CS210 – Machine Problem 2

RPS AI
20 Points
Assigned: September 7, 2017
Due: September 14, 2017

Many students get into computer science to write video games. For this problem we are going to develop a computer game that plays Rock, Paper, Scissors (sorry, no Lizard or Spock) against a human opponent. In the game of Rock, Paper, Scissor, players simultaneously choose one of the eponymous elements by flashing a hand gesture and comparing the results. Here is how a winner is chosen:

- Rock beats Scissors (because it breaks them).
- Scissors beats Paper (because it cuts it).
- Paper beat Rock (because it covers it and makes it sad).

The ideal strategy for an RPS player is to randomly choose every time. However, humans are bad at being random – a fact which we can exploit to produce an Artificial Intelligence that will often beat humans over the long run. It exploits the following observations that have been made of many human players:

- If a player wins, they will tend to stick to the same element.
- If a player doesn't win, they will tend to switch to the next element in "clockwise" order (i.e., Rock will switch to Paper. Paper will switch to Scissors. Scissors will switch to Rock).

Program Design

This program is going to have many tasks that it needs to accomplish. We will develop one function at a time, solving each smaller task, and then combine them all at the end. I will give you the specification of each function. You should write the function and then test it with a `main` program that ensures that the function works by itself. Once a function is working, move on the to the next function, which will require a different `main` function to test it. Once we have all the individual functions working, we will finally write the final `main` that will actually play the game.

Print Greeting

Function Name:	`print_greeting`
Inputs:	None
Returns:	None
Locals:	None

This function simply prints out the greeting and taunt for the user at the beginning of the program. See the sample run for the exact text.

Print Score

Function Name:	`print_score`
Inputs:	`human_score` (integer) `computer_score` (integer) `ties` (integer)
Returns:	None
Locals:	None

This function prints out the current number of human wins, computer wins and ties. All three of the values are passed as parameters to the function. See the sample run for the format of the score output.

Get User's Pick

Function Name:	`get_pick`
Inputs:	None
Returns:	The character the user picked converted to lowercase
Locals:	`ch` (character) - The character from the user `bad` (boolean) - true if the input character is invalid

This function's task is to prompt the user to enter either R for rock, P for paper, S for scissors, or G if they want to give up (quit playing). Here are the steps to accomplish this:

1. Do the following...
   (a) Prompt the user for input
   (b) Read the user's input using: `scanf(" %c", &ch);` (Note: There is a space before the % symbol. This will make scanf ignore any whitespace input characters)
   (c) Convert `ch` to lowercase using the `tolower` function in the `ctype.h` library. (You will have to include it at the top of your file)
   (d) Set `bad` to true if `ch` is not 'r', 'p', 's', or 'g'. Set it to false otherwise.
   (e) If `bad` is true, print out the error message shown in the sample run.
2. ...while `bad` is true.
3. Return the value of `ch` to the calling function.

REMEMBER! Test this function with a throwaway `main`.

Compute Match Result

Function Name:	`compute_result`
Inputs:	`human` (character) - What the human picked `computer` (character) - What the computer picked
Returns:	-1 if the human won 1 if the computer won 0 if a tie
Locals:	none

This functions task is to return a number that represents the outcome of the match. First, it prints out what each player has chosen (e.g. Human picks Rock, Computer picks Paper). Then it will return a -1 if the human wins, a 1 if the computer wins, and a 0 if the match is a tie. It will do this all of this with a series of `if` statements, `printf` and `return` statements.

Compute Next Pick

Function Name:	`compute_next_pick`
Inputs:	`human` (character) - What the human picked `computer` (character) - What the computer picked
Returns:	The character which is the AI's next play
Locals:	none

Similar to the previous function, this one will be a series of `if` and `return` statements. The two rules mentioned above can be described by the following table:

Human	**Computer**	**Next Move**
Rock	Rock	Scissors
Rock	Paper	Scissors
Rock	Scissors	Paper
Paper	Rock	Scissors
Paper	Paper	Rock
Paper	Scissor	Rock
Scissors	Rock	Paper
Scissors	Paper	Rock
Scissors	Scissors	Paper

Main Program

Finally, we arrive at the `main` function of our program. Now that we have all of the pieces

Function Name:	`main`
Inputs:	None
Returns:	0 - because `main` always does
Locals:	`human_wins` (integer) - number of human wins `computer_wins` (integer) - number of computer wins `ties` (integer) - number of ties `result` (integer) - the result of the current match `human_pick` (char) - Human's choice `computer_pick` (char) - Computer's choice

The steps to accomplish to main game loop look like this: (Most of the complicated stuff is in the functions that we've already written).

1. Initialize all the number of wins and ties to 0.
2. Initialize the computer's next pick to Rock (poor predictable computer, always picks Rock).
3. Print the Greeting
4. Start a loop (`while(1)`. Don't worry, we'll break out of the loop later. Inside the loop …
   (a) Print the Score
   (b) Get the human's pick and assign it to `human_pick`
   (c) If the human chose 'g'. Use the `break` command to exit the loop.
   (d) Compute the result of the match and store it in `result`. (Note: This will also print out the result of the match).
   (e) If the result is negative - print "You Win!" and increment the human's score.
   (f) If the result is positive - print "I Win!" and increment the computer's score.
   (g) Otherwise, print "Tie!" and increment the number of ties.
   (h) Compute the computer's next pick and store it in `computer_pick`

5. return 0 to exit the program.

Congratulations on writing your first computer game. Note: If you let your friend play it, make sure they don't know how it picks its next move. Otherwise, it is extremely easy to beat.

Sample Run

```
Welcome to the RPS AI
I will dominate your puny human brain!
The current score is:
Human: 0
AI   : 0
Ties : 0
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:p
You chose paper
I chose rock
You Win!
The current score is:
Human: 1
AI   : 0
Ties : 0
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:S
You chose scissors
I chose scissors
Tie!
The current score is:
Human: 1
AI   : 0
Ties : 1
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:k
Bad input! Try again.
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:R
You chose rock
I chose paper
I Win!
The current score is:
Human: 1
AI   : 1
Ties : 1
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:P
You chose paper
I chose scissors
I Win!
The current score is:
Human: 1
AI   : 2
Ties : 1
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:p
You chose paper
I chose rock
You Win!
```

```
The current score is:
Human: 2
AI   : 2
Ties : 1
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:s
You chose scissors
I chose scissors
Tie!
The current score is:
Human: 2
AI   : 2
Ties : 2
Pick (R)ock, (P)aper, (S)cissors, or (G)ive up:G
```