Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

Mahatma Education Society's

**PILLAI COLLEGE OF ARTS,COMMERCE & SCIENCE**

**(Autonomous)**

**Re-accredited “A” Grade by NAAC (3rd Cycle)**

PILLAI COLLEGE OF ARTS, COMMERCE & SCIENCE
NEW PANVEL
M. E. S.
pcacs

# Project Completion Certificate

**THIS IS TO CERTIFY THAT**

**SHREYA BHATTACHARJEE**

of **M.Sc. Data Analytics Part - II** has completed the project titled **AI-Powered Rock Paper Scissors: Smart Play** of subject **Artificial Intelligence** under our guidance and supervision during the academic year 2024-25 in the department of Master of Data Analytics.

Project Guide

Course Coordinator

Pillai College of Arts, Commerce & Science (Autonomous)
New Panvel

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

# MAHATMA EDUCATION SOCIETY’S

PILLAI COLLEGE OF ARTS, COMMERCE & ARTS SCIENCE

(Autonomous)
NEW PANVEL

PROJECT REPORT ON

“**AI-Powered Rock Paper Scissors: Smart Play**”

IN PARTIAL FULFILLMENT OF

MASTER OF SCIENCE

SEMESTER IV – 2024-25

PROJECT GUIDE
Name: Miss.Sanjana Bhangale

SUBMITTED BY: Shreya Bhattacharjee

ROLL NO: 6861

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

## Introduction

The **Rock Paper Scissors** game is a classic hand game that has been digitized using **Python and Tkinter** to provide an interactive and visually appealing experience. This project features a **Graphical User Interface (GUI)** with engaging visuals and multiple game modes, including **Human vs Computer** and **Human vs Human** options. Additionally, an AI-powered difficulty system allows players to challenge themselves against different levels of AI intelligence.

The game incorporates various **user-friendly enhancements** such as a **dark/light theme toggle, score tracking, game history, and animations**. It leverages **Python's Tkinter library** to create an intuitive interface with real-time feedback, making the game both entertaining and educational. The AI-driven difficulty levels ensure that players can enjoy a dynamic challenge, while the history tracking adds a competitive edge. This project serves as both a fun gaming experience and a practical implementation of **AI-based decision-making and UI design**.

## Objective

The primary objective of this project is to develop an interactive and AI-driven **Rock Paper Scissors** game with an intuitive graphical interface. This implementation aims to:

1. **Provide Multiple Game Modes:** Allow players to compete against either a **computer AI** or another **human player**.
2. **Enhance AI Capabilities:** Introduce difficulty levels (**Easy, Medium, Hard**) to create a more challenging experience.

3. **Improve User Experience:** Implement a visually appealing interface with smooth animations, real-time score updates, and an interactive UI.
4. **Track Scores and History:** Keep track of wins, losses, and draws, displaying a history of the last **five rounds**.
5. **Customize Appearance:** Introduce a **Dark/Light Mode** toggle to enhance user comfort.
6. **Ensure Fair Gameplay:** Implement logic that ensures fair and unbiased competition between the player and the AI.
7. **Increase Accessibility:** Design the game to be intuitive for users of all age groups and experience levels.
8. **Promote Strategic Thinking:** The AI adapts its difficulty based on previous rounds, encouraging players to refine their strategy.

## Tools and Techniques Used

### Programming Language:

- **Python** – The core language used for implementing the game's logic and UI.

### GUI Development:

- **Tkinter** – Used for creating an interactive graphical interface with buttons, labels, and dynamic updates.
- **PIL (Pillow)** – Handles image processing to ensure smooth visual representation.

### Game Logic Implementation:

- **Random Module** – Generates AI's moves randomly in the **Easy mode** to ensure unbiased gameplay.
- **Conditional Statements** – Determines the winner based on the user and AI choices.
- **Pattern-Based AI (Medium & Hard Mode)** – AI makes decisions based on previous player moves, introducing strategy.
- **Event Handling** – Implements button clicks and user interactions seamlessly.

