**PYTHON PROJECT - CSM216**

**Rock Paper Scissor Game**

**Submitted by:** Reddy Manjunath

Submitted to: Aman Kumar Sir

**Registration Number:** 12312640

**DATE: - 25/11/2024**

**Section-K23CH**

**In partial fulfillment of the requirements of the award of the degree of**

**“B. Tech CSE Data Science and Machine Learning”**

**“School of Computer Science and Engineering”**

**Lovely Professional University**

**Phagwara, Punjab**

1

**Acknowledgment**

I would like to express my sincere gratitude to my instructor, **[Instructor's Name]**, for their invaluable guidance and encouragement throughout this project. Their expertise and constructive feedback have played a pivotal role in shaping the design and implementation of this project.

I am also deeply thankful to my peers for their insightful suggestions and support during the brainstorming and testing phases of the project. Their collaboration has been instrumental in improving the user experience and ensuring the game’s functionality.

Additionally, I extend my appreciation to the developers and maintainers of the Python programming language and its libraries, including **Tkinter**, **Pygame**, and **Pillow (PIL)**. These tools provided the foundation for creating the graphical user interface, animations, and sound effects, making the project engaging and interactive.

Lastly, I would like to thank my family and friends for their unwavering encouragement and motivation throughout the development process. Their belief in my abilities has been a source of inspiration and strength.

This project has been a challenging yet rewarding experience, and I am grateful to everyone who contributed to its success.

Name : Reddy Manjunath

Registration number: 12312640

2

**Table of Contents**

**1. Introduction………………………………………………… 4**

**2. Objectives and Scope of the Project……………………….. 5**

**3. Application Tools……………………………………………. 6**

**4.ProjectDesign…………………………………………………7**

**5.FlowChart…………………………..…………………………8**

**6. Project Implementation ………….………………………… 9-11**

**7. Testing and Validation……………………………………….12-17**

**8. Conclusion…………………………………………………….18**

**9. References……………………………………………………19**

**3**

**1. Introduction:**

The Rock-Paper-Scissors game is a universally recognized hand game with simple rules and a wide appeal. Its straightforward mechanics and randomness make it an ideal project to learn and apply programming concepts. This project takes the essence of the classic game and transforms it into a digital, interactive experience using Python.

The purpose of this project is twofold: to develop an engaging application that challenges players and to explore Python’s capabilities for GUI development, multimedia integration, and modular programming. The game is designed to simulate real-world constraints by incorporating a timer for each round, forcing players to make quick decisions. This feature enhances the gameplay and demonstrates Python’s ability to handle event-driven programming effectively.

The significance lies in its scalability and modularity. The project serves as a foundation for learning game development and can be extended to include additional features such as multiplayer modes, leaderboards, or gameplay variations. The implementation highlights best practices in Python programming, such as class-based design, the use of libraries like Tkinter and Pygame, and the handling of user interactions efficiently.

In conclusion, this project is a stepping stone toward mastering Python’s practical applications. It bridges theoretical knowledge and real-world problem-solving, providing an interactive platform to test and improve programming skills while delivering an enjoyable experience to players.

4

**2. Objectives and Scope of the Project**

The Shooter Game in Space The goal of the Pygame project is to create an entertaining

2D arcade game. Designing and implementing a dynamic gaming environment where players may steer a starship, move through space, and engage waves of enemy ships while dodging obstacles is the main goal. This project improves the developer's technical abilities while offering an engaging and dynamic experience that demonstrates the actual application of game development ideas.

**2.1 Objectives**

1. **Interactive Gameplay**: Develop a user-friendly interface that allows players to make choices and receive immediate feedback.
2. **Timed Challenges**: Integrate a countdown timer for each round, progressively reducing the time to add difficulty.
3. **Scalable Design**: Ensure the game structure is modular, allowing easy addition of new features or levels in the future.
4. **Engaging Features**: Incorporate animations and sound effects to enhance the user experience.

**2.2 Scope:**

The game consists of four rounds with time limits of 60, 30, 20, and 10 seconds, respectively. Each round requires the player to score 5 points to progress. If the player fails to meet the target within the given time, the game ends. The design also includes a random choice generator for the computer, which simulates an interactive opponent.

The project focuses on implementing Python libraries effectively. Tkinter is used for GUI development, enabling the creation of interactive buttons and labels. Pygame adds an auditory layer to the experience, while PIL (Pillow) handles image manipulation for animations and flipped displays of computer choices.

While the current implementation is single-player, the project’s modularity allows for future enhancements such as multiplayer modes, difficulty settings, and scoreboards. The scope is intentionally limited to focus on core functionality while ensuring high-quality execution of existing features.

5

**3. Application Tools**

The successful development of this project relied on a range of software tools, programming libraries, and resources. These tools were chosen for their versatility, ease of use, and compatibility with Python.

**3.1 Programming Language**

Python was selected as the primary programming language due to its simplicity, extensive library support, and robust community resources. Its built-in libraries, combined with third-party modules, provided an excellent environment for GUI-based game development.

