CECS 277 – Project 2 – Rock Paper Scissors Mind Reader –

Each round the computer will make a prediction of which type of throw the user will choose, Rock, Paper, or Scissors. Based on this prediction the computer then chooses a throw that will beat the user's choice. Rock beats Scissors, Scissors beats Paper, and Paper beats Rock. The user selects their throw and then the computer's choice is revealed. A point is given to the winner and the game repeats until the user quits.

On the first few rounds the computer obviously does not have enough information to make a prediction, so it randomly makes a choice. But after that, it should use the user's history of choices to determine what is going to be the most likely choice.

At the end of each round, the user's choice plus their previous three choices are stored into memory. Use a HashMap to keep track of the different patterns (keys) that have come up, and the number of times the user has used that pattern (values). Keep a string of the last four choices, add the new choice to the pattern, and discard the oldest.

Create a Pattern class to create Pattern objects to use as the keys of your HashMap. The Pattern class should use a String to hold the pattern. Have a get method, but no set, since the key should never change once it is stored in the HashMap. Override the hashCode() and equals() method.

Create a Computer class that stores Patterns in the HashMap, and makes predictions.

A prediction is based on the patterns, if the pattern of the past four choices is in the HashMap, then examine how many more times the user has chosen Rock vs. Scissors vs. Paper when using that pattern, then base the prediction on this. If the pattern is not in the HashMap, then just make a random guess as a last resort.

Display the score and the percentage of rounds the computer has won after every round.

Begin with storing just the four character patterns and see how well the computer does. Experiment by replacing it with other length patterns. You can also try combining pattern lengths and then basing the prediction on the one with the highest priority. Implement your best outcome to destroy your predictable human opponents.

When the user quits, give them the option to write the contents of the HashMap to a text file, so that if the game is played again the HashMap can be reloaded.

When starting a new game, check to see if there is a save file, if not, start with an empty HashMap, if it does exist, ask the user if they would like to use it. When reloading the file, return the most common pattern in the map so the predictions have something to start with.