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

## User Experience Enhancements:

- **Animations & Hover Effects** – Enhances interactivity and engagement.
- **Dark/Light Mode Toggle** – Allows users to customize the UI theme.
- **Score Tracking System** – Keeps track of player and computer wins dynamically.
- **Game History Display** – Shows the last five results, allowing users to analyze performance.
- **Color-Coded Feedback Messages** – Provides instant visual feedback for win/loss/draw outcomes.
- **Dynamic UI Updates** – Instantly updates graphical elements and scores based on player actions.

## Additional Features:

- **Human vs Human Mode** – Enables two players to compete against each other.
- **Difficulty Level Selection** – Allows switching between **Easy, Medium, and Hard** AI difficulties.
- **Sound Effects & Alerts** – Enhances the gaming experience with audio cues.
- **Customizable Themes** – Future enhancements could allow users to personalize themes beyond dark/light mode.
- **Cross-Platform Compatibility** – Runs on different operating systems without major modifications.

# Code

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

# Basic Console-Based Rock Paper Scissors (Play 10 Rounds)

This version of the game allows the player to compete against the computer for **10 rounds**. It tracks the **number of wins, losses, and draws** and provides a summary at the end.

```
#no. of play 10 times
#result : who win how many , draw how many
#q to quit

import random
userWins = 0
compWins = 0
draws=0
chance=0
options = ["rock" , "paper", "scissors"]
while chance<10:
    chance += 1
    userinput = input("Enter your chooice \n Type
Rock/Paper/Scissors or Q to quit ").lower()
    if userinput == "q":
        break

    if userinput not in options:
        continue

    compchoice = random.choice(options)
    print(" Computer chooses" , compchoice)

    if userinput == compchoice:
        print("It's a DRAW!!")
        draws += 1

    elif userinput == "rock" and compchoice=="scissors":
        print("You Win!")
        userWins += 1

    elif userinput == "paper" and compchoice=="rock":
        print("You Win!!")
        userWins += 1

    elif userinput == "scisssors" and
compchoice=="paper":
        print("You Win!!")
        userWins += 1

    else:
        print("You loose!")
        compWins += 1

print("You wins ", userWins, "times")
print("Computer wins ", compWins,"times")
print("Draws = ", draws)
print("Thankyou for playing")
```

## Simple Console-Based Rock Paper Scissors

This is a **simplified text-based version** where the player selects an option, and the computer randomly chooses its move. The result is displayed immediately.

```
from random import choices, randint
choices = ["rock" , "paper" , "scissor"]
computer = choices[randint(0,2)]

print("Welcome to the Rock , Paper , Scissors Game!\n")
player = input("Your choice: ").lower()
print("computer chooses: ", computer)

if player == computer:
    print("DRAW!")
elif player == "rock" and computer == "paper":
    print("Computer Wins!")
elif player == "rock" and computer == "scissors":
    print("Player Wins!")
elif player == "paper" and computer == "scissors":
    print("Computer Wins!")
elif player == "paper" and computer == "rock":
    print("Player Wins!")
elif player == "scissors" and computer == "rock":
    print("Computer Wins!")
elif player == "scissors" and computer == "paper":
    print("player Wins!")
```

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

# Graphical Rock Paper Scissors Game Using Tkinter

This implementation enhances the **Rock Paper Scissors** game by integrating a **Graphical User Interface (GUI)** using **Tkinter**. It features **animations, score tracking, AI difficulty levels, and multiple game modes** (Human vs Computer and Human vs Human).