**3.2 Libraries and Packages**

1. **Tkinter**: Used for designing the graphical user interface. Tkinter facilitated the creation of interactive buttons, labels, and frames to simulate the game environment.
2. **Pygame**: Enabled the addition of sound effects, such as button click sounds and background music, enhancing the player's auditory experience.
3. **Pillow (PIL)**: Used for image manipulation, including resizing and rotating images to display computer choices dynamically and aesthetically.

**3.3 IDE**

Visual Studio Code was used for coding, debugging, and managing the project files. Its extension support, such as Python plugins, streamlined the development process.

**6**

**4. Project Design**

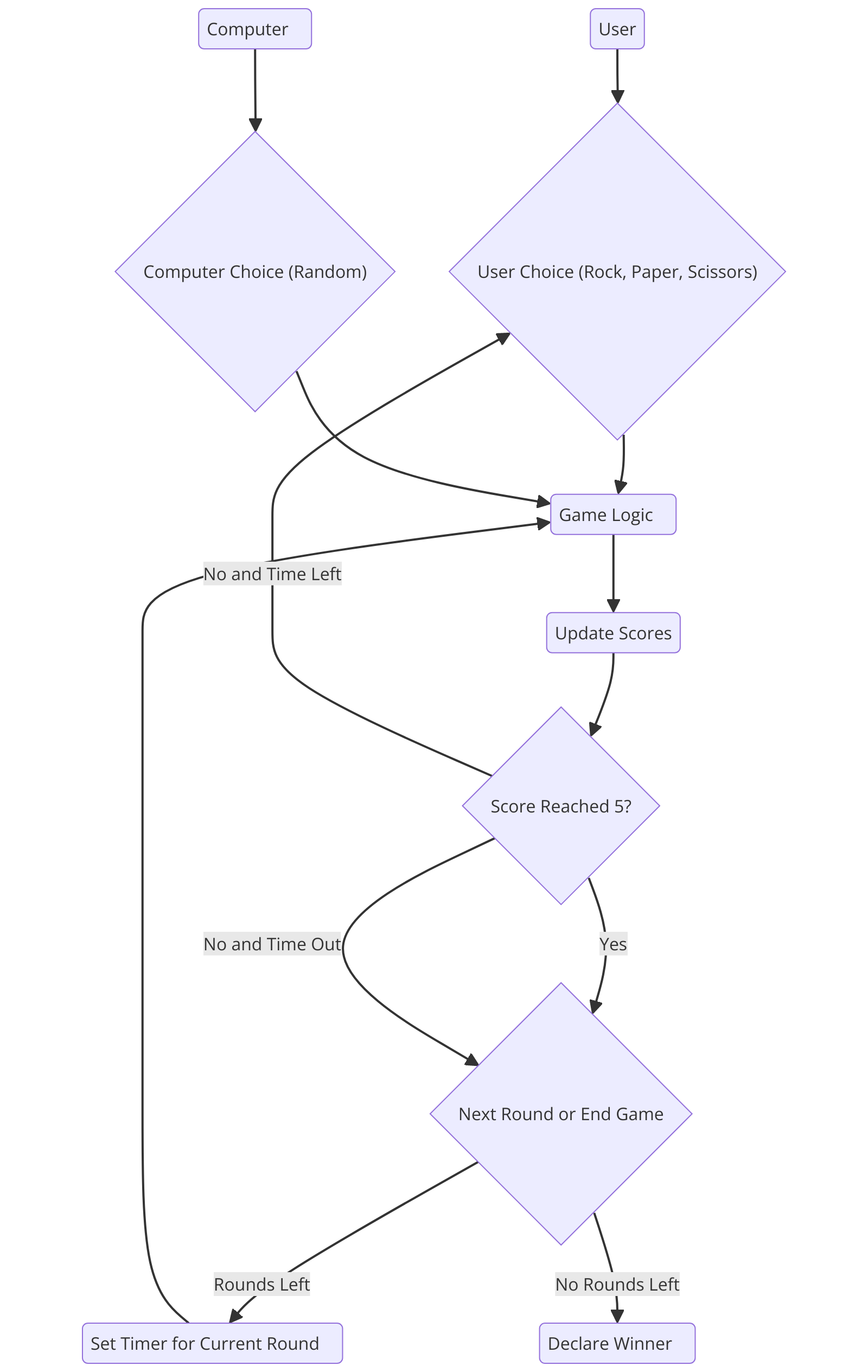
The Rock-Paper-Scissors game is designed around a class-based architecture that promotes modularity and scalability. The primary class, RockPaperScissorsGame, encapsulates all game logic, user interactions, and interface management, ensuring a clean and maintainable codebase.

**Core Components**

1. **User Interface**:
   * The GUI is built using Tkinter, featuring buttons for user choices (Rock, Paper, Scissors) and labels for scores, timers, and the computer’s choice.
   * Each round resets dynamically, updating the displayed components without requiring a separate window.
2. **Game Logic**:
   * A random choice generator determines the computer's selection.
   * The logic compares the user’s and computer’s choices to decide the winner of each round.
3. **Timer and Rounds**:
   * Tkinter's after method manages countdown timers for each round.
   * Each round progresses only if the user scores 5 points within the allocated time.
4. **Animations and Sound**:
   * Pygame plays sounds for button clicks and background music.
   * PIL manipulates images, flipping and resizing them for dynamic feedback.

