Project Title:

Rock-Paper-Scissors GUI Game

Author:

Ritam Mondal

B.Tech Student, IIT Kharagpur

My GitHub: @ritammondal2004

Find this detailed project on **Github**

1. Project Overview

This project is a visually engaging and interactive Rock-Paper-Scissors game built using Python. It features a GUI interface using Tkinter and realistic sound effects via **Pygame**. The game allows players to compete against a bot for a selected number of rounds with real-time feedback and a creative end-screen popup.

2. Problem Statement

- Choose a number of rounds to play (1 to 5)
- Select between Rock, Paper, or Scissor against a bot
- Display results with expressive images and sound effects
- Show win

Design and develop an interactive Rock-Paper-Scissors game using Python with a graphical user interface (GUI).

3.Objective

- Demonstrate use of GUI libraries (Tkinter)
- Implement game logic and score tracking
- Add user experience features like sound (pygame) and animations (PIL)
- Maintain clean project structure using folders like images/, sounds/

4.Tech Stack Used

Tool/Library	Purpose
Python 3.x	Core programming
Tkinter	GUI framework
Pillow	Image handling (resizing, displaying)
Pygame	Sound effects
Git & GitHub	Version control and deployment
VS Code	Development environment

5. Features

I. GUI interface with player/bot hands and emotions

II. Round selector with customizable gameplay

III. Sound effects based on actions (rock, paper, scissor, result, exit)

IV. Creative visuals with emojis, icons, background image

V. End-game popup with result display and action options

VI. Bot randomly selects moves for fair gameplay

6. How It Works

a. Starting Page

- ➢ Displays a welcome background and buttons to choose 1–10 rounds.
- ➢ Plays a looping welcome sound using Pygame.

b. Gameplay Interface

- ➢ Each click triggers:
  - ✓ Image changes (player and bot hand gestures)
  - ✓ Sound effects
  - ✓ Score update
  - ✓ Round progress display

