**Lt. M.J. Kundaliya Arts and Commerce Mahila College**

**(Computer Science Department)**

**Rashtriya Shala Campus, Rajkot**

**Fight Game**

By

Gohel Uravshi.

Under Guidance

Of

Pr. Anil Makvana

Submitted to the Computer Science Department

In partial fulfillment of the requirement

For the degree of

**Bachelor**

**Of**

**Computer Application**

**Semester -06 Date: 11-10-2024**

**Index**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **TOPIC NAME** | **PAGE NO.** |
|  | Preface | 4 |
|  | Achnologdgement | 5 |
| 1 | **CONTENTS** |  |
| 2 | 1.0 INTRODUCTION | 6 |
| 3 | 1.1 Project Summary | 8 |
| 4 | 1.2 Purpose : Goals & Objectives | 8 |
| 5 | 1.3 Hardware and Software Requirements. | 11 |
| 6 | **2.0 SYSTEM ANALYSIS** | 13 |
| 7 | 2.1 Study of Current System | 14 |
| 8 | 2.2 Problem and Weaknesses of Current System | 16 |
| 9 | 2.3 Requirements of New System | **17** |
|  |  |  |
| 10 | 2.4 Feasibility Study | 19 |
| 11 | 2.5 **DATA MODELING** | 22 |
| 12 | 2.5.1 E-R Diagram | 23 |
| 13 | 2.5.2 System Activity or Object interaction Diagram | 24 |
| 14 | 2.5.3 Data Dictionary | 25 |
| 15 | 2.6 Functional and Behavioural Modelling | 26 |
| 16 | 2.6.1 Context Diagram | 27 |
| 17 | 2.6.2 Data Flow Diagram | 28 |
| 18 | **3.0 TESTING** | 30 |
| 19 | 3.1 Testing Plan | 30 |
| 20 | 3.2 Testing Methods | 32 |
| 21 | 4.0 Screen shots and User manual | 36 |
| 22 | 5.0 Limitation and Future Enhancement | 41 |
| 23 | 6.0 Conclusion and Discussion | 42 |

**Preface**

In 21st Century of Technology, Computer and related service is the most useful, automatic and highly speed of work. B.C.A is course in which practical aspect is important as theoretical aspect. In the present world importance of education is increasing. The project training in a student’s life is like a live experience in the industry. If we have experience then & then we get top of the level.

I am developing project in 6th Semester of B.C.A I am the student of Lt. M. J. Kundaliya Arts & Commerce Mahila College [Computer Science Department] Is Affiliated with Saurashtra University, Rajkot.

I am creating a project which is game where a user have to first login and then they can play easily. I want to make it user friendly and it is. It is very easy to play it’s fight game which is anyone can play.

I went through and study necessary information, which is collected and presented in this report. I am trying my best in this project.

I also take care that are information provided in the project is true and project is in a smooth running condition and error free.

**ACKNOWLEDGEMENT**

It is a genuine pleasure to express my profound gratitude and deep regards to my Internal Guide**,** **Prof. Anil Makvana** and our HOD **Prof**. **SHAMIK** **RATHOD** for their exemplary guidance**,** monitoring and constant encouragement**.** I would like to express my special thanks to  **LT. M.J.KUNDALIYA COLLEGE** who gave me the golden opportunity to do this wonderful project on the topic **“Fight Game”,** which helped me in doing a lot of Research and I came to know about so many new things**.**

Thanks to All

Your Faithfully,

**Gohel Uravshi**

**INTRODUCTION**

**\* Preliminary investigation**

I am Gohel Urvashi from BCA sem-6. I decide to prepare game that a user can play itself easily as I decide to make it user friendly.

The Fight Game is an engaging and dynamic combat simulation that allows players to immerse themselves in a world of martial arts and strategy. Built using Python, this game leverages object-oriented programming principles to create a robust and extensible system that can communicate various characters, moves, and game mechanics.

This project aims to working great in this project and make it user friendly. This project aims to Model the working of Fight Game “Naruto vs Sasuke” so, this is the name of the game.

In this project I used Python as the programming language. Using this language to do my project I learned many new things about this language.

* **Core Features:**

1. Character Creation:

In this game there are two characters Naruto and Sasuke which will fight in this game. I am using the game character which are anime in the market. A lot of people watches them as anime. So, I used them in this game.

1. Combat Mechanics:

The game employs a turn-based combat system where players can choose from a variety of moves, including attacks, defenses, and special abilities. In this game I used a weapon which name is shuriken to beat the opposite player.

1. Health and Stamina System:

Each character has a health and stamina meter that affects their performance in combat. Players must manage their stamina wisely to execute powerful moves while avoiding exhaustion.