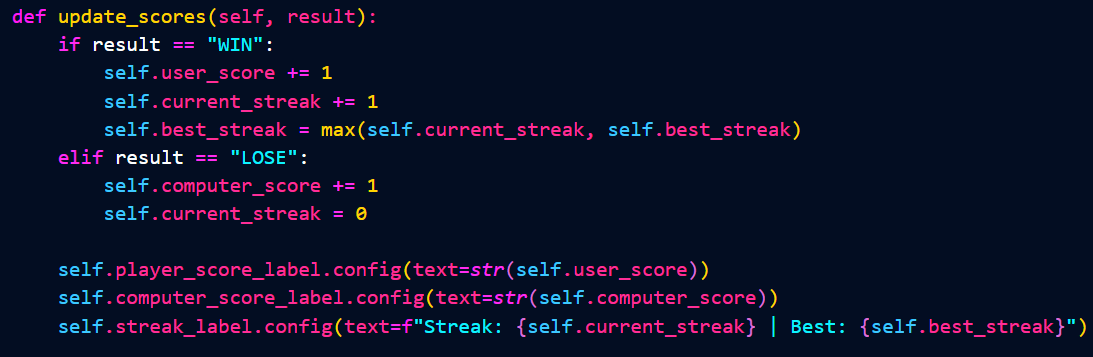
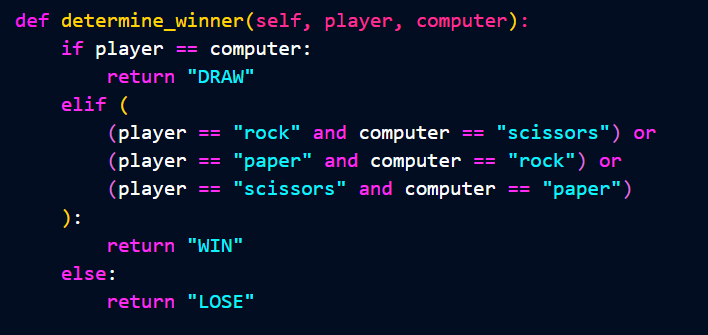
This design ensures that the game is extensible, with a clear separation of concerns, making it easy to add new features or modify existing ones.

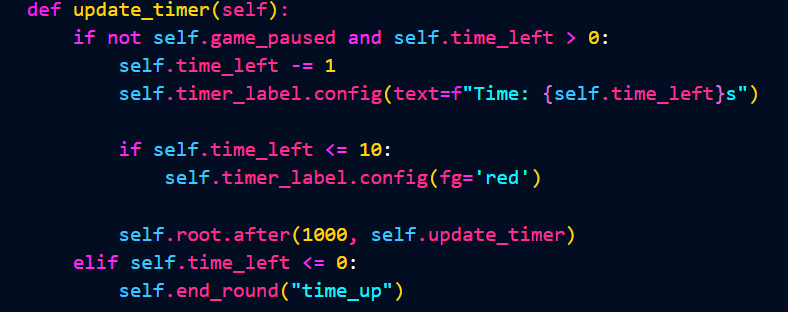
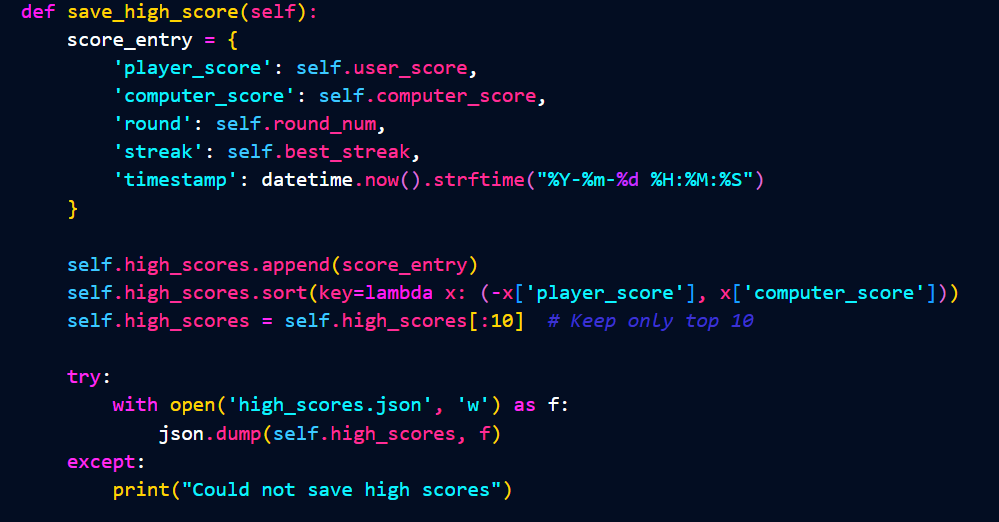
7

