Python Project:

Make a two-player Rock-Paper-Scissors game. (Hint: Ask for player plays (using input), compare them, print out a message of congratulations to the winner, and ask if the players want to start a new game)

Remember the rules:

* Rock beats scissors
* Scissors beats paper
* Paper beats rock

**While loops**

We have already discussed for loops, or loops that look sequentially (one by one) at elements in a list. There is a second type of loop that works in a slightly different way called a while loop.

The idea is simple: while a certain condition is True, keep doing something. For example:

a **=** 5

**while** (a **>** 0):

**print**(a)

a **-=** 1

The output of this code segment is:

5

4

3

2

1

A particularly useful way to use while loops is checking user input for correctness. For example:

quit **=** input('Type "enter" to quit:' )

**while** quit **!=** "enter":

quit **=** input('Type "enter" to quit:' )

The uses for this are infinite and can (and should!) be combined with conditionals to yield the most efficient results.

**Infinite loops**

An infinite loop is a loop that never stops. This means that the condition in the beginning of the while loop will always be true.

For example:

i **=** 5

**while** i **>** 0:

**print**("Inside the loop")

What will happen is the loop will print out the phrase “Inside the loop” forever and ever. If you are running your computer, you will have to “kill the program” to stop it. Each operating system has a different way of “killing a program” to get out of an infinite loop.

On Linux: in the terminal, type “CTRL-C” to kill the program that is currently running in the terminal. If you are using the IDLE Python IDE, then you must press “CTRL-D” to exit your running program. When in doubt, do a Google search before you start programming!

On Windows: type “CTRL-ALT-DEL” and open the task manager to kill the program.

On Mac: right-click on the task, and kill the program that is running forever.

If you find yourself in an infinite loop, your program will never end.

**Break statements**

A break statement stops the execution of a loop before the original condition is met. While the use of a breakstatement will often start an argument about good coding practices, sometimes it is useful.

For example:

**while** True:

usr\_command **=** input("Enter your command: ")

**if** usr\_command **==** "quit":

**break**

**else**:

**print**("You typed " **+** usr\_command)

In this case, the break statement is used to break off the “infinite while loop” that we have constructed with the while True statement.

2 players – computer and player

**Conditions:**

* Rock beats scissors
* Scissors beats paper
* Paper beats rock

|  |  |  |
| --- | --- | --- |
| **Player1** | **Player2** | **Outcome** |
| Rock | Rock | Tie |
| Rock | Paper | Player2 Wins |
| Rock | Scissors | Player1 wins |
| Paper | Rock | Player1 wins |
| Paper | Paper | Tie |
| Paper | Scissors | Player2 wins |
| Scissors | Rock | Player2 wins |
| Scissors | Paper | Player1 wins |
| Scissors | Scissors | Tie |
|  | **Quit** | exit |

Text

Description automatically generated with low confidence

If yes then “please type your choice: rock, paper, or scissors”

If Q then exit

Loop

Player2 makes first move

Quit exits the game

Random selection

If Player2 = “rock” and Player1 = “rock” then print(“Tie”)

If Player2 = “rock” and Player1 = “paper” then print(“Player1 wins”)

If Player2 = “rock” and Player1 = “scissors” then print(“Player2 wins”)

If Player2 = “paper” and Player1 = “rock” then print(“Player2 wins”)

If Player2 = “paper” and Player1 = “paper” then print(“Tie”)

If Player2 = “paper” and Player1 = “scissors” then print(“Player1 wins”)

If Player2 = “scissors” and Player1 = “rock” then print(“Player1 wins”)

If Player2 = “scissors” and Player1 = “paper” then print(“Player2 wins”)

If Player2 = “scissors” and Player1 = “scissors” then print(“Tie”)

If Player2 = “quit” then Exit

# RPC.py

def greeting():

"""This function displays a greeting"""

print("Hi there! Would you like to play a game of Rock Paper Scissors?")

greeting()

import random

user\_choice = input("Please make a choice (rock, paper, scissors):")

game\_options = ["rock", "paper", "scissors"]

computer\_choice = random.choice(game\_options)

print(f"\nYou chose {user\_choices}, the computer chose {computer\_choices}. \n)

if user\_action == computer\_action:

print(f"Both players selected {user\_action}. It's a tie!")

elif user\_action == "rock":

if computer\_action == "scissors":

print("Rock smashes scissors! You win!")

else:

print("Paper covers rock! You lose.")

elif user\_action == "paper":

if computer\_action == "rock":

print("Paper covers rock! You win!")

else:

print("Scissors cuts paper! You lose.")

elif user\_action == "scissors":

if computer\_action == "paper":

print("Scissors cuts paper! You win!")

else:

print("Rock smashes scissors! You lose.")

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

import random

game\_options = ["rock", "paper", "scissors"]

print("Make a choice")

user\_choice = input("Please make a choice (rock, paper, scissors):")

if user\_choice in choices:

computer\_choice = random.choice(game\_options)

print(f"\nYou chose'{user\_choice}',the computer chose'{computer\_choice}'")

else:

print(f"\nYou selected '{user\_choice}' invalid choice")

if user\_choice == computer\_choice:

print(f"We both chose {user\_choice}. Tie!")

elif user\_choice == "rock":

if computer\_choice == "scissors"

print("Rock beats scissors. You win!")

else:

print("Paper beats rock. You lose.")

elif user\_choice == "paper":

if computer\_choice == "rock":

print("Paper beats rock. You win!")

else:

print("Scissors beat paper. You lose.")

elif user\_choice == "scissors":

if computer\_choice == "paper":

print("Scissors beat paper. You win!")

else:

print("Rock beats paper. You lose.")

import random

while True:

user\_choice = input("Please make a choice (rock, paper, scissors): ")

game\_options = ["rock", "paper", "scissors"]

computer\_choice = random.choice(game\_options)

print(f"\nYou chose {user\_choice}, computer chose {computer\_choice}. \n")

if user\_choice == computer\_choice:

print(f"We both chose {user\_choice}. Tie!")

elif user\_choice == "rock":

if computer\_choice == "scissors":

print("Rock beats scissors. You win!")

else:

print("Paper beats rock. You lose.")

elif user\_choice == "paper":

if computer\_choice == "rock":

print("Paper beats rock. You win!")

else:

print("Scissors beat paper. You lose.")

elif user\_choice == "scissors":

if computer\_choice == "paper":

print("Scissors beat paper. You win!")

else:

print("Rock beats paper. You lose.")

again = input("Do you want to play again? Select 'y' if yes, 'n' if no")

if again.lower() != "y":

break

Graphical user interface, text, application

Description automatically generated