1. Multiplayer Mode:

Players can compete against each other in real-time, testing their skills and strategies in head-to-head combat. The players having health bars which shows how much stamina they have in the game.

The Fight Game is “Naruto v/s Sasuke” is implemented using python, utilizing libraries such as pygame for graphics and user interface elements. It’s really fun to play this game.

* 1. **Project Summary:**
* Fight Game is an engaging 2D fighting game developed in Python, designed to provide players with an exciting and competitive experience. Players can choose from a diverse roster of characters, each with unique abilities and fighting styles, to battle against friends.
* The game features intuitive controls that allow players to execute weapon to kill an enemy. The game includes multiple game modes, such as single-player and re-playable experience.
* This game is aimed at casual gamers and fighting game enthusiasts of all ages. It’s easy-to-learn mechanics make it accessible to newcomers, while the depth of game play offers challenges for seasoned players.
  1. **Purpose : Goals & Objectives:**
* The Fight Game project aims to design and develop a computerized system that enables efficient management of user information, and other related operations**.**
* **Purpose:**
* The purpose of the fight game is to create an engaging and interactive experience that allows players to immerse themselves in a competitive environment. The game will focus on strategy, skill development, and player engagement, providing a platform for players to test their abilities against AI or other players.
* **Goals:**
* Engaging game-play: Develop a game that is fun and keeps players engaged through challenging mechanics and rewarding experiences.
* Skill Development: Create a system that encourages players to improve their skills over time, offering various levels of difficulty and training modes.
* Multi Player functionality: Implement a multiplayer mode that allows players to compete against each other, fostering a sense of community and competition.
* Visual and Audio Appeal: Enhance the game with appealing graphics and sound effects to create an immersive atmosphere.
* **Objectives:**

1. **Game Mechanics:**

* Develop core mechanics such as movement, attacks, and defense.
* Implement a combo system that rewards skilled play.
* Create a health and stamina system to add depth to combat.

1. **User Interface:**

* Design an intuitive user interface that allows players to easily navigate menus and understand game mechanics.
* Include tutorials and tooltips to help new players learn the game.

1. **Character Development:**

* There are two characters with unique abilities and fighting styles.
* Implement a re-play system in this game.

1. **Game Modes:**

* Develop multiple game modes, including single-player, multiplayer, and training modes.
* Create a tournament mode where players can compete for high scores or rankings.

**S**

* 1. **Hardware and Software Requirements:**

To develop a robust and efficient Fight Game using Python and MYSQL Server, the following hardware and software requirements are recommended:

**Hardware Requirements:**

➢ Intel I3 2.8 GHz Processor and Above

➢ RAM 1 GB and Above

➢ HDD 20 GB Hard Disk Space and Above

**Software Requirements:**

➢ WINDOWS OS (Windows 11)

* Python language, IDLE for python programming**.**
* Database MYSQL SERVER for (Backend).
* **MYSQL SERVER (BACKEND):**

MYSQL SERVER is an open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for accessing and managing data. It is one of the most popular database systems in the world, widely used for web applications, data warehousing, and various other applications. Here are some key details about MySQL Server.

* **Features:**

1. **Scalabilities: MYSQL** can handle large databases and high-traffic applications, making it suitable for both small and large-scale applications.
2. **Replication: MYSQL** supports various replication methods (master-slave, master-master) for data redundancy and load balancing.
3. **Partitioning: MYSQL** allows partitioning of tables to improve performance and manageability.
4. **Full-text search: MYSQL** provides full-text indexing and searching capabilities for efficient text searching.
5. **SYSTEM ANALYSIS:**

* The system analysis phase is regarded to be one of the most important phases in the system development life cycle. It is extremely important that the software developer make through study of the existing system through study of the system is made and need.
* Requirement analysis is done in order to understand the problem which the software system is to solve e.g., the problem could be computerizing the existing manual system or developing a completely new automated system or a combination of the two.
* Understanding the requirement of the system is an important and major task. The significance in requirement analysis is on identifying what is needed from the system and not how the system achieves it’s goals.
* The most important objective behind automation is to minimize paper work. System analysis is most important phase in project.
  1. **Study of Current System:**
* Current system means the Hardware platform/operating system/management platform on which the server license is currently installed.
* A current system refers to the existing systems, procedures, or structures that are currently in place. The purpose of analyzing current system is to understand their functionality, identify their strengths and weaknesses, and plan for their replacement, integration, or reuse in relation to new system.
* Airline Management System software is a complex system that often involves several different components and technologies.

