KBTG Techtopia Codeoff

Instructions

Create a command line **rock-paper-scissors** game that prompts the player for a choice between `rock`, `paper,` or `scissors`; then randomizes the computer choice, and find out who's the winner. The program should keep track of the current round's winner, the player's score, the computer's score, and the number of games played.

Supported languages

1. JavaScript
2. Go
3. Java
4. Python

Setup

1. Visit `https://github.com/kbtg-techtopia-code-off/[REPO NAME]`
2. Clone the repository
3. Open the cloned project in Visual Studio Code
4. Make sure that GitHub Copilot is running in your editor by testing a short comment prompt: `// write a rock-paper-scissors game`
5. Get started writing code!

Please wait for the MCs to announce the start

Level 1

Challenge #1 - Write the rock-paper-scissors program

Create a file to write your rock-paper-scissors game with an extension of your choice.

Tasks

- The program prompts the user for an input between `rock`, `paper`, or `scissors` in the command line
- The program randomizes the computer choice
- The program outputs the round's result after the user submits in the terminal the choice as follow:

```
Please input your choice >>> paper

The computer chose: rock
The player choice: paper
The winner is: player
Wins: 1
Losses: 0
Games played: 1
========================

Please input your choice >>>

...
```

- After each round, the game prompts the player for the new choice and the game continues until the program is closed

Bonus

- The program checks for invalid user choice and asks the user to provide a valid input again. The program should not continue until the user provides a valid input.

Submission

- Capture the screenshot of the results and save it to the `/results` folder and name the file `1.jpg`. Example: `/results/1.jpg`
- Commit and push your code to Git

> Hopefully your Paper, Rock, Scissors game is working! Remember, GitHub Copilot is probabilistic so you may not get the exact same code suggestions as we did. If you're not happy with the suggestions, you can always press **CTRL + Z** to undo the changes and try again.

You're now ready to start the Level 2 Challenges to see how you can leverage the power of GitHub Copilot to solve more complex programming problems.

END OF LEVEL 1

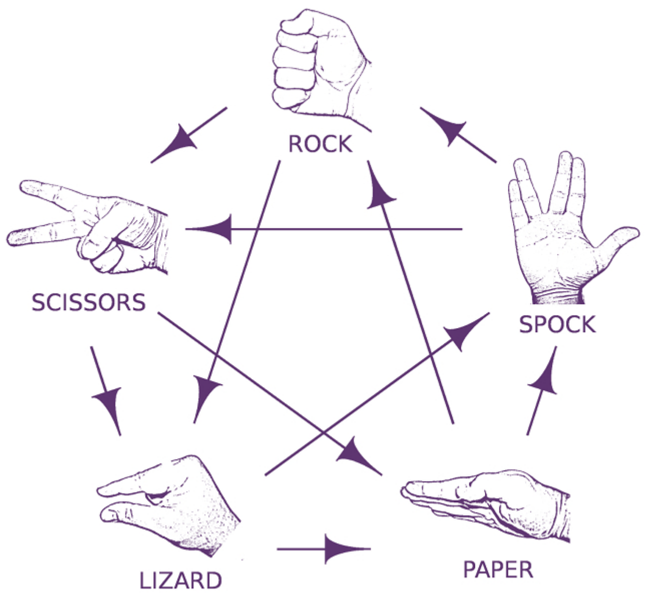
Level 2

Now you've had an opportunity to get started using GitHub Copilot, we have a number of challenges for you to attempt. Remember the goal here is not to test your programming abilities but rather, see how you can use GitHub Copilot to help you complete these tasks. Even if you've never done these programming tasks before, you may be surprised how Copilot can help you be successful with these challenges.

Challenge #2 - Adding Lizard and Spock to the game

Tasks

- Improve the rock paper scissors game by adding Lizard and Spock. Refer to this chart for the updated game logic



- Use the same CLI output from Chellenge #1

Bonus

Extra Kudos for a terminal interface that provides a list of options with keyboard input. Eg.

```
$ Choose your option:
1.  Rock
2.  Paper
3.  Scissors
4.  Lizard
5.  Spock
```

Submission

- Capture the screenshot of the results and save it to the `/results` folder and name the file `3.jpg`. Example: `/results/3.jpg`
- Commit and push your code to Git

Challenge #3 - Adding Unit Tests

Tasks

- Implement unit tests using using any testing module of your choice and capture the code coverage.

Bonus

- Try to aim for over 80% coverage 😃

Submission

- Capture the screenshot of the results and save it to the `/results` folder and name the file `3.jpg`. Example: `/results/3.jpg`
- Commit and push your code to Git

Challenge #4 - Adding a REST API

> 💡 Tips: You might want to create a new file or folder for this challenge to keep your original code from Challenges 1-2.

Tasks

- Turn your program into a REST API
- Sending a POST request (or json payload) should return a 200 OK response with the result in the body

  Endpoint `POST /` JSON Response - 200

  Request

  ```
  {
      "playerChoice": "paper"
  }
  ```

  Response

  ```
  {
      "computerChoice": "rock"
      "playerChoice": "paper"
      "winner": "player"
      "wins": 1
      "losses": 0
      "gamesPlayed": 1
  }
  ```

- Should return a 400 BAD_REQUEST response error message in the body

  Endpoint `POST /` JSON Response - 400

  Request

  ```
  {
      "playerChoice": ""
  }
  ```

  Response

  ```
  {
      "message": "YOUR REASON HERE"
  }
  ```

Submission

- Capture the results and save it to the `/results` folder and name the file `4.jpg`. Example: `/results/4.jpg`
- Commit and push your code to Git

END OF LEVEL 2