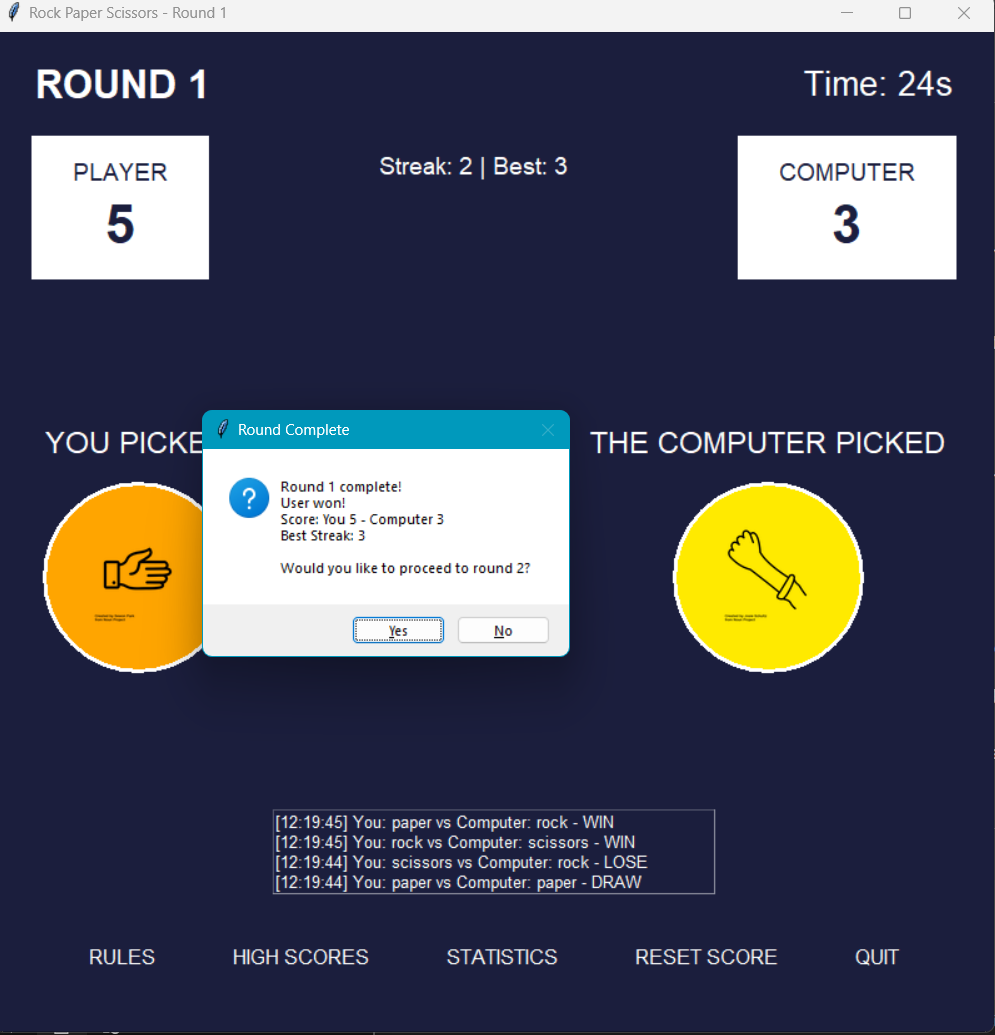
**5. Flowchart** 

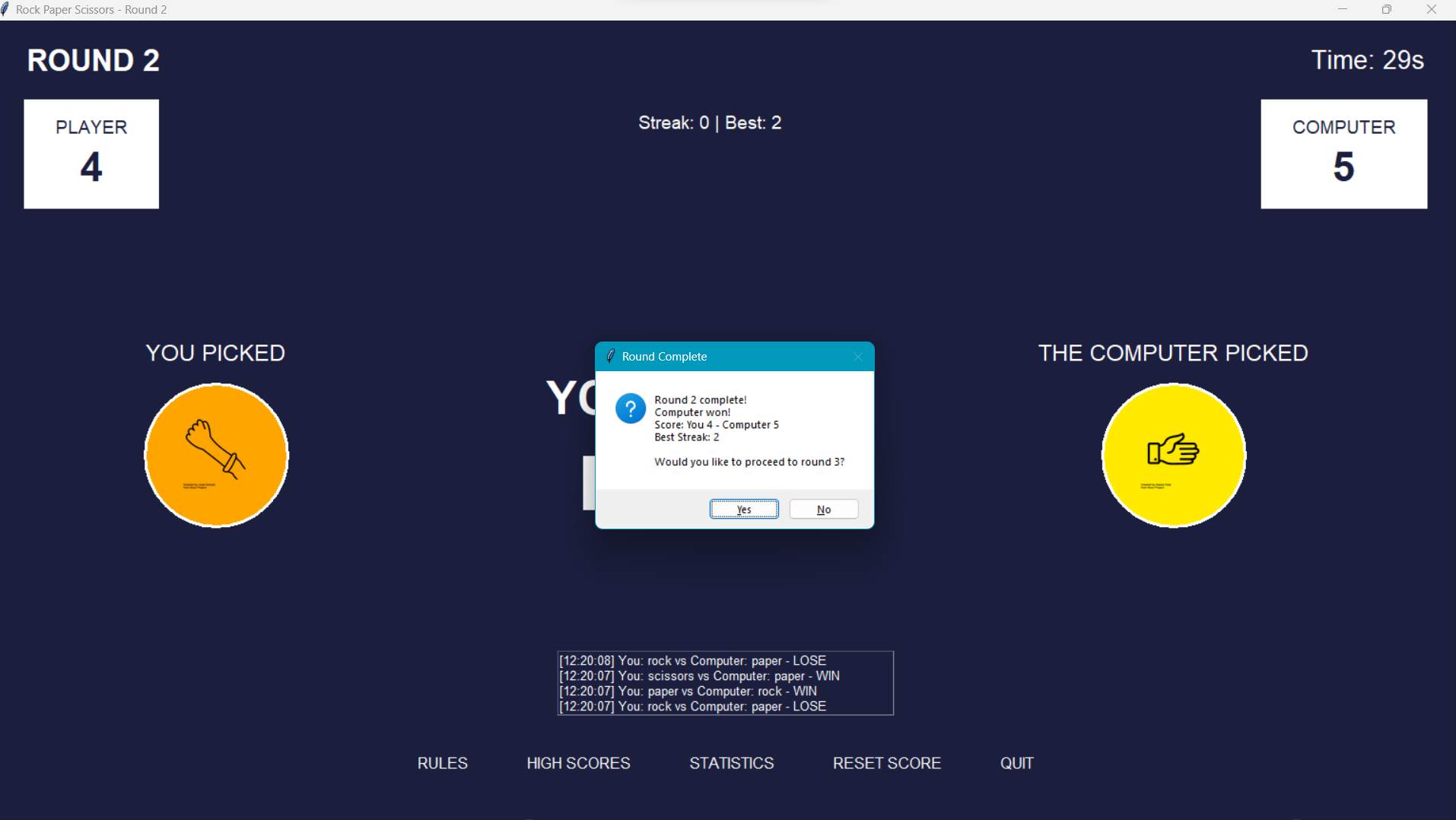
8

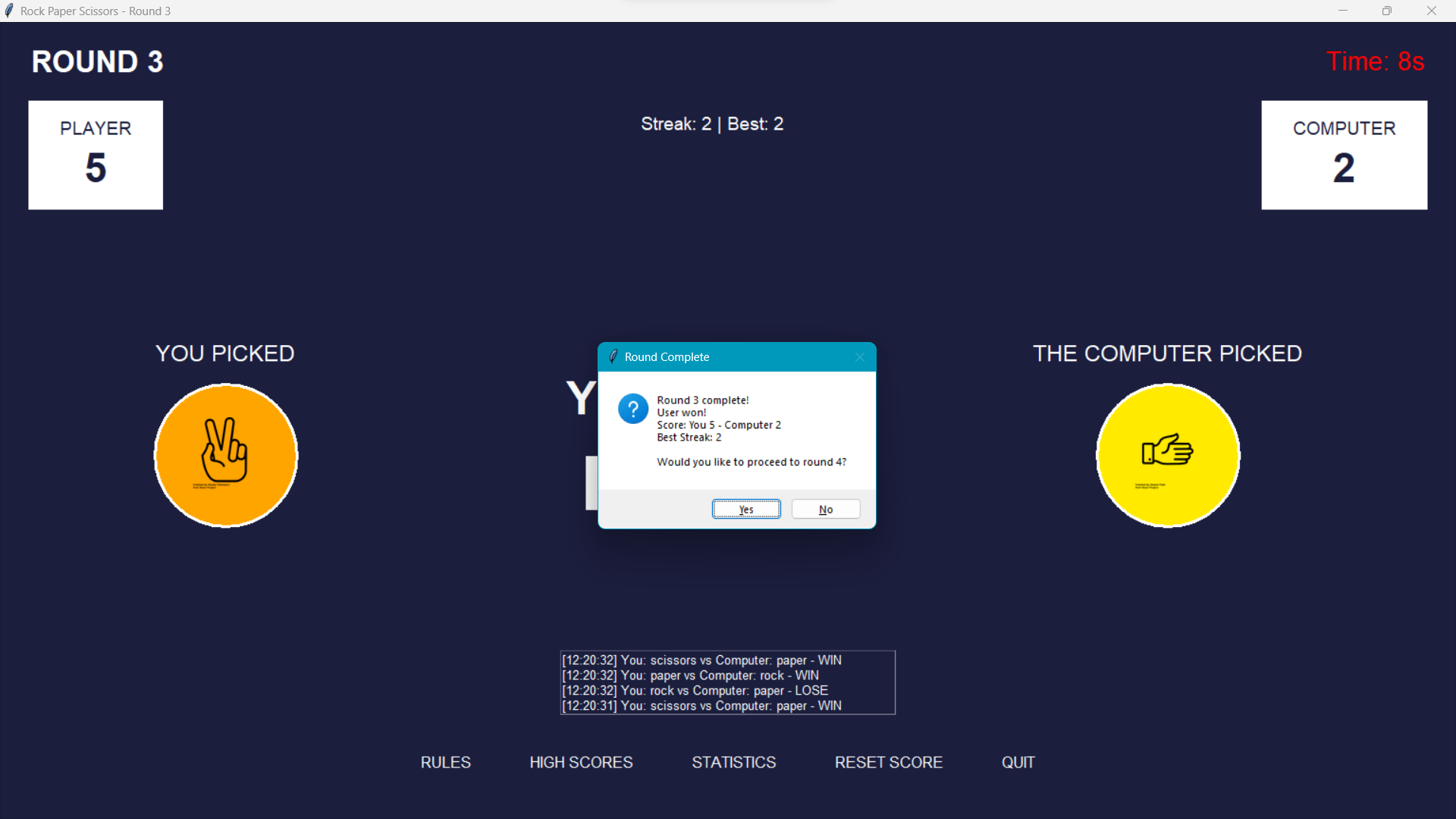
**6. Project Implementation**



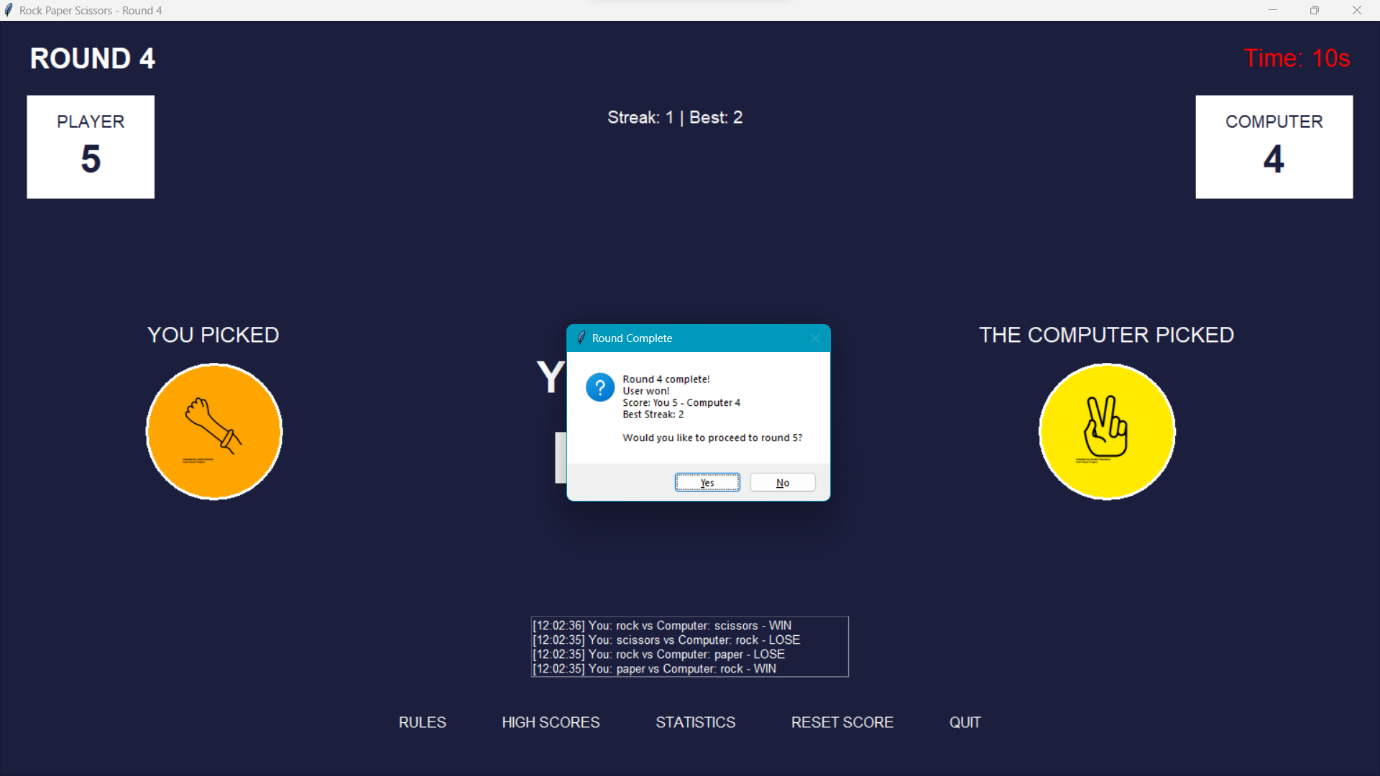


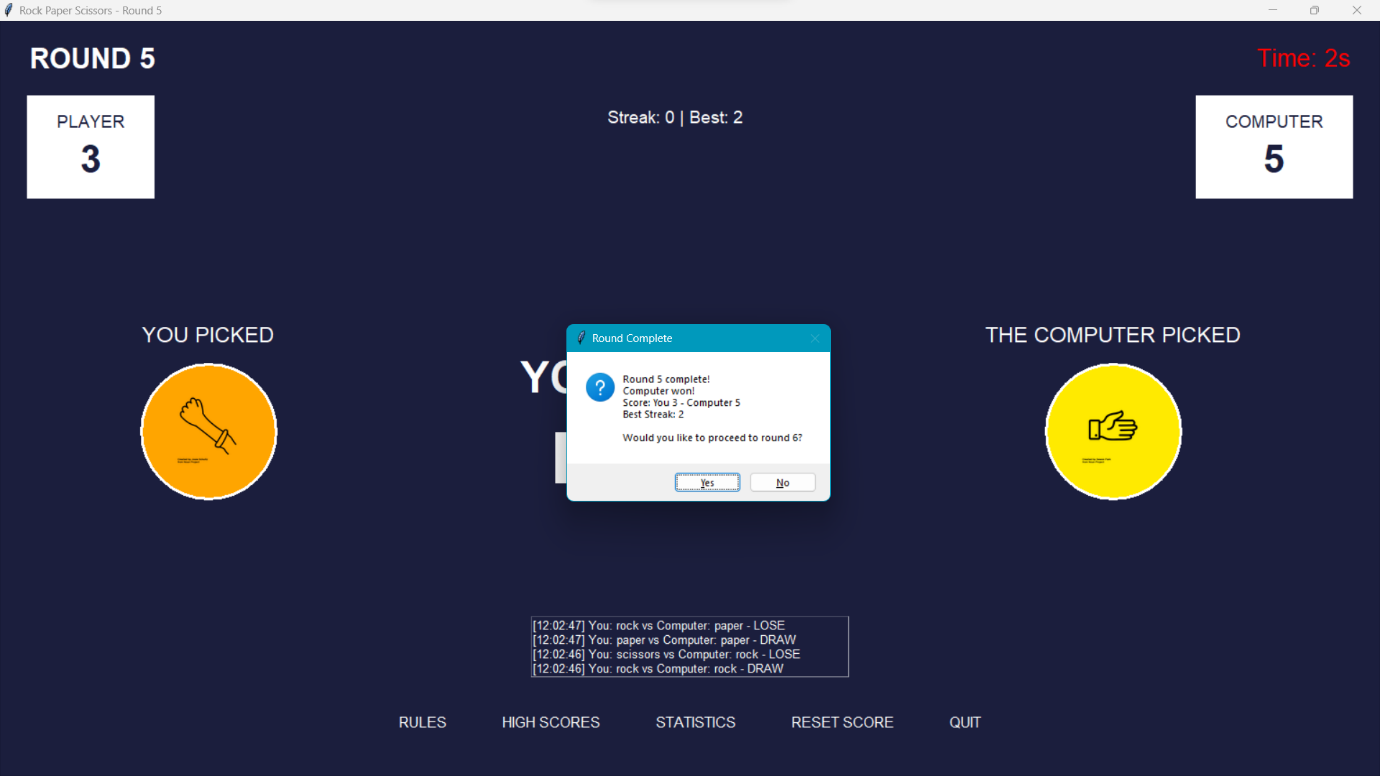
 9





10



****

**11**

**7. Testing and Validation**

**1. Unit Testing Test Cases**

**1.1 Test Case ID: UT\_01**

* **Test Designed by: Reddy Manjunath**
* **Test Priority: High**
* **Test Design date: 22/11/24**
* **Module Name: Game Logic**