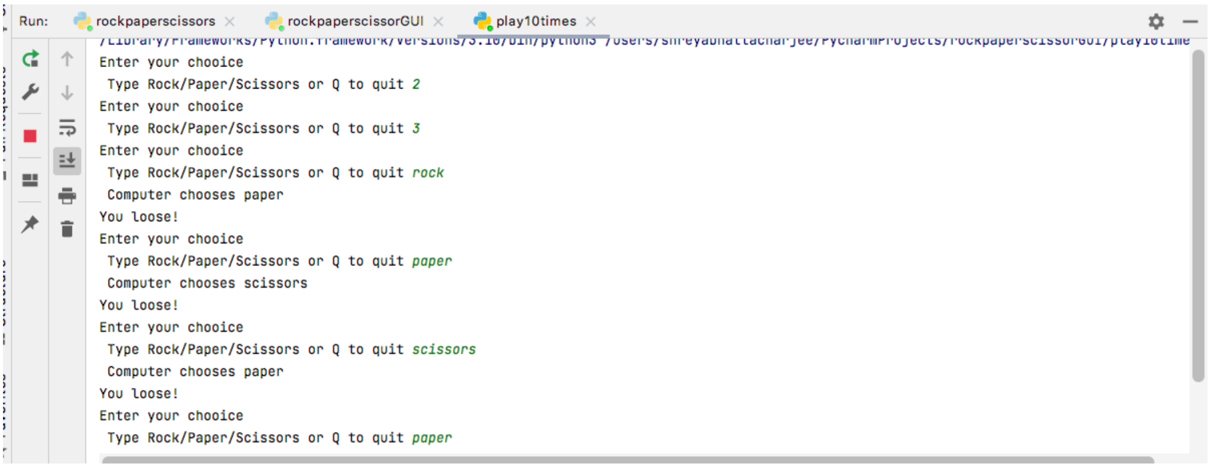
```
from tkinter import *
from PIL import Image, ImageTk
from random import choice, randint
import os

# Create main window
root = Tk()
root.title("Rock Paper Scissors Game")
root.configure(background="#222831")
root.geometry("800x600")


# Load images safely
def load_image(path):
    if os.path.exists(path):
        return ImageTk.PhotoImage(Image.open(path))
    return None


rock_img = load_image("rock-user.png")
paper_img = load_image("paper-user.png")
scissor_img = load_image("scissors-user.png")
rock_img_comp = load_image("rock.png")
paper_img_comp = load_image("paper.png")
scissor_img_comp = load_image("scissors.png")

# Game variables
mode = "Human vs Computer"
difficulty = "Easy"
history = []
is_dark_theme = True
player1_choice = None

# UI Components
comp_label = Label(root, image=scissor_img_comp,
bg="#222831")
user_label = Label(root, image=scissor_img, bg="#222831")
```

```
comp_label.grid(row=1, column=0, padx=20, pady=20)
user_label.grid(row=1, column=4, padx=20, pady=20)

playerScore = Label(root, text="0", font=("Arial", 20,
"bold"), bg="#222831", fg="white")
computerScore = Label(root, text="0", font=("Arial", 20,
"bold"), bg="#222831", fg="white")
computerScore.grid(row=1, column=1)
playerScore.grid(row=1, column=3)

msg = Label(root, text="", font=("Arial", 14, "bold"),
bg="#222831", fg="red")
msg.grid(row=4, column=2)

player_choice_label = Label(root, text="Your Choice: ",
font=("Arial", 12), bg="#222831", fg="white")
computer_choice_label = Label(root, text="Computer's
Choice: ", font=("Arial", 12), bg="#222831", fg="white")
player_choice_label.grid(row=2, column=3)
computer_choice_label.grid(row=2, column=1)

history_label = Label(root, text="History: ",
font=("Arial", 12), bg="#222831", fg="white")
history_label.grid(row=7, column=2, pady=10)


def updateChoice(user_choice):
    global player1_choice

    if mode == "Human vs Human":
        if player1_choice is None:
            player1_choice = user_choice
            msg.config(text="Player 2's Turn",
fg="yellow")
        else:
            comp_choice = user_choice
            decide_winner(player1_choice, comp_choice)
            player1_choice = None
    else:
        comp_choice = computerMove()
        decide_winner(user_choice, comp_choice)


def computerMove():
    if difficulty == "Easy":
        return choice(["rock", "paper", "scissors"])
    elif difficulty == "Medium":
        return choice(["rock", "paper", "scissors",
"scissors"])
    else:
        if history.count("Win") > history.count("Lose"):
            return choice(["rock", "paper"])
        return choice(["rock", "paper", "scissors"])


def decide_winner(player, computer):
    if player is None or computer is None:
        return

    user_label.config(image={"rock": rock_img, "paper":
paper_img, "scissors": scissor_img}[player])
    comp_label.config(image={"rock": rock_img_comp,
"paper": paper_img_comp, "scissors":
scissor_img_comp}[computer])
    player_choice_label.config(text=f"Your Choice:
{player.upper()}", fg="pink")
    computer_choice_label.config(text=f"Computer's
Choice: {computer.upper()}", fg="pink")

    if player == computer:
        msg.config(text="It's a DRAW!!!", fg="red")
        history.append("Draw")
    elif (player == "rock" and computer == "scissors") or
\
            (player == "paper" and computer == "rock") or
\
            (player == "scissors" and computer ==
"paper"):
        msg.config(text="You Win! 🎉", fg="green")
        updateScore(playerScore)
        history.append("Win")
    else:
        msg.config(text="You Lose! 😢", fg="red")
        updateScore(computerScore)
        history.append("Lose")
    updateHistory()


def updateScore(label):
    label.config(text=str(int(label["text"]) + 1))


def updateHistory():
    history_label.config(text="History: " + ",
".join(history[-5:]))


def change_mode():
    global mode
    mode = "Human vs Human" if mode == "Human vs
Computer" else "Human vs Computer"
    mode_btn.config(text=mode)


def change_difficulty():
    global difficulty
    difficulty = {"Easy": "Medium", "Medium": "Hard",
"Hard": "Easy"}[difficulty]
    difficulty_btn.config(text=f"Difficulty:
{difficulty}")


def toggle_theme():
    global is_dark_theme
    if is_dark_theme:
        root.configure(background="#f0f0f0")
    else:
        root.configure(background="#222831")
    is_dark_theme = not is_dark_theme


theme_btn = Button(root, text="Toggle Theme",
command=toggle_theme, bg="gray", font=("Arial", 12,
"bold"))
theme_btn.grid(row=6, column=2, pady=10)

mode_btn = Button(root, text=mode, command=change_mode,
bg="yellow", font=("Arial", 12, "bold"))
mode_btn.grid(row=5, column=1, pady=10)

difficulty_btn = Button(root, text=f"Difficulty:
{difficulty}", command=change_difficulty, bg="orange",
                        font=("Arial", 12, "bold"))
difficulty_btn.grid(row=5, column=3, pady=10)

Button(root, text="ROCK", command=lambda:
updateChoice("rock"), bg="#FF5733", font=("Arial", 12,
"bold")).grid(row=3,

column=1)
```

