Justice Martinez, Oscar Duran, and Andrew McNeill

Hospital Simulation – Final Summary

CS273 Data Structures – SEC 1

Our final project for Data Structures was to build a working hospital simulation. Our original design implemented the standard requirements specified in the assignment rubric including reading in a text file of names, prioritizing patients as the entered the hospital, treating the patients with the appropriate medical staff, and then displaying all of the records saved about the patients. All of these processes took place while a clock incremented the time as each moment passes.

Our different uses of data structures were incorporated in the priority queues, a map, and vectors. How we organized the patients was in a map, by mapping the patients names with the other records kept like number of visits and injury severity. As the simulation progresses, the list of first and last names goes into separate vectors fname and lname. When patients arrive, based on the specified arrival rate, a random number is generated to pull a name at that spot in the vectors to make a random patient. The use of priority queues was the most vital to the functionality of the program as they were the means by which patients were prioritized as they move through the simulation. Other queues were integral to the program, HospitalQueue was the first as the patients arrived. Then, depending on the number of doctors, number of nurses, and their availability (not currently treating a patient), the patients move to the appropriate priority queue, LowPriorityQueue or HighPriorityQueue. The appropriate queue is based on the severity level that was randomly assigned from pre set rates. After put in the right queue if/else statements then determine which staff should be treated, low priority patients can be treated by both available doctors and nurses while high priority patients can only be treated by doctors.

Some problems we encountered in writing the program often came in the clarification of syntax. The code for reading in the residents\_273ville text file using the ifstream lacked the extra backslashes that is necessary for the compiler to recognize the location of the text file, it was simple to fix once we were made aware of it. Another issue that occurred was when we output that stats of the simulation. For the longest time the program only showed two or three patients going through the program, with their number of visits adding up to the total amount, but still not what would happen in a realistic simulation. What was the problem was that the random number generator is seeded by the current time, not the clock used in the simulation. Changing it to be based off of the simulation clock gave us the wider range of names that we were looking for.