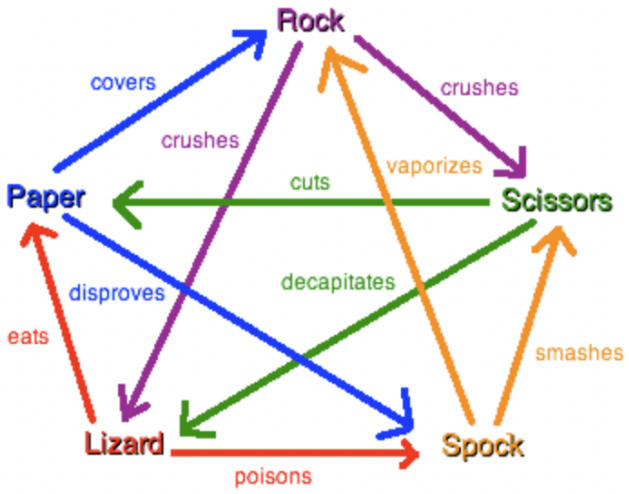
Rock-Paper-Scissors-Lizard-Spok Game

Introduction

In this project, you will enhance the classic rock-paper-scissors game that we created last meeting by adding two additional options: "lizard" and "Spok". This version of the game, popularized by the TV show "The Big Bang Theory," offers more variety and complexity. You will also implement features to keep track of scores and play multiple rounds.



Steps to Create Your Game

1. *Create a new Python script in* https://replit.com/: Make sure to save your file with a descriptive name, such as "rock_paper_scissors.py"
2. *Define functions:* You will need to update the four different functions for this game, which will include:
    a. Get_user_choice(): This function will ask for the user to input "rock," "paper," or "scissors." You will need to add the options "lizard" and "Spok". It will return whatever the user chooses. Ensure the input is valid and ask the user to try again if it's not.
    b. Get_computer_choice(): This function will return a random choice of "rock," "paper," or "scissors." You will need to add the options "lizard" and "Spok".
    c. Determine_winner(): This function will use if-elif-else statements to compare the user's choice with the computer's choice and determine whether the user wins, the computer wins, or if it is a tie. It will return the result of the game.
    d. Play_game(): This function will call the other three functions in the correct order to run the game. This function will ask for user input at the end to see if the user would like to play again, and it will continue to run until the user says they do not want to continue playing (**hint hint: while loop).
3. *Keep track of scores:* Implement a scoring system to keep track of the user's and computer's scores throughout multiple rounds of the game. Update the scores after each round and display them to the user.
4. *Play multiple rounds:* Like the original version, ask the user if they would like to play again after each round. Alternatively, you could ask before the user plays how many rounds they would like to play.
5. *Test your game:* Once you've implemented all the components, test your game thoroughly to ensure it works as expected. Try a different combination of choices to verify the game's logic.
6. *If you finish early:* implement more creative features, such as:
    a. Allowing the user to choose the number of rounds to play
    b. Introduce special game modes such as "Best of 3," "Best of 5," etc.

c. Add a strategy for the computer to make its choices more dynamic and challenging (i.e. implement a simple algorithm that tries to predict the user's next move based on their previous choices, which will make the computer harder to beat)

Tips and Suggestions

- Break down the problem into smaller, manageable tasks. Focus on one function at a time.
- Use meaningful variable names and comments to make your code east to understand.
- Test your code frequently as you write it. Fix any errors or bugs as you find them.
- Feel free to experiment and add your own features to the game once the basic version is working! ☺

MG's code for previous version of Rock-Paper-Scissors (see below: rock-paper-scissors.py):

```python
import random

def get_user_choice():
    while True:
        user_choice = input("Enter your choice (rock, paper, or scissors): ").lower()
        if user_choice in ["rock", "paper", "scissors"]:
            return user_choice
        else:
            print("Invalid choice! Please enter rock, paper, or scissors.")

def get_computer_choice():
    return random.choice(["rock", "paper", "scissors"])

def determine_winner(user_choice, computer_choice):
    if user_choice == computer_choice:
        return "It's a tie!"
    elif (user_choice == "rock" and computer_choice == "scissors") or \
         (user_choice == "paper" and computer_choice == "rock") or \
         (user_choice == "scissors" and computer_choice == "paper"):
        return "You win!"
    else:
        return "Computer wins!"

def play_game():
    print("Let's play Rock-Paper-Scissors!")
    while True:
        user_choice = get_user_choice()
        computer_choice = get_computer_choice()
        print("You chose:", user_choice)
        print("Computer chose:", computer_choice)
        print(determine_winner(user_choice, computer_choice))
        play_again = input("Do you want to play again? (yes/no): ").lower()
        if play_again != "yes":
            print("Thanks for playing!")
            break

if __name__ == "__main__":
    play_game()
```