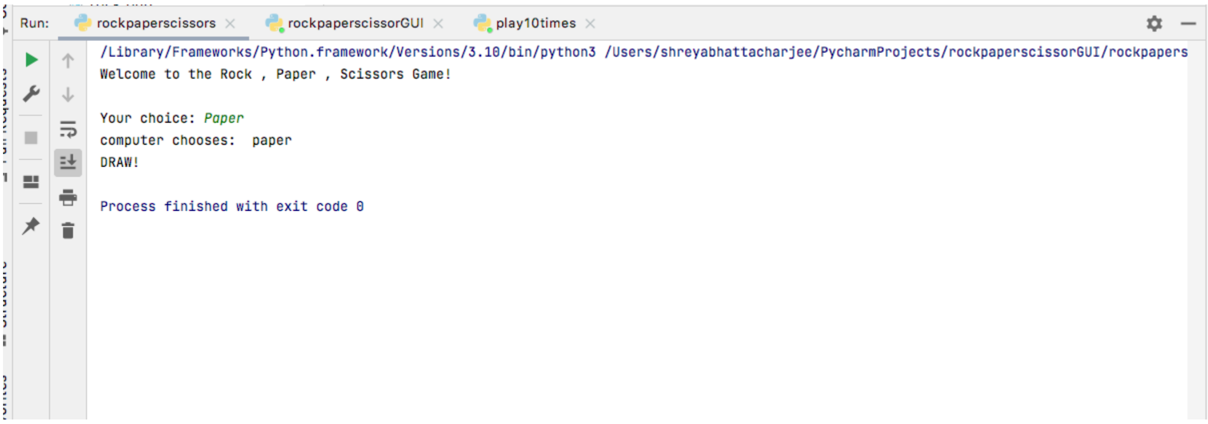
Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

```
Button(root, text="PAPER", command=lambda:
updateChoice("paper"), bg="#33A1FF", font=("Arial", 12,
"bold")).grid(row=3,

column=2)
Button(root, text="SCISSORS", command=lambda:
updateChoice("scissors"), bg="#2ECC71", font=("Arial",
12, "bold")).grid(
    row=3, column=3)

root.mainloop()
```

