CS 151: Object-Oriented Design

Fall Semester 2013

Department of Computer Science
San Jose State University
Prof. Ron Mak

Assignment #6

Assigned: Tuesday, November 26
Due: Friday, December 6 at 11:59 pm
Team assignment, 100 points max

Multithreaded RPS game server

Create a multithreaded Rock-Paper-Scissors **game server** that can play multiple games simultaneously, each with a separate human player. (Of course, the same human player can play in each game.)

In Assignment #5, you created a GUI "game window" that allowed a human to play a game against the computer. For this assignment, create an additional "server window". From this server window, **open one or more game windows** that each plays a separate game with a human. For example, you can press a button in the server window to open a new game window. You can limit the number of concurrent game windows to three.

The server window should display a running tally of the number of wins and ties of each game, and the total numbers of wins and ties by the human and computer players across all the games.

To support the "smart" computer algorithm, record the choice sequences from all the games. To support the algorithm even further, **write your record of choice sequences out to a disk file** whenever you terminate your server, and read it back in whenever you restart. Then your program will accumulate more and more sequences over time and the computer should get smarter and smarter.

Hints

Start by putting your code from the previous assignment into a Java thread. Create the server window and start up two or more threads to make sure you can play several games simultaneously, each with a separate record of choice sequences.

Then do the running tally in the server window and share the choice sequence record across all the games. Identify the critical regions in your threads. Add the necessary thread synchronization (see Chapter 9 about locks and conditions, to be discussed in class).

What to turn in

Each team should create a zip file containing all the Java source files of its application. Zip the **src** subdirectory that NetBeans creates when you create a new project. Only zip the **src** subdirectory.

Email the zip file as an attachment to ron.mak@sjsu.edu. CC all your team members. Do not email any executable files because some mailers will reject the entire message. Your subject line should be:

`CS 151 Assignment #6` ***team name***

Your email message should include instructions on how to run your game server.

This is a team assignment. Each member of the team will receive the same score.