* **Test Executed by: Reddy Manjunath**
* **Test Title: Verify the winner determination logic**
* **Test Execution date: 22/11/24**
* **Description: Test if the function determining the winner between the player and computer works correctly.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Call the function with "Rock" and "Scissors"** | **Player: Rock, Computer: Scissors** | **Player wins** | **Player wins** | **Pass** |  |
| **2** | **Call the function with "Paper" and "Rock"** | **Player: Paper, Computer: Rock** | **Player wins** | **Player wins** | **Pass** |  |
| **3** | **Call the function with "Scissors" and "Rock"** | **Player: Scissors, Computer: Rock** | **Computer wins** | **Computer wins** | **Pass** |  |

**12**

**1.2 Test Case ID: UT\_02**

* **Test Designed by: Reddy Manjunath**
* **Test Priority: Medium**
* **Test Design date: 22/11/24**
* **Module Name: Timer Functionality**
* **Test Executed by: Reddy Manjunath**
* **Test Title: Verify the timer decrements correctly**
* **Test Execution date: 22/11/24**
* **Description: Test if the timer counts down correctly during a round.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Start the timer at 30 seconds** | **Initial Time: 30** | **The timer decreases by 1 every second** | **Timer decrements** | **Pass** |  |
| **2** | **Let the timer reach 0** | **-** | **The timer stops and the round ends** | **Timer stops** | **Pass** |  |
| **3** | **Call the function with "Scissors" and "Rock"** | **Player: Scissors, Computer: Rock** | **Computer wins** | **Computer wins** | **Pass** |  |