## Output

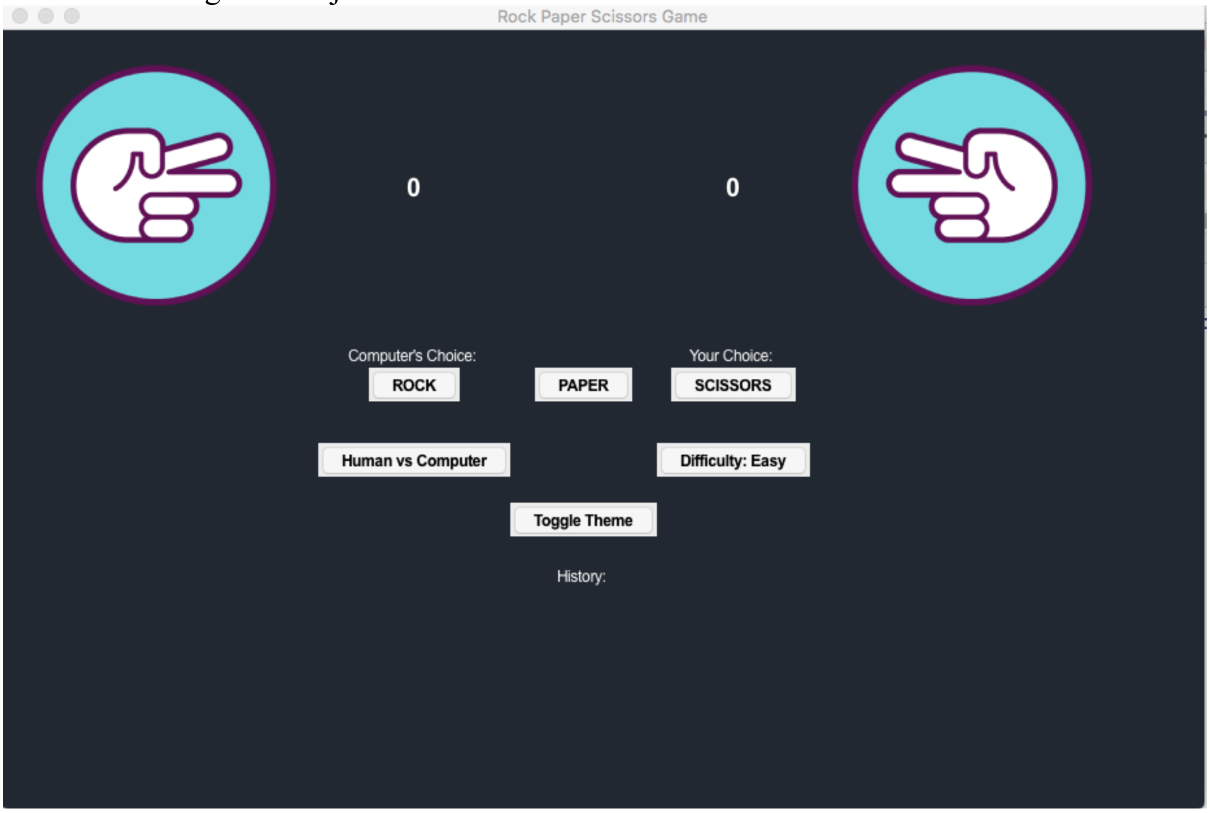


```
/Library/Frameworks/Python.framework/Versions/3.10/bin/python3 /Users/shreyabhattacharjee/PycharmProjects/rockpaperscissorGUI/play10time
Enter your chooice
 Type Rock/Paper/Scissors or Q to quit 2
Enter your chooice
 Type Rock/Paper/Scissors or Q to quit 3
Enter your chooice
 Type Rock/Paper/Scissors or Q to quit rock
 Computer chooses paper
You loose!
Enter your chooice
 Type Rock/Paper/Scissors or Q to quit paper
 Computer chooses scissors
You loose!
Enter your chooice
 Type Rock/Paper/Scissors or Q to quit scissors
 Computer chooses paper
You loose!
Enter your chooice
 Type Rock/Paper/Scissors or Q to quit paper
```



```
/Library/Frameworks/Python.framework/Versions/3.10/bin/python3 /Users/shreyabhattacharjee/PycharmProjects/rockpaperscissorGUI/rockpapers
Welcome to the Rock , Paper , Scissors Game!

Your choice: Paper
computer chooses:  paper
DRAW!

Process finished with exit code 0
```

