CS-273 Final

Emergency room simulator

Final Summary

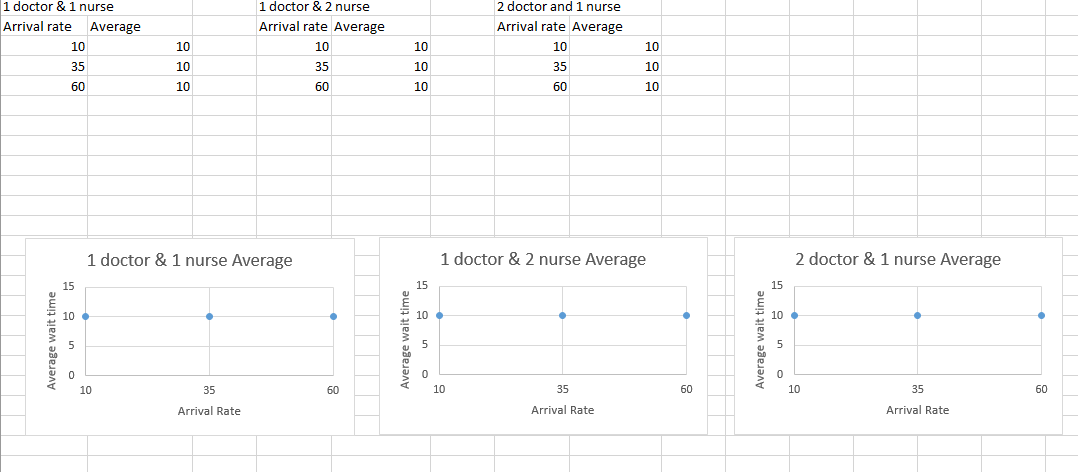
Average patient time visit:

1 doctor and 1 nurse

1 doctor and 2 nurse

2 doctors and 1 nurse

Plot:



Comparison:

The plots were the same because we were unable to get doctors to service the patients, only nurses serviced the patients. Because of this, in every scenario we got the same average wait time.

Change over time:

Our first UML outline was very simple to begin with. Over time we added many get and set functions as well as other functions to pass values around. We initially had one priority queue for all the patients but decided to split it into two separate priority queues. We also utilized a lot more vectors than expected, for example we used a vector to contain both the doctors and the nurses of the hospital as well as a vector for all the residents in the town. Inheritance and Polymorphism was also later added: doctor, nurse and patient all inherited from a person class, while initially we just had a patient class. During implementation, we ran into problems with the clock and checking whether patients were done being served. It took a while to figure it out but we eventually overcame the challenge. Another short roadblock was figuring out how to use a multimap. Not only did it take time to decide that a multimap was a good data structure to use, it took time to figure out how to effectively use it as a patient record. We were unable to iterate through the multimap and display all that were treated but were able to access certain elements. Another problem we ran into was file output and how to combine different file data randomly. That was when we utilized a vector and random number generation. The problem that took the biggest toll though was getting and keeping high priority values for the patients. The patients were only receiving priorities 10 or lower and so were only treated by nurses. Really this problem was an extensive of a bigger pass by reference problem we believe. In the end we were unable to get patients to keep their higher priority they originally generated.

As mentioned earlier, we learned more about multimaps, like its purpose. We learned that multimaps were good for multiple key entries and accessing single elements by key (name in this case). We learned more about clock use to keep things running by ticks and checking values to update the progress of the hospital. Some of us had not used polymorphism a lot before so this project gave practice and mastery of it.