that timer. After 30 loops, the setup would have been complete and the ride will begin.

In order to find the average length and wait times for the various queues, I would check their lengths and times every 5 minutes and then find the average of them at the end of the day. I would add up all the averages for the 100 days, and then find the average for that.

In order to find the number of wasted tickets, I set a ticket count for how many tickets were bought in total. Then, each ride would keep track of tickets they have collected throughout the day. At the end of the day, I add all of the tickets gathered by the rides and subtract it from the total.

In the excel file that will be stored in the same repository as this report, you can see the data that I collected from running my simulation. Each test is run with only 1 aspect of the simulation different from the original. The following is the conclusions I have come up with after running these various simulations.

**Observation/Analysis**

When the simulation is run with the original state, I observed that for the most part, there was nothing unusual going on. However, the only major issue came with the Merry Go Round wait time and line length. After considering why this could be happening, its pretty clear that a capacity of 20 is inefficient. It takes roughly 6 minutes for 20 people to get through the ride. In contrast, the roller coaster accepts 60 people and can get them through in about 5 or less in some cases. That means there are a lot of people possibly being sent to the line. That means the lines for other rides are shorter because everyone is spending time waiting to get on the Merry Go Round. I expected that adding more seats on the ride would be more efficient in the long run, which is exactly what ends up being observed. With a shorter wait time for this ride, people can go do more activities in the park, which should balance out the wait times and result in less ticket wasting.

Adding more seats to the Merry Go Round was definitely the way to go, and the overall profit loss may have been a lot, but the fair is still making money. There is still room to make money by manipulating the stalls.

When messing around with how many stalls the fair could get away with, I decided to try taking away stalls in order to make some more money back lost from the Merry Go Round. After lowering the count to 7, I found that, unlike when there were 10, the stalls actually reached capacity, which isn’t always a bad thing. This did not happen too often, and it seemed like the perfect number of stalls to have in order to reach maximum efficiency without having any useless stalls. After adding more stalls, it had little impact on the overall effectiveness of the fair, so I came to the conclusion that adding more is not efficient or effective.

Switching my attention to the lines, I noticed that using the original layout (1 of each type), the cash line at the beginning of the day was very long, and usually stayed very long for the first part of the day. By adding another cash line, the average line length went down to 2 from the original 4. This makes sense because the insane line length from the early customers was split up into 2 lines, which I assumed would cut the original average in half.

If we look at the express booth instead, we see that the original layout left the express booth with an average of 5 people in the line. This is not unrealistic, because most of the early customers will enter the express line at the beginning. However, as the morning passed, the express line reached low numbers while the cash line stayed long, so new customers would arrive and choose the express line instead of waiting long for the cash line. This constant line switching would keep the average length a little higher. After adding a 2nd express line, the average line length went down to 2. By allowing the early customers to split their original length by splitting into two lines, the average went down, resulting in smaller lines.

When there are 2 of each booth, the averages are both 2, which is the most efficient way of getting people into the park. When we look at the overall average profit of the different types of line setups, we can see that the setup with 2 cash lines results in the highest profit. However, the overall objective is to create shorter lines and make the most profit, so the best way to setup the lines is to have 2 of each type.

In conclusion, it seems as though adding more seats on the Merry Go Round, getting rid of 3 stalls, and having 2 of each line is the best setup for the fair. Doing this will result in decent profit and lower line wait times.

If the fair could make advertisements, they should try to get more people to show up during the day. The lines at the beginning of the day are very large compared to the later parts of the day. If the rate of people showing up went up to around 20-25 instead of 15, the profit made would definitely be way higher and the workers at the cash booths would have less wasted time. Also, we can see that the park could accept more people, because the coaster line average is less than how many people could actually fit on the coaster at one time. However, increasing the rate of people coming in may result in adding more game stalls.

In conclusion, the simulation proved that the ticket lines help determine how much money is being made in the park. Cash lines end up making more money by selling more tickets, but in order to decrease line size, it is worth adding an extra express line as well. This would result in profit and smaller line lengths. Also, the only major line problem came from the Merry Go Round. In order to achieve our small line objective, we must increase the size of the ride, even if we must increase the price.