## Day 2: Make It Replayable, Add Quit Option & Track Score

**Make sure to test your game thoroughly before submitting the link :)**

### 🎯 Objective:

**You will improve your Day 1 Rock, Paper, Scissors game by:**

- **Letting the user play multiple rounds**
- **Adding a "quit" option**
- **Keeping track of the player and computer scores**

### Step 1: Open Your Day 1 Code

Find your Day 1 Python file (where the game plays one round).
We will now modify that code piece by piece ✨.

### Step 2: Update the program description

Since we are adding new features, we are going to update the program description.

```
# Your Name
# 7th Grade Computer Science
# Today's Date
# Rock, Paper, Scissors Game - Day 2
# This program lets the player compete against the computer in a game of Rock, Paper, Scissors.
# The player can play unlimited rounds, choose to quit at any time,
# and the program keeps track of scores.
```

### Step 3: Add Score Variables (Above the Player Input Line)

Right after your welcome message, **add two new variables** to keep score.

👉 **Before:**

```
print("Welcome to Rock, Paper, Scissors!")
```

👉 **Change to:**

```
print("Welcome to Rock, Paper, Scissors!")
print("Type 'quit' at any time to stop playing.")

player_score = 0
computer_score = 0
```

### Step 4: Wrap Everything in a `while True` Loop

Put your entire Day 1 code inside a `while True:` loop so the game can **repeat automatically**.

👉 At the top of your code, add this:

```
while True:
```

👉 Then indent all your existing game logic underneath it.

### Step 5: Replace the Player Input Section

We want to:

- Keep asking until the input is **valid**
- Allow the user to **type "quit" to exit**

👉 Replace your old player input code with:

```
player_choice = input("Enter rock, paper, or scissors (or 'quit' to exit): ").lower()

# Quit the game
if player_choice == "quit":
    break

# Check if input is valid
if player_choice not in choices:
    print("Invalid choice! Please try again.")
    continue
```

This code makes sure:

- Only `"rock"`, `"paper"`, or `"scissors"` continue the game
- Any other input shows an error and **restarts the loop**
- `"quit"` **ends** the game loop

### Step 6: Keep Your Computer Choice and Display Code

This part stays the same:

```
computer_index = random.randint(0, 2)
computer_choice = choices[computer_index]

print("You chose:", player_choice)
print("Computer chose:", computer_choice)
```

### Step 7: Add Score Tracking to Your Winner Code

Update your winner-checking code so that it **adds 1 point** to the winner's score.
👉 Replace this section:

```
if player_choice == computer_choice:
    print("It's a tie!")
elif (player_choice == "rock" and computer_choice == "scissors") or \
     (player_choice == "paper" and computer_choice == "rock") or \
     (player_choice == "scissors" and computer_choice == "paper"):
    print("You win!")
else:
    print("Computer wins!")
```

👉 With this:

```
if player_choice == computer_choice:
    print("It's a tie!")
elif (player_choice == "rock" and computer_choice == "scissors") or \
     (player_choice == "paper" and computer_choice == "rock") or \
     (player_choice == "scissors" and computer_choice == "paper"):
    print("You win!")
    player_score = player_score + 1
else:
    print("Computer wins!")
    computer_score = computer_score + 1
```

### Step 8: Show the Current Score

After determining the winner, show the updated scores.

👉 Add this line **at the end of the loop**:

```
print("Score - You:", player_score, "| Computer:", computer_score)
```

### Step 9: After the Loop, Print Final Score

After the loop ends (when the player types `"quit"`), show the **final score** and say goodbye.

👉 At the very bottom of your file (no indentation), add:

```
print("Final Score:")
print("You:", player_score)
print("Computer:", computer_score)
print("Thanks for playing!")
```

### Step 10: 🧪 Test Your Game

- Typing `"rock"`, `"paper"`, or `"scissors"` → ✅
- Typing `"quit"` → ✅ ends the game and shows final score
- Typing `"hello"` or `"banana"` → ❌ error message, asks again

- Multiple rounds → score updates each time

Your complete program should look like this:

```
# Ms. Liu
# 7th Grade Computer Science
# 3/25/2025
# Rock, Paper, Scissors Game - Day 2
# This program lets the player compete against the computer in a game of Rock, Paper, Scissors.
# The player can play unlimited rounds, choose to quit at any time, and the program keeps track of scores.

import random

# Define valid choices
choices = ["rock", "paper", "scissors"]

print("Welcome to Rock, Paper, Scissors!")
print("Type 'quit' at any time to stop playing.")

# Initialize scores
player_score = 0
computer_score = 0

# Start the game loop
while True:
    player_choice = input("Enter rock, paper, or scissors (or 'quit' to exit): ").lower()

    # Quit the game
    if player_choice == "quit":
        break

    # Check if input is valid
    if player_choice not in choices:
        print("Invalid choice! Please try again.")
        continue

    # Computer makes a random choice
    computer_index = random.randint(0, 2)
    computer_choice = choices[computer_index]
```

```
        # Display both choices
        print("You chose:", player_choice)
        print("Computer chose:", computer_choice)

        # Determine the winner
        if player_choice == computer_choice:
            print("It's a tie!")
        elif (player_choice == "rock" and computer_choice == "scissors") or \
             (player_choice == "paper" and computer_choice == "rock") or \
             (player_choice == "scissors" and computer_choice == "paper"):
            print("You win!")
            player_score = player_score + 1
        else:
            print("Computer wins!")
            computer_score = computer_score + 1

        # Display current score
        print("Score - You:", player_score, "| Computer:", computer_score)

    # Print final score after quitting
    print("Final Score:")
    print("You:", player_score)
    print("Computer:", computer_score)
    print("Thanks for playing!")
```

Reflection Questions:

Answer the following questions in complete sentences.

- Why did we use .lower() when getting the player's input?
- What happens if the user enters something other than "rock", "paper", or "scissors"?
- What would happen if you forgot to initialize player_score = 0 and computer_score = 0 at the beginning?
- What new Python concept did you understand better after doing this project?
- What extra feature would you want to add to your game?