Algorithm Design

Dr. Rouse

Ancel Carson

Simulation Group 10

10/9/2018

Overview:

This algorithm design covers three main pieces of the code. Moving patients from their queue to an open room, deciding which queue has the patient with the highest priority, and the output with all necessary information

Used Data Types:

* A queue array that holds objects from the Patient class
* An integer that holds the number of rooms chosen by the user
* An integer that holds the number of patients still in the waiting room
* An integer array to hold the number of patients serviced per rating
* An integer array to hold the total wait time of patients serviced per rating
* A boolean to control driving while loop
* A boolean to control if more patients can enter program
* A Patient array to act as rooms for Patients
* A StopWatch object to keep track of time in program

Ending program steps:

* Check if StopWatch says the program has been running more than 10 minutes
  + Calls first half of output
  + Tells program to no longer create patients
  + Checks if there are still patients waiting in queues
    - Calls second half of output
    - Ends while loop

Moving Patients:

Method receives queue array, StopWatch object, Patient array, array of patients per type, array of total wait time per type

* A loop will run a number of times equal to the number of rooms the user specified
  + An if statement will catch if the current room is empty or not
    - If a patient is in queue 4 they are immediately added to the current room
    - If queues 1, 2, or 3 have a patient they are check for priority
      * The patient out of the three queues that has the highest priority is placed in the current room via a switch statement.
    - If a patient was added, depending on their rating, the total number of patients of their type and the total amount of time that type of patient waited is added together.

Getting priority:

Method receives the queue array

* For loop retrieves first patient in each queue
  + The priority is retrieved from each of the patient objects
    - A single patient is selected based on which one has the highest priority
      * Highest priority is always queue of rating 4s
    - The rating of the patient is returned to correspond with the correct queue

Output:

After 10 hours:

Method receives number of patients per each type, both waiting and serviced, total sum of wait times for each type, and number of empty rooms

* Four loop that runs 4 times, one for each queue
  + Displays to screen output text and total wait time of specific rating group divided by number of patients serviced in same group
* For loop that runs 4 times, one fir each queue
  + Displays to screen number of patients still in each queue
* Display to screen number of rooms that are empty

When program ends:

Method receives additional time needed to finish program and number of patients that were serviced in that time

* Displays to screen time to finish and number of patients