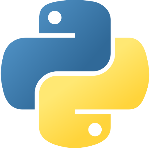
** ROCK PAPER SCISSORS**

**GAME IN PYTHON**

**Institude name:**

**Besent technology**

**Submitted to:**

**Gowthami**

**Coordinater Name:**

**Akilan**

**Submitted by:**

**S.Saranya**

**Acknowledgement**

I would like to express my sincere gratitude to Gowthami madam for their valuable guidance, support, and encouragement throughout the development of this project. Their insights and expertise have been instrumental in shaping this work.

Additionally, I am grateful to online communities and resources such as Stack Overflow, Python documentation, and open-source libraries that provided invaluable support in troubleshooting and enhancing the functionality of this project.

Finally, I appreciate the support of my family and friends, whose motivation and encouragement helped me complete this project successfully.

S. Saranya

23.03.2025

Table of contents

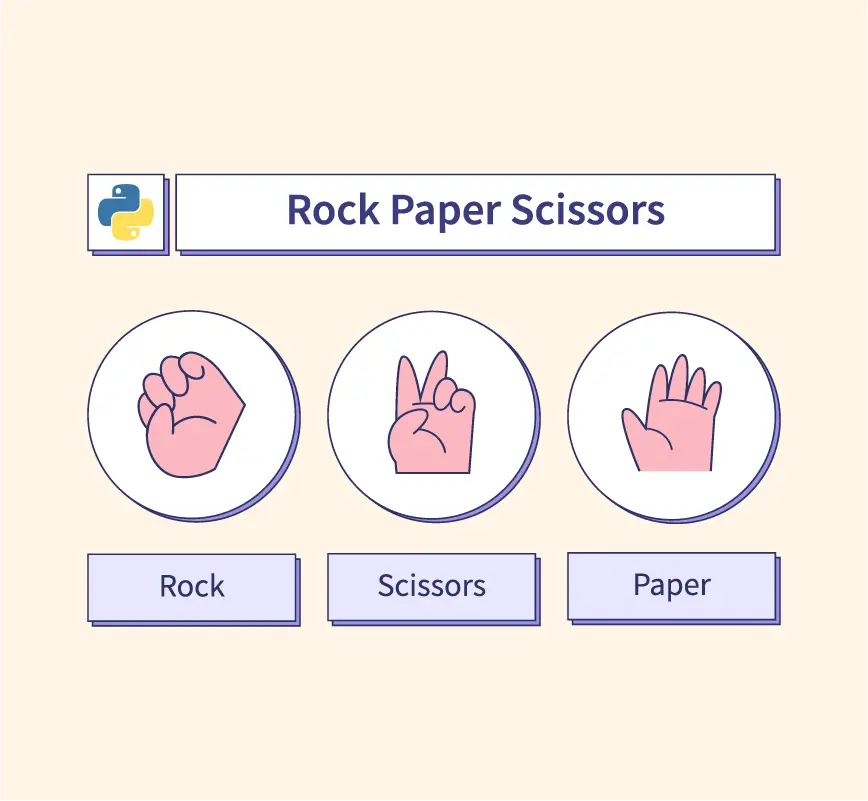
1. Introduction 4
2. Concept used in project 5
3. Source code 9
4. Description of Source code 10
5. Output 12
6. Conclusion 13

Introduction

* The Rock, Paper, Scissors game is a classic hand game played between two opponents, where each player randomly selects one of three options: rock, paper, or scissors. The game follows these simple rules:
* Rock beats Scissors (Rock crushes Scissors)
* Scissors beats Paper (Scissors cut Paper)
* Paper beats Rock (Paper covers Rock)
* This Python script provides a simple text-based implementation of the game, where the player competes against the computer.
* The computer makes a random choice using Python’s random module, and the script determines the winner based on predefined rules.

