ELEC 2543 Object-Oriented Programming and Data Structures

Assignment 1

Due Date: Mar 20, 2017

Overview: This assignment simulates a simplified Monopoly game. The number of players, the initial cash amount each player has, and the size of the path are parameters passed to the program.

The Simplified Monopoly Game (SimMP)

*Overview*

There are *n* players, each with the same amount of cash, *c*, when the game starts. They all walk on a circular path with length *size* that composes of a number of lots. In each round, they all make a move. The distance to move is determined by rolling a die. Depends on what location a player moves to, the player may buy a lot, have to pay rent, or get some cash. The game ends when at least one player is broke (with –ve cash balance) or the game has proceeded for 20 rounds. In this SimMP game, there is no “bank”.

*size*, *n*, and *c* are passed as parameters to the game driver (main method). That is, the three parameters are provided when execute the game.

*The Path*

The path composes of a “GO” lot, and *size*-1 other lots on them. The “GO” lot is numbered as zero, while the other lots are numbered from 1 to *size*-1. It is a circular path, and the players move from a smaller numbered lot to a higher numbered lot. When a player is now at lot *size*-1 and it moves one step, it will be in the GO lot.

Except the GO lot, a player can be the owner of the lot by buying the lot. A play can buy a log only if the lot is not owned by any other player.

*The Players*

There are *n* players, and their ID’s are 0, 1, 2, …, *n*-1. Each player starts at the GO lot. Each has *c* dollars in the beginning.

*In each Round*

In each round, each player, from small ID to large ID, takes turn to make a move. In each move, the player first rolls a die. The range of distance is [1, 6]. It then advances to another lot according to the face value of the die.

* If the new position is the GO lot, the player gets 100 dollars.
* If the new position is not the GO lot, and the lot does not have an owner, the player can decide whether to buy it or not by rolling a die if he has enough money. The chance of buying the lot is 50%. If the player decides to buy the lot, the price of the lot is randomly selected from 100\**k* dollars where 1 ≤ *k* ≤ 5.
* If the new position is not the GO lot, and the lot is not owned by the player, the player has to pay rent to the owner. The rent is 10\**k* dollars where 1 ≤ *k* ≤ 10 (*k* is randomly generated). That is, the cash amount of the player is reduced but the owner has more money. The rent has to be paid no matter whether the player has enough money or not.

*Game Termination*

The game ends when either one of the following conditions is satisfied:

* one or more players have negative cash balance after each making a move in that round
* the game has proceeded for 20 rounds

*Output*

Before the game starts and after each round, the positions and the cash balances of the players have to be printed out. The ownership information about the lots should be provided below the player position information. The player ID is displayed for the lot.

The path is printed in a linear manner from the GO lot, with the player position indicated by the player ID. A separate line for each player.

In each round, apart from the above information, the rent/money each player pays/gets should be printed out as well. Sample outputs are available in Moodle. The first line of the file is the command for running the program. It tells *size*, *n*, and *c*.

Handin

**You must name your driver program class file as SimMPGameDriver.java.** You can develop other class files and name them by yourself. You should submit the following:

1. ALL java files that are needed for running your application. The main method must be in class file SimMPGameDriver.java. You can zip all the files for submission or you can upload the file individually.
2. A pdf file that lists the instance variables of each class in your implementation. Briefly explain what each instance variable represents in your classes. To enhance readability, you have to list the class names and instance variable types and names clearly.