

## CECS 277 – Project 3 – Computini the Great –

Play the game at the following link: <http://www.nytimes.com/interactive/science/rock-paper-scissors.html>. When you're finished playing, click the "See What the Computer is Thinking" button to learn what the computer is doing to predict your throws.

**Part A (10 Points)** – Make a version of this game where there are only two inputs: 1, and 2 (or A and B, + and -, T and F... whatever you'd like). Use a sentinel to allow the player to quit.

On every round, the computer makes a prediction of what value the user will choose, but keeps it hidden from them. The user then enters their choice. If they are the same, then the computer gets a point, otherwise the player gets a point.

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.

After each round is over, the user's choice along with their past four choices are stored into memory. Use a hashMap to keep track of the different patterns that have resulted (the key), and how often the user chose A and B following that pattern (the value). Keep a running pattern of the last four choices, and add the new choice to the pattern. The previous fourth choice is now thrown away, the others are moved up, and the new choice is stored as the first.

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 A over B following that pattern and base the prediction on this. If the pattern is not in the hashMap, then just make a random guess.

Begin with storing just the four item pattern and see how well the computer does. Experiment by replacing it with a three item pattern, and then a two item pattern. See which one performs best, then try combining them by keeping track of all of them and giving the prediction from the best one priority over the others. Implement your best outcome, and hopefully your computer player can easily win, sometimes by a two or three to one ratio.

**Part B (15 Points)** – Using the same principles as above, implement the Rock, Paper, Scissors game. But now instead of trying to tie with the user, the computer should predict what the user will throw, and then throw the weapon that beats it. When the game is over, write the pattern memory to a file, so that when the game is played again, the information can be loaded and it now has past information to start with. Maybe allow the user to choose which version of the computer they would like to play, beginner or veteran.