

COMP 208: Computers in Engineering

Winter 2019 – Assignment 4

Rock, Paper, Artificial Intelligence!

Integrity

These assignments are to be done individually. You can collaborate on understanding the problem but you must write the solution **individually**. Your submission might be subject to plagiarism detection software.

Due Date

Assignment 4 is due on March 14 at 11:59PM. The cutoff is automated and is exactly at this time. Assignments submitted within the next hour will be considered late. After that time they will not be accepted.

Introduction

Programs are typically designed to solve problems with pre-determined solutions. However, some problems offer far too many different possibilities, too numerous for all of them to be evaluated. Some of these possible solutions may be better than others but not all can be explored. In such cases, it may be simpler to create an algorithm which allows the program to learn from its mistakes and progressively become better at accomplishing a given task. This is the basis of artificial intelligence and machine learning.

The goal of this assignment is to create a Rock, Paper, Scissors simulator, which learns from previous games and becomes progressively better. The simulator begins with no knowledge of the rules of the game. It will learn how to play as the game progresses.

If you are unfamiliar with Rock, Paper, Scissors, visit:

<http://en.wikipedia.org/wiki/Rock-paper-scissors>.

Assignment

Your task is to write a program which asks the user for a move (Rock, Paper or Scissors) and then, based on the user's move, chooses a response (Rock, Paper or Scissors). This process continues until the user decides to exit the game.

If the program "knew" the rules of the game, it would be trivial to make the program choose Paper when the user plays Rock, and Scissors when the user plays Paper, etc. Instead, your program should play randomly at first, and change its play based on previous games, through reinforcement learning.

After every game, the program should output the user's move, the computer's move, the outcome (win / draw), and the probability of the computer's move.

Example Output:

```
Game #1
User Move: Paper
Computer Move: Rock
User Wins.
Probability of Move: 33.3%
```

```
Game #2
User Move: Paper
Computer Move: Paper
Draw.
Probability of Move: 34.4%
```

Methodology

Reinforcement learning uses a reward and punishment system to weed out bad moves and select good ones.

In Rock-Paper-Scissors, every combination of moves can be represented by the following table of values, in which one dimension is the User's move and the other is the Computer's move. (ie. V_{rr} is a value associated with the

rock-rock combination). You should use a two dimensional array to represent these values. The values are numbers assigned to indicate how “good” the move is.

	Rock	Paper	Scissors
Rock	Vrr	Vpr	Vsr
Paper	Vrp	Vpp	Vsp
Scissors	Vrs	Vps	Vss

Initially, every combination has the same positive, non-zero initial value (for example, 10), since every combination is equally likely. This value is then increased (ex: +1) or decreased (ex: -1) based on whether the computer wins or loses (this is analogous to a reward or punishment).

Every time the user makes a new move, the computer must randomly select between 3 possibilities (Rock, Paper or Scissors), with the probability of selection being (P_{ij}) (in which i is the user’s move and j is the computer’s move):

$$P_{ij} = \frac{V_{ij}}{V_{ir} + V_{ip} + V_{is}}$$

Thus, the program must generate a random number (using `rand();`) and based on the value of this random number, should select one of the 3 choices. Then, it needs to assess whether the computer won or lost, and give itself a reward or punishment based on the result (which changes future probabilities).

Requirements

Your code must meet these requirements:

- The program must be written in **C**
- Use **sensible** variables names
- **Comment** and **indent** your code
- Use at least **1 function** (besides `main( )`), for example to select the move or to check the winner.
- Submit **only** the .c file. **Don’t** submit the .exe (name your source file A4_123456789.c where 123456789 is your ID)
- Your artificial intelligence should not attempt to take over the world

If any of the above requirements is not respected you will lose marks.

Acknowledgement

Thanks to Daniel Peykov for his work in preparing this assignment.