**🪨📄✂️ Rock Paper Scissors AI Battle Game**

**Project Builder:** P. Vishwateja

**🎮 Overview**

An engaging and interactive Rock-Paper-Scissors game developed in Python using the Pygame library. The game allows users to challenge different AI opponents, each employing a unique strategy. Designed with a user-friendly interface featuring emojis, it works seamlessly even in simple text editors like Notepad.

**✨ Features**

* 🎯 **Opponent Selection Menu**: Choose from four distinct AI strategies:
  + 🤖 **Random Bot** – Picks a move at random
  + 🪨 **Rock Repeater Bot** – Always plays rock
  + 🔄 **Cycle Bot** – Cycles through rock, paper, and scissors
  + 🧠 **Copycat Bot** – Mimics your last move
* 🖼️ **Emoji-based Interface**: Uses emoji icons for rock (✊), paper (🖐️), and scissors (✌️)
* 🔙 **Back to Menu** and ❌ **Exit Buttons** for better navigation
* 🖥️ **Simple GUI** designed with Pygame for ease-of-use and compatibility

**⚙️ Requirements**

* Python 3.x
* Pygame library

**🛠️ Setup Instructions**

1. Download and install Python from [python.org](https://www.python.org/downloads/)
2. Open your command prompt and install Pygame:
3. pip install pygame
4. Save your Python script as rps\_game.py using Notepad or your preferred editor.
5. Run the game using the command:
6. python rps\_game.py

**▶️ How to Play**

* Launch the game and select an AI opponent from the menu.
* Use your mouse to click on one of the emoji buttons:
  + ✊ Rock
  + 🖐️ Paper
  + ✌️ Scissors
* View results after each round.
* Choose to play again, return to the menu, or exit the game.

**📁 Project Structure**

/rock\_paper\_scissors\_project

|-- rps\_game.py

**🚀 Future Enhancements**

* 📊 Scoreboard to track wins/losses
* 🔊 Sound effects for better engagement
* 🧑‍🤝‍🧑 Two-player mode
* 🧠 Advanced AI that learns and adapts over time

**👨‍💻 Author**

Built with passion by **P. Vishwateja**.

Feel free to fork, star, and contribute on GitHub!

**📌 Notes**

* The use of emojis ensures smooth operation even without external image files.
* Built and tested in a simple environment using Notepad and command-line Python.

Happy Coding! 🚀

**⚙️ Step-by-Step Development Process**

**1. Define Game Logic and Rules**

* The basic rules of Rock-Paper-Scissors were implemented using Python.
* The beats() function was created to determine the winner of each round.

**2. Design Multiple AI Bots**

Each bot has its own logic:

* random\_bot: Chooses a move randomly.
* repeat\_rock\_bot: Always plays rock.
* cycle\_bot: Cycles through R → P → S.
* copy\_last\_move\_bot: Mimics the opponent's last move.

**3. Develop the Adaptive Player**

* This player analyzes the opponent's move history and adapts using 3 strategies:
  + **Frequency analysis**
  + **Pattern detection**
  + **Cycle prediction**
* The best strategy is chosen based on its recent success rate.

**4. Create Game Simulator**

* A test loop was written to simulate multiple rounds.
* Statistics like wins, losses, and draws are calculated to validate performance.

**5. Integrate GUI with Pygame**

* Used pygame to create a minimal GUI.
* Replaced image assets with **emoji-based icons** (✊, 🖐️, ✌️) to ensure it runs smoothly in simple setups like Notepad without needing extra files.

**6. Add Interactivity**

* Menu screen to choose the opponent bot.
* Game screen with rock, paper, scissors buttons.
* Result display after each round.
* “Back to Menu” and “Exit” buttons for better navigation.

**7. Refine UX/UI**

* Emojis added for game actions.
* Font sizes adjusted for readability.
* Buttons styled with rounded corners and hover effects.

**8. Test the Game**

* Each AI bot was tested over multiple rounds to ensure fairness and stability.
* UI interactions, buttons, and game state transitions were checked manually.

**🧪 Expected Errors & Fixes**

| **Issue** | **Fix** |
| --- | --- |
| pygame.error: video system not initialized | Ensure pygame.init() is called before any drawing or display function. |
| Font/emoji not displaying correctly | Make sure your terminal or editor supports Unicode fonts. |
| Crash on first move | Handled empty history lists for bots like copy\_last\_move\_bot. |
| Button click not working | Added accurate collision detection using collidepoint(). |