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project



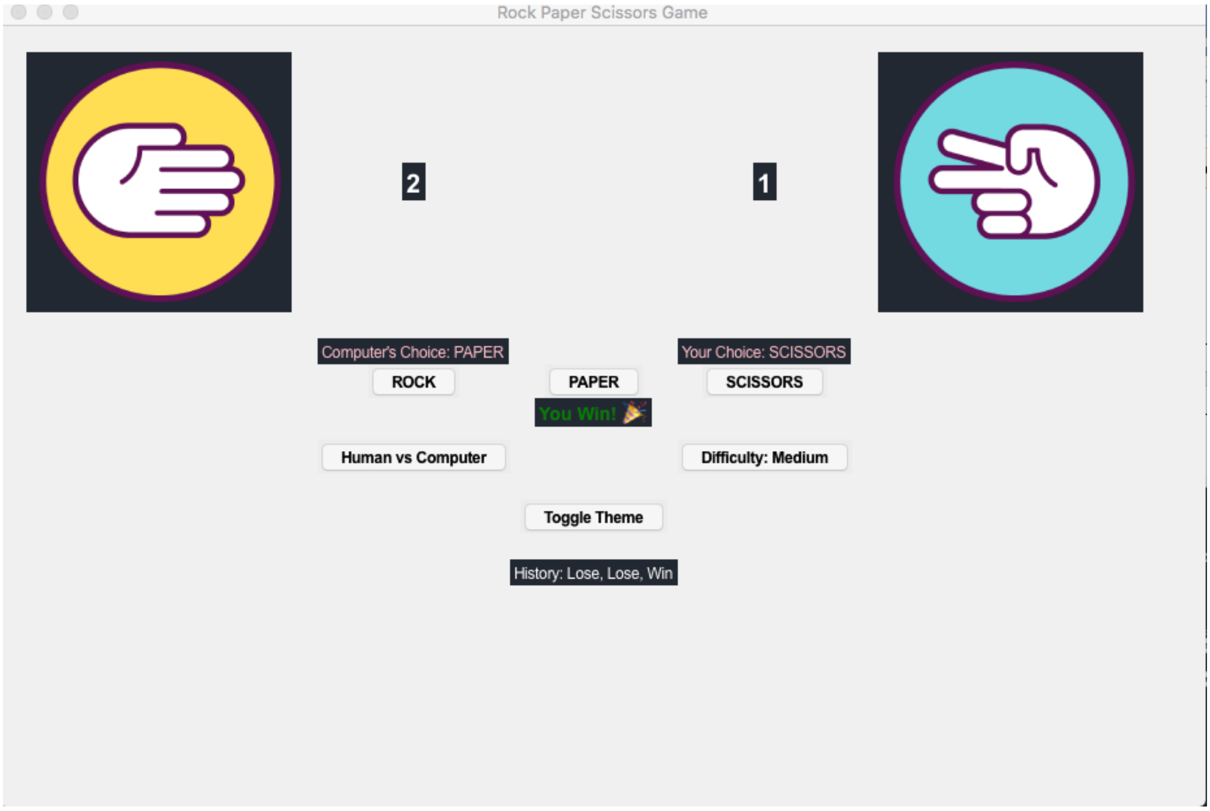
Rock Paper Scissors Game

0 0

Computer's Choice:

Your Choice:

History:



Rock Paper Scissors Game

2 1

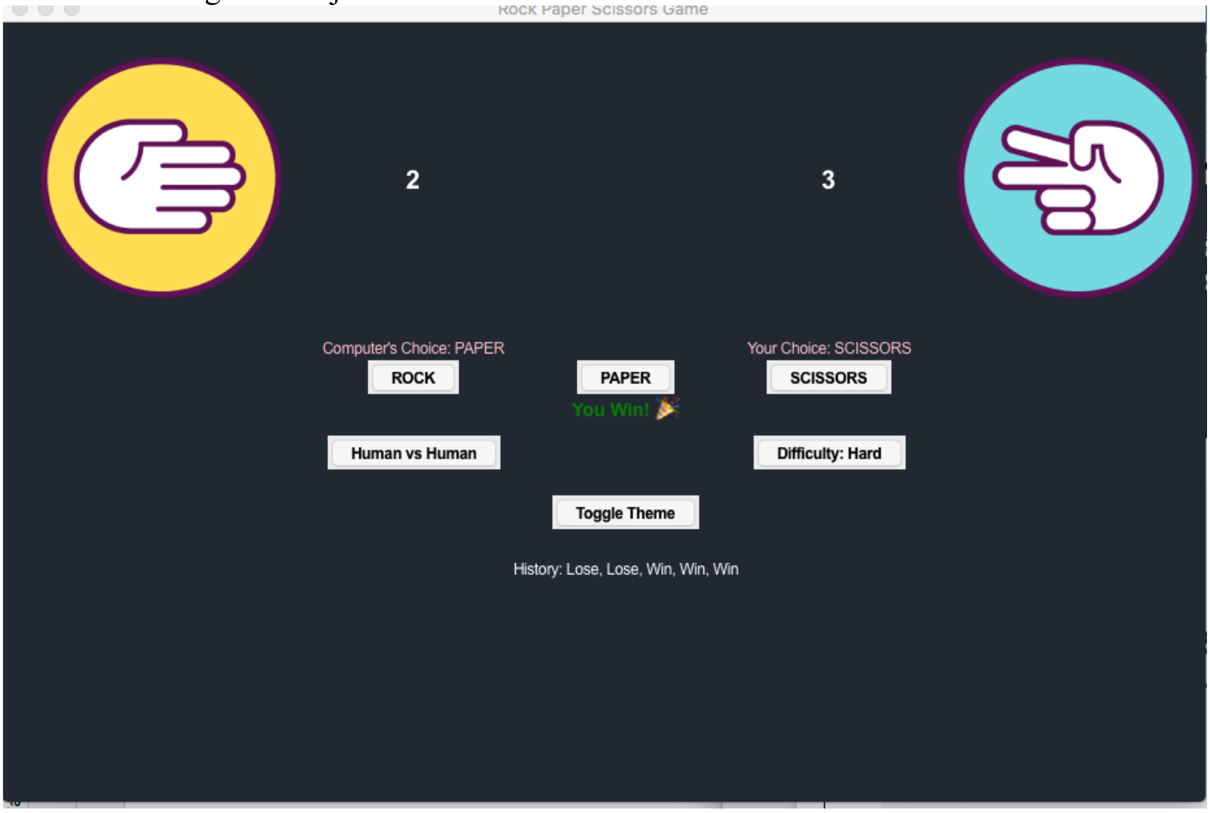
Computer's Choice: PAPER

Your Choice: SCISSORS

You Win! 🎉

History: Lose, Lose, Win

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project



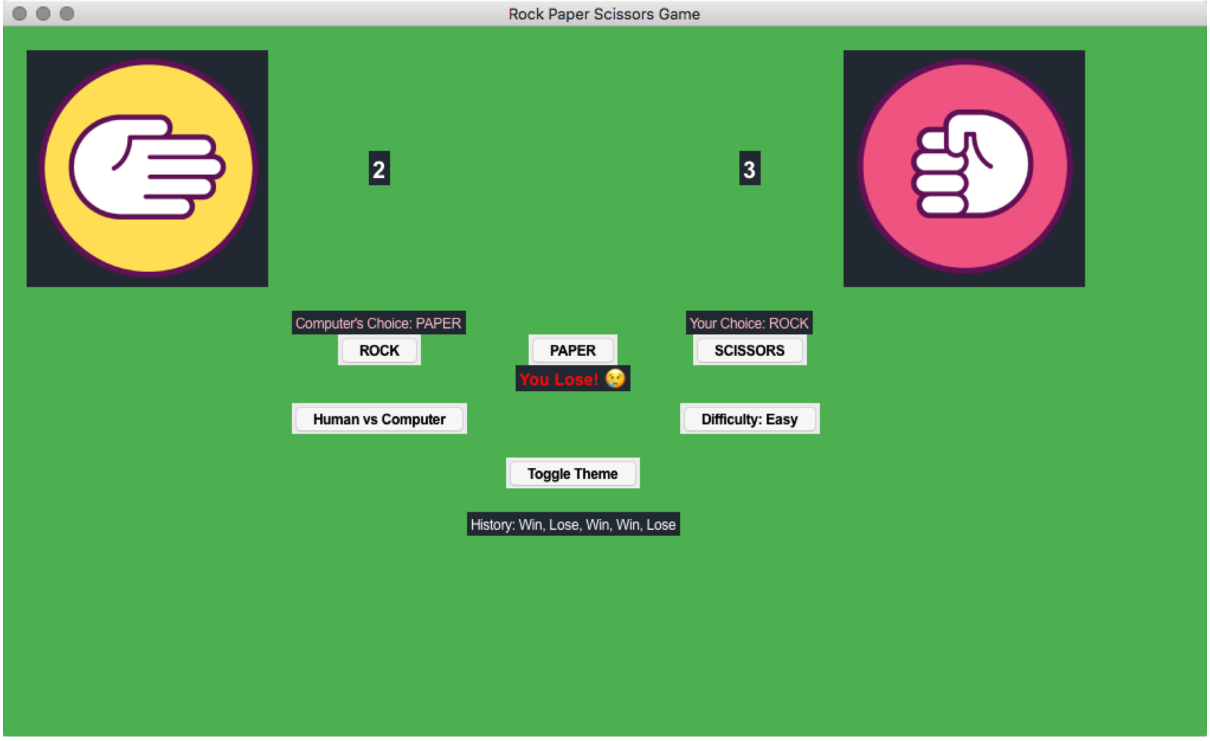
Computer's Choice: PAPER

Your Choice: SCISSORS

You Win! 🎉

Difficulty: Hard

History: Lose, Lose, Win, Win, Win



Computer's Choice: PAPER

Your Choice: ROCK

You Lose! 😢

Difficulty: Easy

History: Win, Lose, Win, Win, Lose

Name: Shreya Bhattacharjee
6861
Artificial Intelligence Project

## Conclusion

The Rock-Paper-Scissors game built using Python and Tkinter successfully provides an engaging and interactive experience for users. It allows players to compete against the computer or each other, with adjustable difficulty levels and a dynamic theme toggle feature. The inclusion of game history tracking and real-time score updates enhances the user experience, making it both entertaining and educational.

This project demonstrates the effective use of GUI programming in Python, integrating images, event handling, and game logic to create a seamless and enjoyable application. Future improvements could include adding more animations, sound effects, or an AI-powered opponent for an even more challenging gameplay experience. 🚀