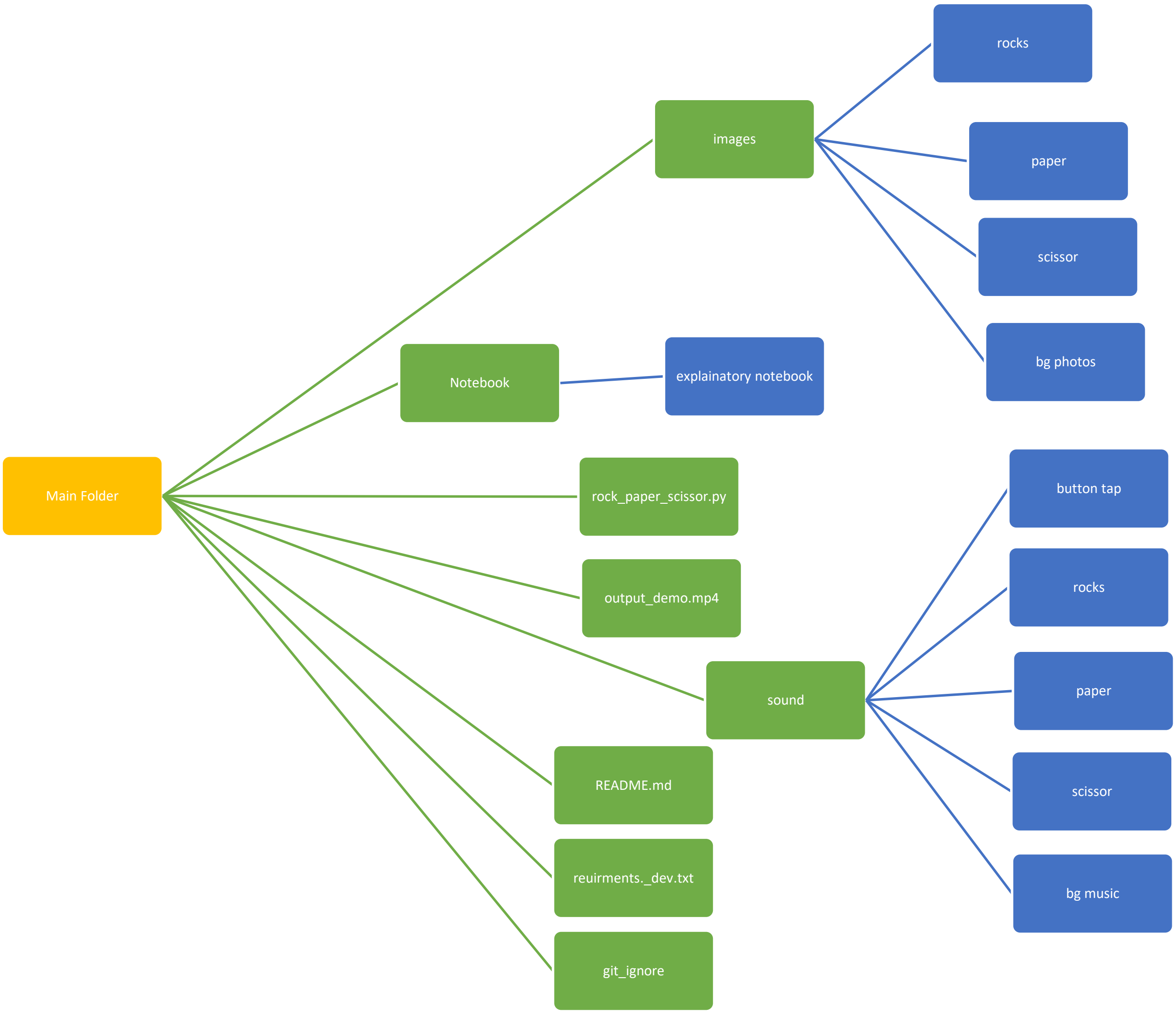
c. Result Handling

- ➢ Final score is evaluated after all rounds.
- ➢ Shows result image (win/lose/draw) with sound.
- ➢ Popup allows player to restart or exit.

7. Challenges Faced

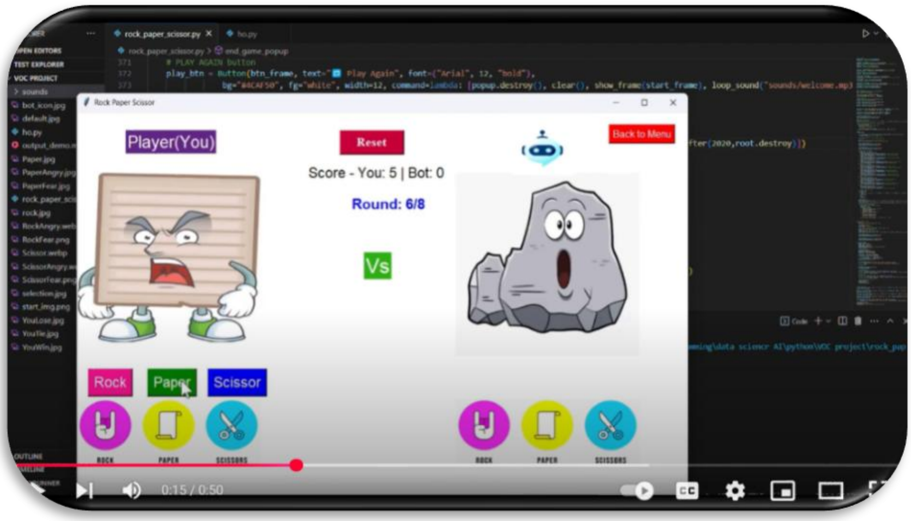
- Handling sound play concurrency
- Centering popup and dynamic image resizing
- GUI layout management across frames

8. Folder Structure on Github :



9. Demo Video

Embedded in README (or available via **YouTube**):



```
# PLAY AGAIN button
play_btn = Button(btn_frame, text="🔄 Play Again", font=("Arial", 12, "bold"),
            bg="#4CAF50", fg="white", width=12, command=lambda: [popup.destroy(), clear(), show_frame(start_frame), loop_sound("sounds/welcome.mp3
```

```
fter(2020,root.destroy)])
```

Score - You: 5 | Bot: 0

Round: 6/8

10. Future Improvements

- Add player vs player mode
- Track and display match history
- Store high scores using a file or database
- Add animations or use GIFs for more realism
- Deploy on the website using Fast API or Flask

Conclusion

This project is a fun application of Python GUI development. It helped explore real-world skills like event handling, sound/image integration, and GitHub-based collaboration and deployment. It’s not just a game — it’s a polished, user-friendly software product.