Nate Esrey

Kelly Olivier

Hospital Simulation Final Specs

Requirements Specifications:

* Create a working Hospital Simulation for the residents of CS273ville that is efficient as possible, maximizing all functions of a hospital in the quickest way possible.
* Patients that come into the Emergency Room are organized based upon their condition, one to ten in one queue and ten to twenty in another queue.
  + Doctors are able to take one to twenty, but are prioritized on those with conditions over ten. Nurses can only treat those with one to ten conditions.
  + Nurses take up to ten minutes with a patient while doctors take up to twenty minutes with treatment.
* Simulation in the E.R runs one week time or 10,080 minutes.
* User is allowed to input the rate at which patients enter the E.R with the number of doctors and nurses currently working the E.R at that time
* Simulation will: calculate the average patient wait time and be able to hold a record of each patient from cs273ville, allowing the user to go back and search a name with the patients’ past visits.

Pseudo-Code Program:

prompts user for number of patients per hour

User: user inputs number of patients

User: enters inputs out of range

Program: throws exception

Program: prompts user for number of doctors per hour

User: enters inputs out of range

User: inputs number of doctors

Program: throws exception

Program: prompts user for number of nurses per hour

User: inputs number of nurses

User: enters inputs out of range

Program: throws exception

Program: shows stats, and prompts user to choose from menu.

User selects option number one

User selects out of range

Program clears input and displays menu again

Program displays all patients treated, shows menu again

User selects option number two

Program prompts user for patient name

User enters patient name

User enters invalid name

Program: throws exception

Program displays that patients record, displays menu again

User: selects option three

Program ends the program

EmergencyRoom::medicsFree()

Test to identify if a doctor is free

If not free, return false

If free, return true

EmergencyRoom::doctorsFree()

Use number of doctors that the user has entered

Test to see if there is a free doctor that is not being used from doctor queue

For number of medics available, determine which ones are doctors

Increment number of doctors

If a doctor is busy, increment test to show that doctor is busy

If test = number of doctors, return false

Else, there are free doctors available

EmergencyRoom::updateDoctors()

For all unsigned number of medics that are in the E.R

If all medics are busy

Update clock, record the patient that was treated, how long they were in the E.R, and other record information that needs to be recorded

While

Patients over 10 and under 10 are in the E.R queue

Check for free medics

Assign patients that are free into the respectable priority queues, whether that is doctor or nurse.

If patients in queue that do not have over 10 priority

Assign patients to medics since doctors that are free can take under 10 only if there are no over 10 patients

EmergencyRoom::EmergencyRoom(int numDoctors, int numNurses, int arrivalRate) : arrivalRate(arrivalRate), numTreated(0), totalwait(0), clock(0))

Build database for emergency room that includes user input and calculations from the random generator

Create new nurse and new doctor for the numbers input by user

EmergencyRoom::Update()

Update clock

If new patient come in

Place patient in the correct queue based upon priority level and time spent in Emergency room waiting

Increment the visit count for that patient to be “saved” in records

Update doctors()

EmergencyRoom::ListNames()

Iterate through names from CS273ville

Keys for the iterator are the names of individuals

EmergencyRoom::getRecord

Map

If name is not it, no records found

Else, name found

Output patient name, number of visits, and illness priorities.