Homework #2:
Rock Paper Scissor Simulator

1 Objectives

This homework is designed to give you experience using the Strategy Pattern in a fun context. While the Strategy Pattern is a pretty simple pattern to apply, I hope that this project provides you an opportunity to explore the possibilities for different implementations.

2 Background

Before we get to the objectives, think for a moment about the game Rock Paper Scissors. When two people play Rock Paper Scissors, the outcome of the game is seemingly random. If the game is fast paced, players have little time to think about playing the game with respect to strategy. While a savvy or smart player might be able to recall the last seven to ten moves thrown by their opponent, they will quickly be overwhelmed if they play many games in succession.

On the other hand, some players might devise a simple strategy that involves remembering the last few moves their opponent threw. However, with a computer, possibilities for what to throw and how to decide what to throw abound.

3 Overview

For this homework, you will implement four strategies for playing rock paper scissors. The first three strategies will be of your choosing. They can be arbitrarily simple, or incredibly complex. They can be sequential, stochastic or even more complex. The fourth strategy will be a composite strategy, so to speak. It should dynamically choose between the three strategies to use when playing the game while the game is being played.

After the due date, we will be having a bracketed tournament between all valid submissions. We will do one level of the bracket each day. The person who wins the tournament will get a special surprise. The contender that beats the instructor will also get a special surprise.

4 Requirements

The requirements for this project are very simple and straightforward:

- You must provide three strategies, named (yourUsername)(nameOfStrategy). (See the provided code for examples) following the Standard Java Naming Conventions.
- You must provide a fourth, composite strategy, that decides at runtime which strategy to use. This must be named (yourusername) CompositeStrategy.
Each of the four strategies must implement the RockPaperScissorsInterface.

Each of the four strategies must provide a toString() method that identifies the name of the contender and the name of the strategy.

You must provide an implementation of the RpcTournament driver class that instantiates a match between all four of your strategies (16 total).

With your submission, you must provide a readme.txt in the project folder that provides an overview of each of your strategies, which one the most matches, which implementation you would like to use for our in-class tournament, and the output of your RpcTournament application class.

You must also good programming practices.

- You must use proper naming conventions
- You must provide adequate documentation in the form of comments.
- Your code must not crash, and exit gracefully in the face of errors and exceptions (there should be none).
- Your code must provide user-friendly output (don’t call me a fool).
- Your code should run suitably quickly. While I am not imposing a time limit, please do not make a set of 10,000 games take more than a minute.
- You may not use a Random number generator to decide what move to throw. If your code involves using a random number generator, please consult me first.

You may utilize a shell project that I provide on my website.

5 Submission Instructions

This project will be due at 11:59:59 PM, Friday March 14th.

Please name your project yourusername_homework2. When your project is complete, archive your project (not the source) as yourusername.zip and submit your work to handin.cs.clemson.edu. Use the handin command to submit your work. Please submit your Eclipse project as a .zip archive using the http://handin.cs.clemson.edu website.

6 Grading

This is an easy homework. Your submission will primarily be graded on implementation details, documentation, style and correct application of the Strategy pattern. The quality of your strategy, the number of wins or your placing in the tournament will not be taken into account for your grade.

7 Collaboration

You may not work with a partner; however, I encourage you to test your strategies with your fellow classmates; unless you feel you are losing a competitive edge.