CS 150: Project I - Restaurant Management

Version as of: 12:42 Wednesday 18th February, 2015
Due: 12:05am, Monday March 9, 2015

Introduction

Project Description

One of the problems that restaurant managers have to deal with is managing the wait list, particularly at popular restaurants. The example restaurant that you should consider is a ramen noodle restaurant called *Ippudo* in New York city. The restaurant does not take reservations. Customers show up and put their name on a wait list and they are called when a table is available.

Your project is to design a program that will help them to manage the wait list under the following conditions:

1. the restaurant is open from 6pm until 2am, i.e., customers are not seated after 2am or if it is predicted that they will be seated after 2am
2. the restaurant has a seating capacity broken up into the following areas:
   (a) bar seating for approximately 20,
   (b) an oval seating area of 30,
   (c) 5 tables with seating capacity of 4 each
   (d) 5 booths, each with the capacity of 8
3. customers can be broken up into two groups:
   (a) small groups of 1-4 persons: The mean arrival time is 20 secs with a standard deviation of 10 secs. Each group will spend 45 mins time at dinner (standard deviation of 10 mins).
   (b) large groups of 5-8 persons. The mean arrival time is 30 secs with a standard deviation of 15 secs. Each group will spend 60 mins at dinner (standard deviation of 15 mins).
4. Upon arrival, each group of customers is given an estimated seating time.

The goal of the simulation is to determine the seating strategies that will (1) maximize revenue, and (2) minimize the maximum wait time for any group. At the same time, you wish to give reasonably accurate wait times to each arriving group. So you will want to consider the effect of your seating strategy on the wait times.

To see how to generate random numbers with a Gaussian (normal) distribution see `http://www.javapractices.com/topic/TopicAction.do?Id=62`.

Analysis

Your simulation should help you to answer the following questions (and more):

1. Which seating strategies will maximize the revenue? You should have at least 3 seating strategies, preferably 5.
2. For each seating strategy, how good is your estimation (of seating time)? How is the accuracy effected (or not) by the length of the wait time?
3. For each seating strategy, what is the longest wait time to be seated
4. Given a seating strategy, when should customers be turned away? (This answer should depend on the size of the group). If the estimate is incorrect and the customers are still not seated by 2am, they will be compensated with the value of 4 meals for each person.

Your conclusions and analysis should be supported by data from your simulations.

Constraints

The following constraints apply to the project:

1. The project is to be completed individually. The only person you can consult is the instructor.
2. Each configuration of parameters should be run *at least* 5 times with different random seeds to obtain an ”average” value.

Simplifications

You can simplify your project in one (or all) of the following ways:

1. Dedicate each type of space to groups of a particular size.
2. Have only two types of seating space.

Grading

Your project will be graded on the following criteria (assuming the program compiles and runs):

1. correctness of the program
2. documentation (methods and classes) including javadoc
3. unit testing
4. object oriented design
5. quality of the simulation and analysis

Submission

Your submission on the due date will be composed of the following:

1. source files (*.java) that are commented and have javadoc directives
2. test files (Test*.java), one test file per class
3. a README.txt file that contains instructions on how to run your program
4. a draft of the project report (see project report guidelines) - 50% of the report grade is assigned to this. The remaining 50% will be given to a final report. The final report will be due several days after the draft is corrected and returned.