* **Advantages:**
* **Ease of Learning:** Python has a simple syntax that is easy to understand, making it accessible for beginners.
* **Rapid Development:** The high-level nature of python allows for quicker prototyping and development cycles.
* **Large Community Support:** Python has a vast community, which means plenty of resources, tutorials, and libraries are available.
* **Cross-platform Compatibility:** Python games can run on various operating systems without significant changes to the code.
* **Rich Libraries:** Libraries like pygame and panda3D provide the tools and functionalities specifically for game development.
* **Integration Capabilities:** Python can easily integrate with other languages and technologies allowing for more complex game features.
* **Disadvantages:**
* **Performance issues:** Python is generally slower than compiled languages like c++ or c#, which can be a drawback for performance intensive games.
* **Limited Graphics Capabilities:** while there are libraries available, python may not offer the same level of graphics performance and capabilities as other languages.
* **Less industry adoption:** Many games studios prefer languages like c++ for their performance and control, leading to fewer job opportunities for python game developers.
* **Memory Consumption:** Python can consume more memory than other languages which may be a concern for larger games.
  1. **Problem and Weaknesses of Current System**
* **Problems:**
* Python may not handle real-time processing as efficiently as other languages, leading to lag during gameplay, especially in fast-paced fighting scenarios.
* Poor memory management can lead to performance issues, such as frame drops or crashes during intense battles.
* If the game is not designed with cross-platform capabilities, it may limit the player base to a single operating system.
* Python's libraries may not support advanced graphics and animations, resulting in a less visually appealing game.
* Compared to other game development environments, there may be fewer resources, tutorials, and community support for developing fighting games in Python.
* **Weaknesses**
* The user interface may not be intuitive, making it difficult for players to navigate menus, select characters, or access game modes.
* Players may have few options to customize characters or gameplay settings, leading to a less personalized experience.
* If the game includes online features, inadequate security can expose player data to breaches or cheating.
* The game may not provide sufficient feedback to players regarding their performance, making it hard to improve skills.
* Without diverse game modes or un-lockable content, players may lose interest quickly after completing the game.
  1. **Requirements of New System**
* **User Interface Requirements:**

1. **Main menu page:** The system should have a main menu page that allows users to start a new game, load a saved game, view settings, and exit the game.
2. **Character Selection page:** The system should have a user-friendly character selection page where players can choose their fighters and view their stats.
3. **Settings page:** The system should have a settings page to allow users to adjust game settings such as sound, controls, and difficulty levels.
4. **Game over page:** The system should have a game over page that displays the results of the match and options to restart or return to the main menu.
5. **Game Arena page:** The system should have a game arena page where the actual fighting takes place, displaying the characters, health bars, and action prompts.

* **Functional Requirements:**

1. **User Authentication:** The system should have a secure login mechanism to authenticate users and allow them to save their progress and settings.
2. **Character Abilities:** The system should enable each character to have unique abilities and moves that can be executed during combat.
3. **Score Tracking:** The system should keep track of scores, wins, losses, and player statistics throughout the game.
4. **Save/Load Game:** The system should allow users to save their game progress and load it later.

* **Non-Functional Requirements:**

1. **Scalabilities:** The system should be designed to handle multiple players and matches simultaneously without performance degradation.
2. **Security:** The system should ensure the security of user data, including login credentials and game progress, to prevent unauthorized access.
3. **Performance:** The system should have fast response times for user inputs and smooth animations during game-play to enhance the user experience.
4. **User Experience:** The system should provide an engaging and intuitive user experience, with clear instructions and feedback during game-play.
   1. **Feasibility Study**

* A Feasibility is an analysis of the practicality of a proposed project or system.
* A Feasibility study is the detailed study expanded from the result of initial investigation.
* A Feasibility study is a preliminary exploration of a proposed project or under taking to determine its merits and viability.
* It is the test of a system proposal according to its work ability, impact on organization, ability to meet user needs and effective use of resources.
* **There are four Key Considerations involved in feasibility analysis namely.**

1. Technical Feasibility
2. Operation Feasibility
3. Economical Feasibility
4. Social Feasibility
5. **Technical Feasibility:**

* Python is a versatile language with a rich ecosystem of libraries (e.g., pygame for graphics, Numpy for calculations) that can facilitate game development.
* The game can be developed on various platforms (windows, macos, Linux) using python, making it accessible to a wide audience.
* The use of object-oriented programming allows for modular design, making it easier to implement features.
* Python may not be as fast as compiled languages like c++, but for a 2D Fight Game, it is generally sufficient performance can be optimized by profiling and refining code as needed.

1. **Operational Feasibility:**

* Assess the skills of development team. Familiarity with python and game development concepts is crucial.
* If the team lacks experience, consider training or hiring skilled developers.
* Estimate the time required for each phase of development (design, coding, testing, and deployment). A realistic timeline will help manage expectations and resources.
* Plan for ongoing maintenance and updates post-launch. This includes bug fixes, feature enhancements, and community support.