**13**

**1.3 Test Case ID: UT\_03**

* **Test Designed by: Reddy Manjunath**
* **Test Priority: High**
* **Test Design date: 22/11/24**
* **Module Name: Score Updates**
* **Test Executed by: Reddy Manjunath**
* **Test Title: Verify the score update mechanism**
* **Test Execution date: 22/11/24**
* **Description: Test if the user's and computer's scores are updated correctly after each round.**
* **Score update logic is implemented.**
* **The round has started, and choices are made.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **The user wins the round** | **Player: Rock, Computer: Scissors** | **The user's score increments by 1** | **User's score increments** | **Pass** |  |
| **2** | **The computer wins the round** | **Player: Paper, Computer: Scissors** | **The computer's score increments by 1** | **Computer's score increments** | **Pass** |  |
| **3** | **Call the function with "Scissors" and "Rock"** | **Player: Scissors, Computer: Rock** | **Computer wins** | **Computer wins** | **Pass** |  |

**14**

**2 System Testing Test Cases**

**2.1 Test Case ID: ST\_01**

* **Test Designed by: Reddy Manjunath**
* **Test Priority: High**
* **Test Design date: 22/11/24**
* **Module Name: Game Start and Initialization**
* **Test Executed by: Reddy Manjunath**
* **Test Title: Verify the game starts successfully with all UI elements loaded**
* **Test Execution date: 22/11/24**
* **Description: Test if the game starts correctly and displays all required elements.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Launch the game** | **-** | **A game window opens successfully.** | **Game window opens** | **Pass** |  |
| **2** | **Verify UI elements** | **-** | **UI elements (buttons, labels, timer) are visible** | **All elements visible** | **Pass** |  |
| **3** | **Call the function with "Scissors" and "Rock"** | **Player: Scissors, Computer: Rock** | **Computer wins** | **Computer wins** | **Pass** |  |

**15**

**2.2 Test Case ID: ST\_02**

* **Test Designed by: Reddy Manjunath**
* **Test Priority: Medium**
* **Test Design date: 22/11/24**
* **Module Name: Round Transitions**
* **Test Executed by: Reddy Manjunath**
* **Test Title: Verify transitions between rounds**
* **Test Execution date: 22/11/24**
* **Description: Test if the game transitions smoothly between rounds when the user achieves the target score.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **The user scores 5 points.** | **-** | **The game transitions to the next round.** | **Transition occurs** | **Pass** |  |
| **2** | **Verify UI updates** | **-** | **UI reflects new round details (timer, scores)** | **UI updates correctly** | **Pass** |  |
| **3** | **Call the function with "Scissors" and "Rock"** | **Player: Scissors, Computer: Rock** | **Computer wins** | **Computer wins** | **Pass** |  |

**16**

**2.3 Test Case ID: ST\_03**

* **Test Designed by: Reddy Manjunath**
* **Test Priority: High**
* **Test Design date: 22/11/24**
* **Module Name: End Game Functionality**
* **Test Executed by: Reddy Manjunath**
* **Test Title: Verify game ends correctly when the timer runs out**
* **Test Execution date: 22/11/24**

**Description: Test if the game ends and displays the appropriate message when time runs out without meeting the target score.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Test Data** | **Expected Result** | **Actual Result** | **Status** | **Notes** |
| **1** | **Let the timer reach zero.** | **-** | **The game ends with a "Time's Up" message** | **Correct message** | **Pass** |  |
| **2** | **Verify no further actions.** | **-** | **The game does not accept further inputs.** | **No inputs processed** | **Pass** |  |
| **3** | **Call the function with "Scissors" and "Rock"** | **Player: Scissors, Computer: Rock** | **Computer wins** | **Computer wins** | **Pass** |  |

**17**

**8. Conclusion**

The Rock-Paper-Scissors game project demonstrates the practical application of Python for developing interactive, multimedia-rich applications. By combining GUI components with animations, sound effects, and dynamic gameplay mechanics, the project successfully recreated a classic game in a modern digital format. The implementation of timed rounds with progressively increasing difficulty provided an additional layer of challenge, engaging users and testing their decision-making under pressure.

One of the key achievements of this project is its modular design, which ensures scalability and maintainability. The use of a class-based structure simplified the organization of game logic, user interactions, and round transitions. Additionally, the integration of Python libraries such as Tkinter, Pygame, and PIL showcased the flexibility of Python for handling diverse programming tasks, including graphics, sound, and real-time events.

While the project achieved its primary goals, there are areas for improvement. Expanding the game to include multiplayer modes, leaderboards, and additional themes could enhance its replayability and user experience. Future iterations may also refine the graphical interface and incorporate more advanced animations.

In conclusion, this project serves as an excellent learning opportunity, blending programming skills with creative problem-solving. It not only highlights Python's capabilities but also lays the groundwork for further enhancements and applications.

**18**

**9. Reference**

1. Python Documentation: <https://docs.python.org/>
2. Tkinter Documentation: <https://docs.python.org/3/library/tkinter.html>
3. Pygame Documentation: https://www.pygame.org/docs/
4. PIL (Pillow) Documentation: <https://pillow.readthedocs.io/>

19