Rock Paper Scissors

Matthew's Stats

- Time taken: 40 minutes
- Files: 7
- Lines of Code: 309 including whitespace and comments

What to Submit

- A zip file containing
  - Your .cpp and .h files that make up your solution
  - A CMakeLists.txt file that will generate an executable named **RockPaperScissors** from your .cpp and .h files
    - In your CMakeLists.txt in the add_executable line, make sure you have `add_executable(`**`RockPaperScissors`** `your_.cpp_files your_.h _files)` so the correctly named executable gets built
- Make sure to zip the files you want to submit and **NOT** the folder that contains the files. Submitting the folder with the files will cause your program to fail to build.

Restrictions and Requirements

- No global variables may be used
- Your submission must contain at least 2 or more .cpp files and one or more .h files
- Your submission must have at least 3 user defined functions in it in addition to main

Problem Description

You will be implementing the game of rock, paper, scissors. In this game, players randomly select rock, paper, or scissors. Rock beats scissors, paper beats rock, and scissors beats paper. If both players choose the same they choose again until someone wins. For your version, you will have a human player play against the computer. In addition, when the game completes the player has the option to play another game.

Additional Details

Input

Command Line Arguments

- The first command line argument is the seed to the random number generator
  - Optional
  - It will always be valid
  - It is an integer

Standard Input

- Will **NOT** always be valid
  - If invalid input is entered you should prompt the user again for input until valid input is entered
- When asked for a move, valid input consists of any of the following with 0 or more whitespace characters before or after it. Capitalization does not matter.
  - Rock: r, (r)ock, or rock
  - Paper: p, (p)aper, or paper
  - Scissors: s, (s)cissors, or scissors
- When asked if the player would like to play again, valid input consists of any of the following with 0 or more whitespace characters before or after it. Capitalization does not matter.
  - Yes: y, (y)es, or yes
  - No: n, (n)o, or no

Random Number Generator

- Use the `std::minstd_rand` random number generator
- If a seed is provided use the seed to the random number generator
- If a seed is not provided use `time(nullptr)`, the current time, to seed the random number generator
- To have the computer randomly select a move you should have the computer randomly generate a number between 0 and 2
  - 0 = Rock
  - 1 = Paper
  - 2 = Scissors
- There should be code covered in class to make the choice of this random number fairly easily for you, so make sure to use that

- Since different operating systems implement random number generators differently it is still possible that if you implement everything perfectly you will see different results on **YOUR** computer compared to what the solutions are. If you submit and you've done things correctly it will pass the test cases.
  - This happens most often with people who are running their code on MACs.
    - I think people that are running Cygwin or Linux should have the same results as those on Kodethon but don't quote me on this.
  - If you run your code on Kodethon and you've done things correctly your output will match mine. So if you think you've done everything correctly but your outputs are different than mine, run your code on Kodethon and see what you get.

Example

- Input has been underlined so that you can differentiate between what is input and what is output
  - You do not have to underline anything
- I've also provided an example executable named ExampleRockPaperScissors that you can run by doing `./RockPaperScissors seed` where seed is an integer
  - It is only guaranteed to run on Kodethon and may not run on your personal computer

Example 1

```
./RockPaperScissors 3
Enter (R)ock, P(aper), or S(cissors) for your move: bob
Enter (R)ock, P(aper), or S(cissors) for your move: rock man
Enter (R)ock, P(aper), or S(cissors) for your move: Rock
The ai played rock.
You and the AI both played rock.
Keep playing until someone wins.
Enter (R)ock, P(aper), or S(cissors) for your move: p
The ai played rock.
You win!
Would you like to replay the game?
Enter (Y)es or (N)o: Y
Enter (R)ock, P(aper), or S(cissors) for your move: s(c)
Enter (R)ock, P(aper), or S(cissors) for your move: (s)cIssOrs
The ai played scissors.
You and the AI both played scissors.
Keep playing until someone wins.
Enter (R)ock, P(aper), or S(cissors) for your move: p
The ai played scissors.
The AI wins :(
Would you like to replay the game?
Enter (Y)es or (N)o: n
```