1. **Economic Feasibility:**

* **Outline the costs associated with development, including software license, hardware, marketing, and potential salaries for team members.**
* **Consider how the game will generate revenue.**
* **Options include:**
* Selling the game as a one-time purchase.
* Offering the game for free with in-game purchases or ads.
* Using platforms like kick starter to raise funds before development.
* Research the target audience and competitors, Identify trends in the gaming industry, particularly in the fighting game genre, to gauge potential demand.

1. **Social Feasibility:**

* Social feasibility is one of the feasibility study where the acceptance of the people is consider regarding the product to be launched.
* Social feasibility in a game project using python refers to the evaluation of the project on the users, communities, and others.
* This assessment helps to ensure that the game is socially acceptable and sustainable.
* The game should be designed in a way that is user-friendly and meets the needs and expectations of the users.
  1. **Data Modelling**
* A data model for Fight game involves designing a structured representation of the data entities, attributes, and relationships to store and manage user information.
* Here's a high-level overview of the data model:
* Game Initialization:
* Set up the game environment
* Load assets (images, sounds, etc)
* Initialize game variables
* Player Module:
* Define player attributes (health, strength, defense, speed)
* Implement player actions (attack, defend, special moves)
* Handle player input (keyboard, controller)
* Enemy Module:
* Define enemy attributes (health, strength, behavior)
* Implement enemy actions (attack, defend, special moves)
* Create different enemy types with varying difficulty levels
* User Interface Module:
* Create a graphical user interface (GUI) for the game
* Display player and enemy stats, health bars, and action buttons
* Implement menus for starting the game, pausing and quiting
* Sound and Music Module:
* Load and play background music and sound effects.
* Implement sound effects for actions
  + 1. **E-R diagrams**

Player Winner

**Play Game**

**High Score**

**Player**

* + 1. **System Activity**
    2. **Data Dictionary**

**1) Login:**

Here we require a security for admin for that we think that we must provide a login facility because any admin cannot do the changes without permission.

|  |  |  |  |
| --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Description** |
| Username | varchar | 30 | Primary Key |
| Password | varchar | 30 | - |
| Score | Int | 30 | - |

* 1. **Functional and Behavioural Modelling**

Functional modeling for a flight game involves identifying the key functions that the game needs to perform to provide an engaging experience for its users. Here are some of the key functions that a flight game should have:

1. Flight Controls:

* The system should allow players to control the aircraft's movements, including takeoff, landing, and navigation.

1. Mission Objective:

* The game should provide various missions or objectives for players to complete, such as delivering cargo, performing aerial stunts, or engaging in dogfights.

1. Scoring System:

* The system should track player scores based on completed missions, time taken, and accuracy of maneuvers.
* **Behavioural Modelling**

1. Use Cases:

* The system should have use cases that describe how players interact with the game, such as:
* Starting a new game
* Completing a mission

1. State Transition Diagram:

* The system should have state transition diagrams that describe the different states of the game, such as:
* Main-Menu
* Mission Completed
* Game over
  + 1. **Context Diagram**

**System (game)**

**Player**

**2.6.2 Data Flow Diagram**

* **1st Level DFD**

**Level Complete**

**P1 v/s P2**

* **2nd Level DFD**

**Player**

Play player Data

Score data

1. **Testing**

Testing is a crucial phase in any software development project, including Fight game project. The goal of testing is to ensure that the system meets the required specifications, is free from defects, and provides a good user experience.

**(3.1)Testing Plan:**

A testing plan outlines the approach, scope, and timeline for testing Fight game project. Here is a sample testing plan:

* **Test Plan Overview:**

**Project Scope:** The Fight game project aims to develop a user-friendly game to play.

**Test Scope:** The testing scope includes all features and functionalities of the Fight game.

**Test Approach:** The testing approach will include a combination of automated and manual testing.

* **Test Objectives:**

**Ensure Functionality:** Verify that the system meets the specified requirements and functions as expected.

**Ensure Performance:** Verify that the system performs well under various loads and conditions.

**Ensure Security:** Verify that the system is secure and protects sensitive user information.

**Ensure usability:** Verify that the system is user-friendly and provides a good user experience.

* **Test Environment:**

**Hardware:** The testing environment will include a combination of Windows and Linux machines.

**Software:** The testing environment will include the latest versions of Chrome, Firefox, and Edge browsers.

**Network:** The Testing

**(3.2) Testing Methods:**

Testing methods are essential for ensuring the quality and reliability of fight game project. Here are some testing methods that can be used:

* **Black box Testing:**

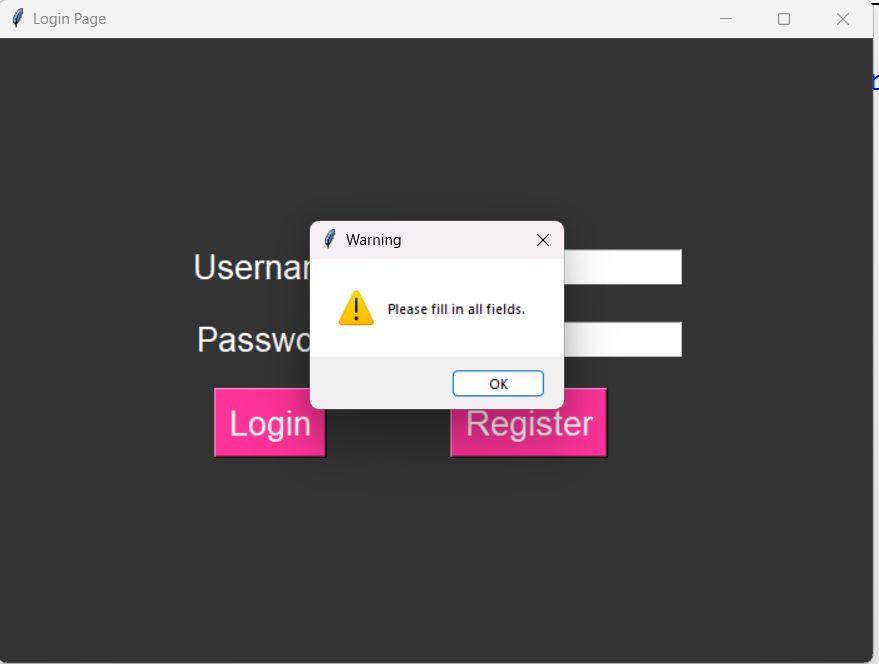
**Equivalence Partitioning:** Divide input data into partitions based on specifications and test each partition at least once.

**Boundary value Analysis:** Test the system with input data at the boundaries of the specified ranges.

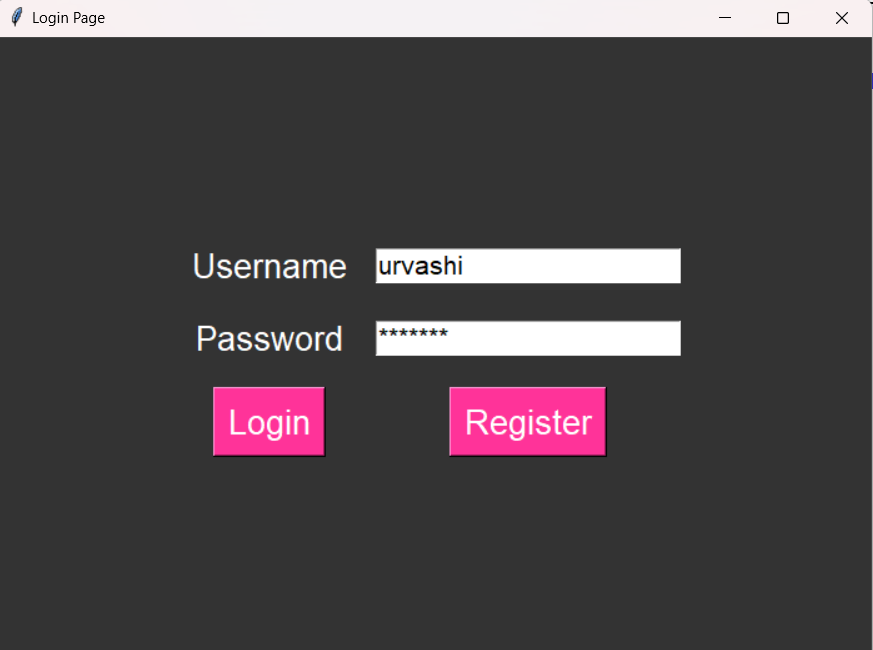
**State Transition Testing:** Test the system's behaviour in different states and transitions

**Login**

Error: invalid entered username or password.

****

Solution: Enter valid password and username.

