**CS273 Final Project Summary Report – Emergency Room Sim**

Cameron Rutherford

Average Patient Visit Time

Using a fixed arrival rate of 15 patients per hour, averaging out 20 different runs of the program (using a for loop to run the simulation 20 times and calculate the result), the following average times were found for each combination of Doctors and Nurses:

|  |  |  |
| --- | --- | --- |
| Number of Doctors | Number of Nurses | Averaged Average Wait (min) |
| 1 | 1 | 9.1516 |
| 2 | 1 | 8.4957 |
| 1 | 2 | 8.0419 |

The following graph comparing wait time to arrival rate was created using the same method of averaging out over the course of 20 runs. The chosen fixed amount of nurses and doctors was 2 nurses and 2 doctors.

Changes from initial design to implementation

The first main thing that changed from my initial implementation was the separation of the incoming patient queue into two separate priority queue. This allowed for easy search of all of the patients above 10 in illness priority, as well as allowing for a Boolean check of whether there were any to begin with.

Another thing that changed was how the Physicians went about storing the patients they were treating. Initially I tried using pointers to do so, but in the end I found it was easiest just to store them as regular values, and use a different method checking to see if a Physician was busy. This ended up being setting an internal value of the treatment time to 0 if the physician was not busy, working out to be quite a nice solution.

The final thing that changed, and coincidentally proved to be the most difficult was the method of updating the emergency room. The pseudocode changed completely, as there were a lot of unexpected intricacies of the algorithm that I did not initially consider, and so things had to be adjusted multiple times in order to maintain proper operation of the function.

Some other things to note that I did not consider in my initial implementation was using type\_id to check for the type of Physician, the way about which I would generate the random outcomes that were necessary, and the order of insertion of the different Physicians into the vector. Type\_id could have been avoided by using more polymorphism, and it ended up being too late to revert the initial fix that was implemented at the time. The way of generating random events was not considered remotely upon the initial design, and so a lot of smaller decisions had to be made, more particularly regarding how names were generated. Finally, the order of insertion of Nurses and Doctors into the Physician \* vector mattered a lot as it would have a significant impact on efficiency if done in the wrong order, and was something hard to see even in the final implementation, let alone the initial design.