Casey Sweet

Tucker

CS 273

7/20/18

In my final project, I set out to make an event that would be similar to what the Portland Trial Blazers may one day (or maybe already do) host. This event in total would have food trucks, and booths run by a bunch of local companies that are related in some way to NBA, Portland or the Pacific North-West, or the Trail Blazers, chances to meet some of the players, and a bunch of events that anyone could join in. These events could be like a dunk contest, trivia game, obstacle course, or, like I did, a shooting competition. This shooting event is simulated because the user can choose how long the event is, how many courts are in use, how long it takes to get the next player on the court shooting, and how long the players get a chance to shoot. This program has been very fun to work on, and there is so much more that could be done to it. There could be more simulation on the registration and people moving side of the event, or more simulation on the shooting and basketball mechanics side of the event, there is a lot of room for expansion with this program and I like that a lot. For the most part, coding this was straight forward, the only problem I ran into was getting the podium finish to appear properly, since the map value would not work right with the priority\_queue value. I was not able to figure out how to fix that problem, no matter what I did either the scores went with the highest to lowest scores, or lowest to highest. Between first thinking about it and final implementation, the only main thing that changed was the classes I used. Instead of having a RipCityEvent, Competition, and Registration class, I have a RipCityEvent, ShootingEvent, and Player class. This switch made for easier programming and a better design that was more simpler and followed more of the object-oriented-programming thinking better.

For this simulation, the focus was how to move the most people through in the most reasonable amount of time. For my graph I wanted to see what the correlation is for number of people and time allowed to shoot. With constants for the length of the event (seven hours), time between people (two minutes), and baskets (eight) I increased time to shoot from one to ten minutes and these are the results:

From this graph, it seems to be an exponential decline which is not what I would have guessed originally. I thought it was going to be linear. I also tried fifteen minutes of shooting for fun, and 170 people were able to shoot in that scenario, which is interesting that five minutes of difference only made seventy people worth of difference. Compared to the start when the difference between one and two minutes is about 200 people.

If I were to do this again, I would have focused more on the basketball side of the event. I tried to simulate a little bit of both, but I think by doing that I did not really simulate anything very much. If I did do it again, I would make the event time, basket number, amount of time to shoot, and time between shooters constant and focused more on the analysis of who is shooting where and how much and is it effective enough. Asking question like what is the most effective shot time-wise? Is the half-court shot even worth it because it has such a low percentage chance of going in, though it carries the most points? Also, one that I would really be interested in seeing, especially because this is a big talk in the NBA right now, is what is more effective the two-pointer or three-pointer? These type of questions I would focus more on if I could do it over, not only because I would have more of a simulation then but also because I would find this project even more fascinating. However, once I realized all of this, it was to late to turn back and start over, but it has been a great experience none the less and I am a more confident programmer from it.