# Concepts used in project:

The game Rock, Paper, Scissors involves several fundamental programming concepts:



1.Conditional Statements (if-elif-else)

* Used to determine the winner by comparing the player’s choice with the computer’s choice.

Example:

if player ==computer:

print(“It’s a tie!”)

elif (player == “rock” and computer == “scissors”) or \

(player == “scissors” and computer == “paper”) or \

(player == “paper” and computer == “rock”):

print(“You win!”)

else:

print(“You lose!”)

1. Random Module:

* The computer’s choice is typically selected randomly using random.choice().

Example:

import random

choices = [“rock”, “paper”, “scissors”]

computer = random.choice(choices)

1. User Input Handling

* Taking input from the player and ensuring it is valid.

Example:

player = input(“Enter rock, paper, or scissors: “).lower()

while player not in choices:

player = input(“Invalid choice. Try again: “).lower()

1. Loops (Optional)

* To allow multiple rounds without restarting the program.

Example:

while True:

play\_again = input(“Play again? (yes/no): “).lower()

if play\_again != “yes”:

break

6.Functions (For Code Reusability)

* Encapsulating the game logic in functions for better organization.

Example:

def get\_winner(player, computer):

if player == computer:

return “Tie”

elif (player == “rock” and computer == “scissors”) or \

(player == “scissors” and computer == “paper”) or \

(player == “paper” and computer == “rock”):

return “You win!”

else:

return “You lose!”

Source code:

import random

def get\_computer\_choice():

choices = ["rock", "paper", "scissors"]

return random.choice(choices)

def get\_winner(player, computer):

if player == computer:

return "It's a tie!"

elif (player == "rock" and computer == "scissors") or \

(player == "scissors" and computer == "paper") or \

(player == "paper" and computer == "rock"):

return "You win!"

else:

return "You lose!"

def main():

while True:

player = input("Enter rock, paper, or scissors: ").lower()

if player not in ["rock", "paper", "scissors"]:

print("Invalid choice. Try again.")

continue

computer = get\_computer\_choice()

print(f"Computer chose: {computer}")

result = get\_winner(player, computer)

print(result)

play\_again = input("Play again? (yes/no): ").lower()

if play\_again != "yes":

print("Thanks for playing!")

break

if \_\_name\_\_ == "\_\_main\_\_":

main()

Description of source code:

* This Python script implements a simple **Rock, Paper, Scissors** game using the random module.

Code explanation:

1. **get\_computer\_choice()**
   * This function randomly selects **"rock"**, **"paper"**, or **"scissors"** using random.choice().
2. **get\_winner(player, computer)**
   * Compares the player’s choice against the computer’s.
   * Returns:
     + "It's a tie!" if both choices are the same.
     + "You win!" if the player beats the computer.
     + "You lose!" if the computer wins.
3. **main()**
   * Runs the game in a loop.
   * Asks the user to enter **"rock"**, **"paper"**, or **"scissors"**.
   * Validates user input; if invalid, prompts for a new input.
   * Calls get\_computer\_choice() to generate the computer’s move.
   * Calls get\_winner() to determine the outcome.
   * Displays the result and asks if the user wants to play again.
   * The game stops when the user enters "no".
4. **if \_\_name\_\_ == "\_\_main\_\_":**
   * Ensures the script runs only when executed directly.

Output:

Enter rock, paper, or scissors: rock

Computer chose: scissors

You win!

Play again? (yes/no): yes

Enter rock, paper, or scissors: paper

Computer chose: rock

You win!

Play again? (yes/no): yes

Enter rock, paper, or scissors: scissors

Computer chose: scissors

It's a tie!

Play again? (yes/no): no

Thanks for playing!

Conclusion:

* The Rock, Paper, Scissors game implemented in Python provides a simple yet engaging way to practice fundamental programming concepts such as user input, conditional statements, loops, and randomization.
* This script allows the player to compete against the computer, with results determined based on standard game rules. The game runs continuously until the user decides to exit, making it interactive and dynamic.