****

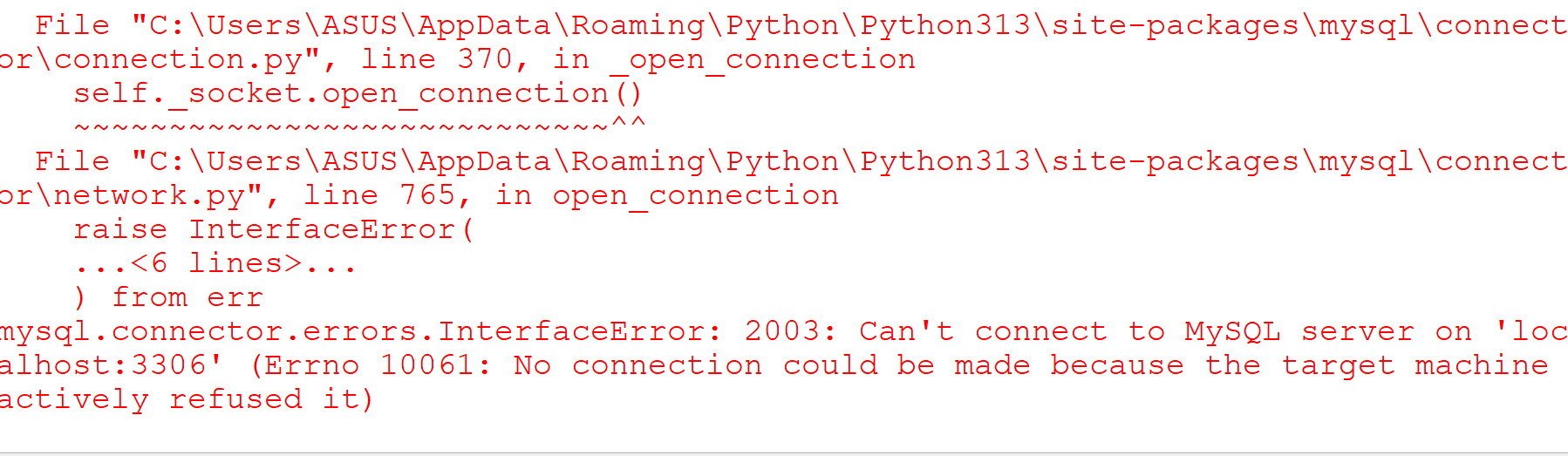
* **White box Testing:**

**Statement coverage:** Test each statement in the code at least once.

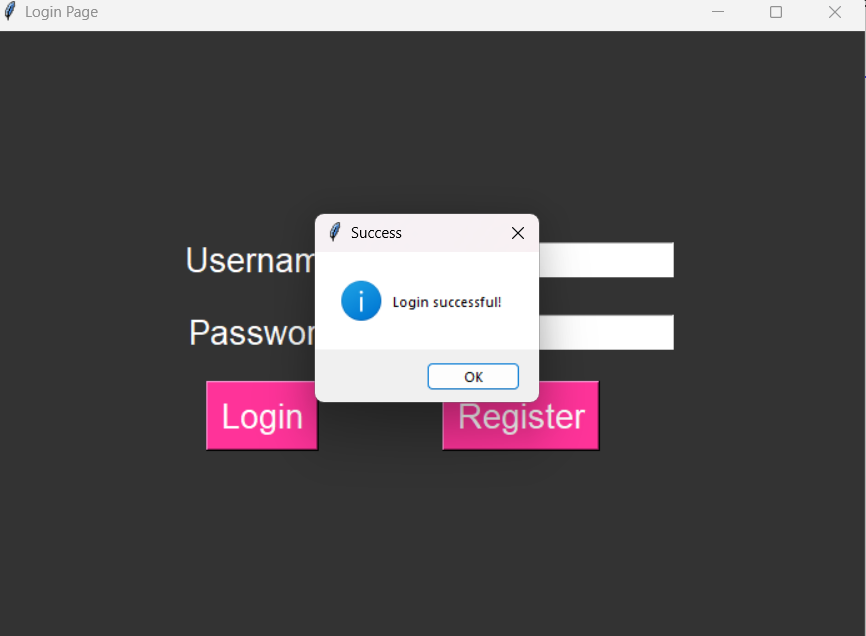
**Decision coverage:** Test each decision point in the code, such as if-else statements.

**Path coverage:** Test each path through the code.

Error: connection string is not match with database connection string.

****

Solution: First match your connection string with database connection string.



