CMSC22: Object Oriented Programming
1st Semester AY 2013-2014

WX-6L: MasageBang! Massage and Spa Simulation

Description: An massage and spa simulation using the Java programming language.

Group Size: 2-3 members only.

Overview:

- The user starts out using parameters (including capital and initial available rides) that should be read and processed from a serialized file. Only a single pair of massage counter and spa is available at start.
- Each spa and massage pair can be upgraded to handle at most 5 people at any given time.
- Up to 10 spas and massage counters can be bought.
- Guests will be queued outside the establishment. The queue will increase in size after a specified interval (5-10s).
- Clicking on a spa removes a number of guests from the queue, depending on the capacity of what the ride can handle.
- Guests stay in the spa for 5-8s. Guests automatically leave the spa at the end of the specified time and MAY OR MAY NOT proceed to a vacant massage counter. Guests will enjoy a massage service for 8s and then leave.
- For each guest that visits the establishment, there is a fixed fee that will be paid upon the end of the spa and/or massage service.
- The establishment will incur maintenance costs which the user will have to pay to keep operations going.
- The game ends when the user either runs out of capital to spend or when the user quits.
- At the end of each run, details should be printed neatly to a report in a the form of a text file.
- Occupied spa and massage counters should be distinguishable from those on standby. The number of guests per spa and massage counter should also be visible.
- Time spent riding on each spa and massage counter should be clearly visible.
- Guests introduced to the waiting queue must be randomized: They could either be a member or a non-member.
- Guest queues are represented by a counter on screen.
- All activities executed within the simulation should be

Details:

- A data container in the form of the "Parameter" class will be provided with the specifications.
- An input file called "parameters.ser" containing the serialized instances of the Parameter object will be provided with the specifications.
- The following parameters should be expected in the "parameters.ser" file:
    - Starting capital
    - Starting spa and massage counters available
    - Rates per spa rental per guest
    - Rates per massage service per guest
    - Maintenance cost per second
    - Spa cost / Massage counter cost
    - Upgrade cost per spa
    - Upgrade cost per massage counter
    - Initial size of queued guests
    - Rate of queued guests per minute
    - Membership discount
- The following metrics should also be present on the interface, and should also be printed at the end of a run:
    - Current cash
    - Number of current spa sessions
    - Number of current massage sessions
    - Total cost paid for maintenance
    - Average earnings per minute
    - Total sales per second
    - Sales for spa sessions
    - Sales received for massage sessions
- Provide a quit button that will end the session and print a report containing the variables specified in the previous item.

Technical Specifications:

- Separate each GUI element in your application whenever possible.
- Do not unnecessarily cram classes for extended/customized listeners, swing components and interfaces.
- Extend the "SNMStat" class and implement the "SNMStatInterface" interface (to be provided with the specifications) that handles the update mechanism fo the metrics display mentioned above.

printed on an on-screen massage and spa log (who just queued, who's checking out, how much was eared, etc.)

- The number of seconds before the next queued car should be shown as well.