💥

Rock, Paper, Scissors Game [Completed]

🎮 Project: Rock, Paper, Scissors Game in Python

✅ Concepts Used:

- Functions (Chapter 8)
- Conditionals (Chapter 4)
- Loops (Chapter 7)
- User Input (Chapter 1)
- Strings & Logic (Chapter 3 & 5)
- Random Module (new concept)
- Clean code structure

🧠 Idea:

User plays Rock, Paper, Scissors against the computer. Game repeats until user quits or a score target is hit. Tracks score, validates input, and gives feedback.

📦 Step-by-step code with comments

```
# 📌 Import random module to let computer randomly choose an option
import random

# 🎯 Define valid choices in a list
choices = ["rock", "paper", "scissors"]

# 🧠 Score trackers
user_score = 0
comp_score = 0

# 📌 Function to decide the winner of a round
def decide_winner(user, comp):
    """
    Takes user and computer choice, returns outcome string:
    'win', 'lose', or 'draw'
    """
    if user == comp:
        return "draw"

    # Define winning conditions
    if (user == "rock" and comp == "scissors") or \
       (user == "paper" and comp == "rock") or \
       (user == "scissors" and comp == "paper"):
        return "win"

    return "lose"

# 🚀 Main game loop
while True:
    print("\n--- Rock, Paper, Scissors Game ---")
```

```
print("Enter your choice (rock/paper/scissors) or 'q' to quit:")
user_choice = input("👉 Your move: ").lower().strip()

# 🛑 Check for quit
if user_choice == "q":
    print("👋 Thanks for playing!")
    break

# ❌ Invalid input handling
if user_choice not in choices:
    print("⚠️ Invalid input. Please enter rock, paper or scissors.")
    continue

# 💻 Computer makes a random choice
comp_choice = random.choice(choices)
print(f"🤖 Computer chose: {comp_choice}")

# 🔍 Decide winner and update scores
result = decide_winner(user_choice, comp_choice)

if result == "win":
    print("✅ You win this round!")
    user_score += 1
elif result == "lose":
    print("❌ You lost this round.")
    comp_score += 1
else:
    print("🤝 It's a draw!")

# 🧾 Display current score
print(f"📊 Score → You: {user_score} | Computer: {comp_score}")
```


🛠️ Suggestions for Improvement (Try These if You Have Time):

Feature	Concept Used	Level
Track history of moves	Lists	Easy
Add round number	Loops & Counters	Easy
Play up to N rounds	Loop + Break Conditions	Medium
Create a score file (`score.txt`)	File Handling	Medium
Add emojis using Unicode	Fun UX	Easy
GUI version (Tkinter)	Python Module	Medium
Function to replay game	Function Composition	Medium
Use `main()` structure	Good Coding Practice	Medium
Use `*args` or `**kwargs` to handle optional input	Chapter 8	Medium

🔁 Example Output:

```
--- Rock, Paper, Scissors Game ---
Enter your choice (rock/paper/scissors) or 'q' to quit:
👉 Your move: rock
🤖 Computer chose: scissors
✅ You win this round!
📊 Score → You: 1 | Computer: 0
```

📚 Summary of Learning (from this project)

Skill	How it was used
`functions`	To handle logic like winner check
`loops`	Repeating game rounds
`input()` & `print()`	UI interaction
`random` module	Computer decision making
`if-else`	Game logic control
Clean code	Modular, readable, commented structure

🧠 Bonus Tip:

To keep your projects clean:

- Use `main()` to start programs
- Group logic inside functions
- Keep global variables minimal
- Add docstrings and comments