* **Gray box Testing:**

**Integration Testing:** Test how different components interact with each other.

**Error Guessing:** Test the system with invalid or unexpected input data.

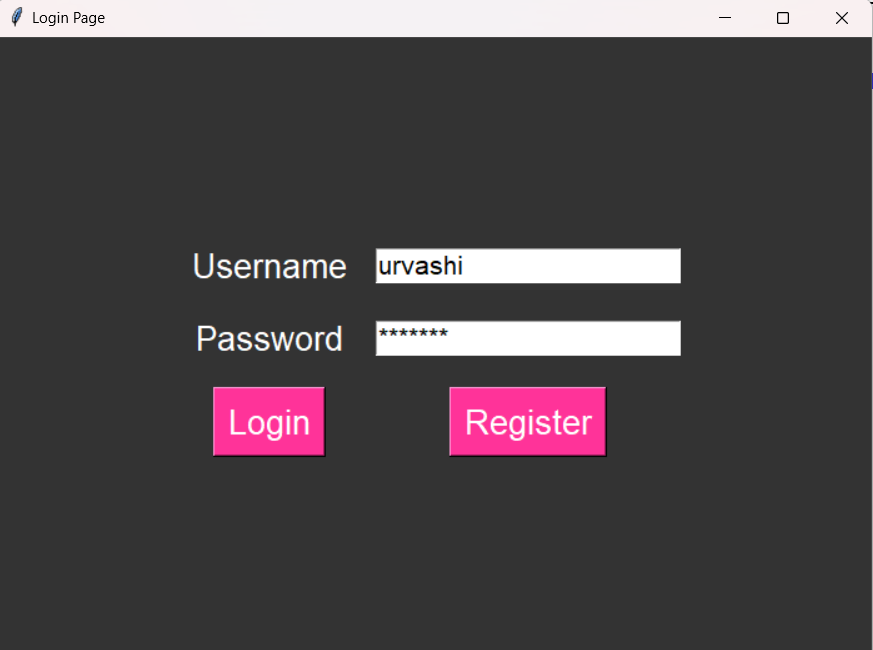
* **Manual Testing:**

**User Acceptance Testing:** Test the system with real users to ensure it meets their acceptance criteria.

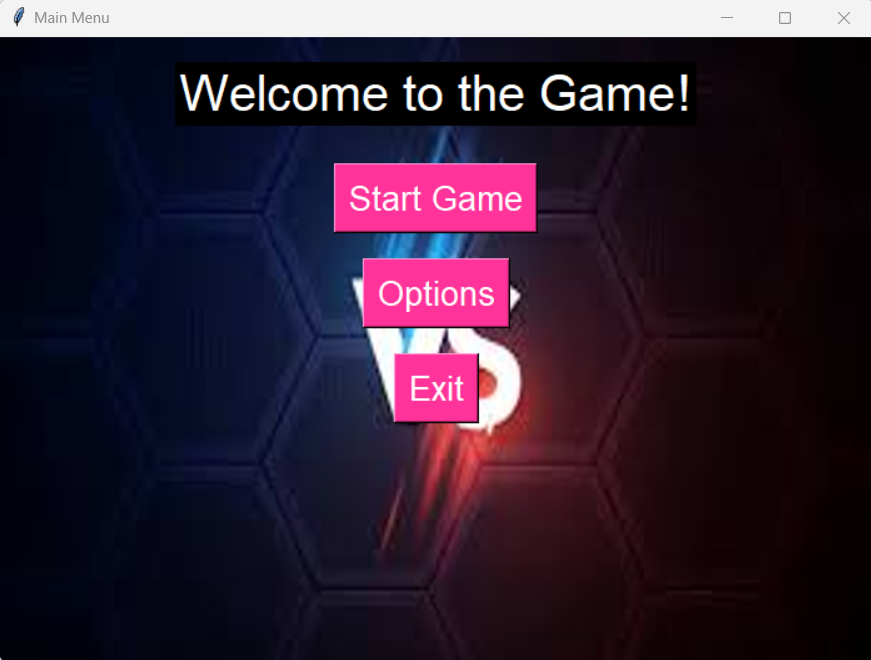
**Exploratory Testing:** Test the system without a predefined set of test cases, focusing on user experience and usability.

* 1. **Screen shots and User manual**

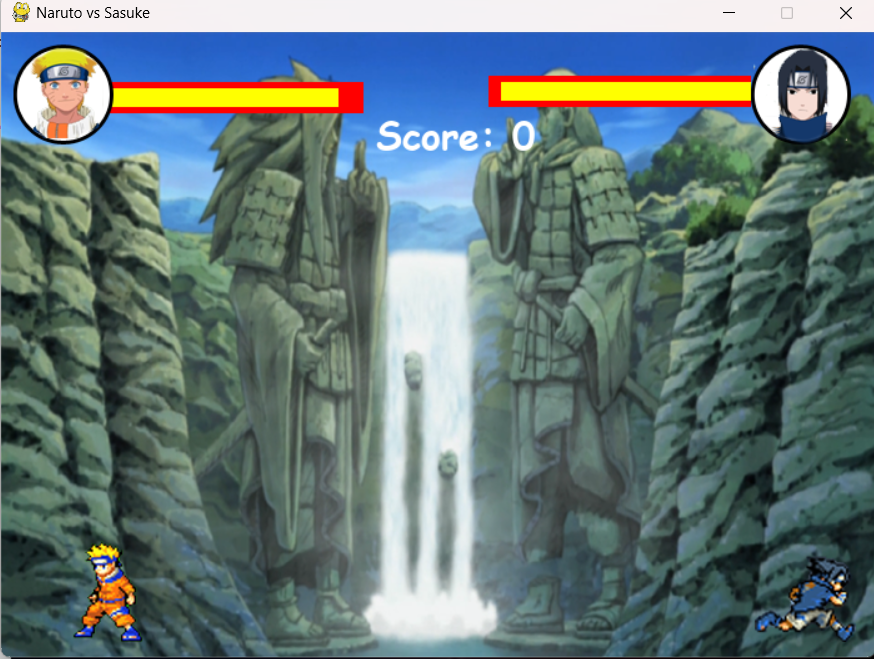
**Login Page**

****

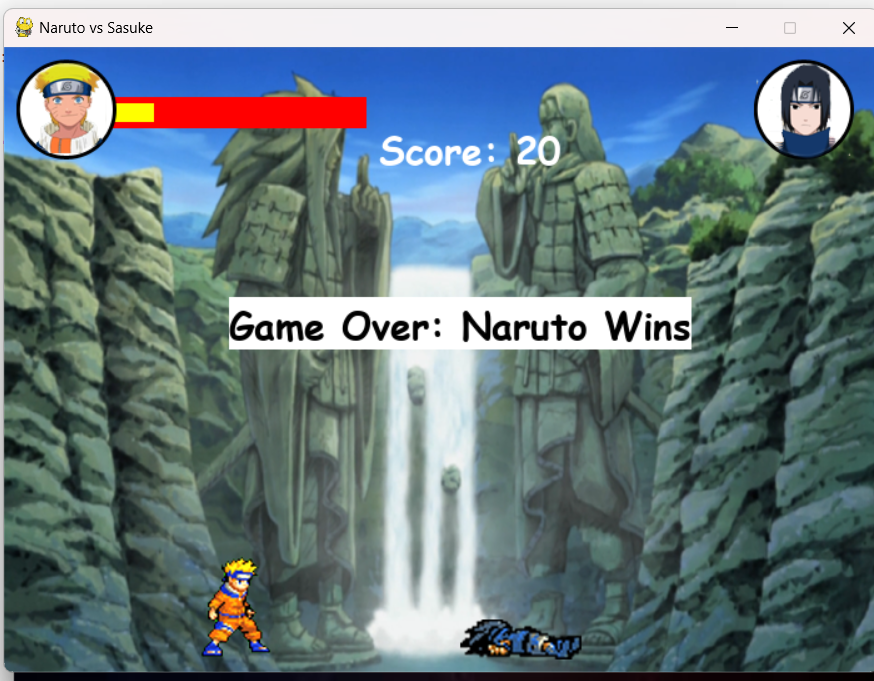
**Main Menu**

****

**Fight game**

****

**Game Over**

****

* 1. **Limitation and Future Enhancement**
* **Limitation**

The main limitations of the proposed Fight game are:

* Python is generally slower than languages like C++ or C#. This can be a limitation for real-time applications like games, especially if they require complex physics or graphics rendering.
* While libraries like Pygame and Panda3D can be used for 2D and 3D graphics, they may not offer the same level of performance or visual fidelity as engines like Unity or Unreal Engine.
* Although there are libraries available for game development in Python, they may not be as comprehensive or well-supported as those in other languages. This can limit the features you can implement.
* **Future Enhancement**
* Use libraries like Cython or PyPy to improve performance. You can also optimize your code by profiling and identifying bottlenecks.
* Integrate with more powerful graphics libraries (like OpenGL) or game engines (like Godot) that support Python scripting to enhance visual quality.
* Integrate a more advanced physics engine to improve collision detection and response, making the fighting mechanics more realistic.
* Implement more sophisticated AI for NPCs using machine learning techniques, which can make the game more engaging and challenging.
* Develop robust networking capabilities to support online multiplayer features, including matchmaking and server management.

1. **Conclusion and Discussion**

* **Conclusion**
* The development of a fighting game in Python presents a unique opportunity to explore game design, programming, and creative storytelling. While Python may not be the first choice for high-performance game development, it offers a range of libraries and frameworks that can facilitate the creation of engaging 2D or simple 3D games.
* The project can serve as an excellent learning experience, allowing developers to understand the fundamentals of game mechanics, graphics rendering, and user interaction.
* **Discussion**
* Developing a fighting game in Python can be a valuable educational experience. It allows developers to apply programming concepts in a practical context, enhancing their problem-solving skills and understanding of game development principles.
* Engaging with the Python game development community can provide support, resources, and inspiration. Collaborating with others can lead to new ideas and improvements, enriching the project.
* The skills and knowledge gained from this project can be applied to more complex game development endeavors or even transitioning to other programming languages and game engines. Developers may choose to expand their projects into larger games or explore different genres.