**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI -590018, KARNATAKA**



**An Internship Report**

**On**

**“Puzzle Game using Python”**

Submitted for fulfilment of the requirement for

**Bachelor of Engineering**

**in**

**Electrical & Electronics Engineering**

Submitted by

|  |  |
| --- | --- |
| **MD REZABUL** | **1GC21EE004** |
| **CHAITHANYA M B** | **1GC21EE001** |
| **KEERTHANA K G** | **1GC21EE003** |

of

**VIII Semester**

**Under the guidance of**

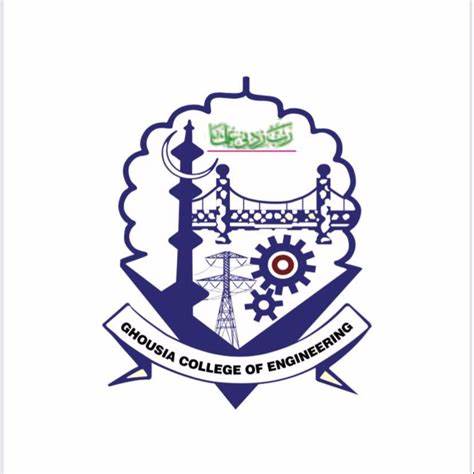
**Asst. Prof. LATHAMANI**

**Department of Electrical & Electronics Engineering,**

**Ghousia College of Engineering**

Ramanagaram, Ramanagara Dist., Karnataka. Pin - 562159

**Ghousia College of Engineering**



Ramanagaram, Ramanagara Dist., Karnataka. Pin - 562159

Approved by AICTE, Affiliated to VTU, Belagavi

**Department of Computer Science and Engineering**

**CERTIFICATE**

Certified that the mini project work entitled **“Puzzle Game using Python”** carried out by **Mr. Md Rezabul, USN: [1GC21EE004],** **Chaithanya M B, USN: [1GC21EE001Keerthana K G, USN: [1GC21EE003],** are bonafide students of Ghousia College of Engineering, in fulfillment of the requirement for Internship project of Bachelor of Engineering in Computer Science and Engineering of the Visveswaraya Technological University, Belgaum during the year 2024-25. It is Certified that all corrections/suggestions indicated for Internal assessment have been incorporated in the report deposited in the department library. The Internship project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Lathamani Mrs. Afroza Firdose Dr. Javid Akhtar

Asst. Professor, EEE Asst. Professor, EEE HOD, EEE, GCE

(Project Guide) (Project Coordinator)

|  |  |  |
| --- | --- | --- |
|  | **Name** | **Signature with date** |
| **Examiner 1** |  |  |
| **Examiner 2** |  |  |
| **Examiner 3** |  |  |

**DECLARATION**

We, Md Rezabul, Chaithanya M B, Keerthana K G, hereby declare that the Internship project work entitled “Puzzle Game using Python” is an authenticated work carried out independently by us at Ghousia College of Engineering. Under the guidance of Lathamani, Assistant Professor, Department of Electrical & Electronics Engineering for the fulfillment of the requirement of BACHELOR OF ENGINEERING in ELECTRICAL & ELECTRONICS ENGINEERING, and this work has not been submitted for the award of any degree/ diploma of an institute/ university.

|  |  |  |
| --- | --- | --- |
| **Name** | **USN** | **Signature** |
| **MD REZABUL** | **1GC21EE004** |  |
| **CHAITHANYA M B** | **1GC21EE001** |  |
| **KEERTHANA K G** | **1GC21EE003** |  |

I

**ACKNOWLEDGEMENT**

The completion of Internship project work brings with a sense of satisfaction, but it is never complete without thanking the person responsible for its successful completion.

We express our special in-depth, heartfelt, sincere gratitude to **Lathamani**, **Assistant Professor**, **Department of Electrical & Electronics Engineering, Ghousia College of Engineering**, for her constant support and valuable guidance for the completion of the project work.

We express our heartfelt sincere gratitude to our beloved **HOD**, **Dr. Javid Akhtar, Department of Electrical & Electronics Engineering, Ghousia College of Engineering**, **Ramanagara** for his valuable suggestions and support.

We express our deep sense of sincere gratitude to **Dr. N S Kumar, Ghousia College of Engineering**, **Ramanagara,** for providing us an opportunity to do the Project.

Finally, we would like to thank to **All the Teaching and Non-Teaching staff** of the **Department of Computer Science and Engineering**, **Ghousia College of Engineering**, **Ramanagara**, for their support.

II

**ABSTRACT**

This project was designed to provide students with a practical, engaging way to enhance their problem-solving skills and deepen their understanding of Python programming. By building a sliding tile puzzle game, students learn important concepts such as graphical user interface (GUI) design, object-oriented programming, and image processing. The game allows users to import custom images and includes three difficulty levels—Easy, Medium, and Hard—offering a tailored challenge for different skill levels. Not only does the project help develop logical thinking and creativity, but it also introduces students to the core mechanics of game development, such as handling user input, animations, and sound effects. Furthermore, by packaging the game for distribution, students gain valuable insight into software deployment, making this project a well-rounded educational experience that blends fun with learning.

III

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Chapter** | **Page No.** |
| 1 | Declaration | I |
| 2 | Acknowledgement | II |
| 3 | Abstract | III |
| 4 | Contents | IV |
| 5 | List of Figures | V |
| 6 | Introduction | 1 |
| 7 | Scope of the Project | 2 |
| 8 | Literature Review | 3 |
| 9 | Methodology | 4 |
| 10 | Details of the Project | 6 |
| 11 | Conclusion | 11 |

IV

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| Figure No. | Description | Page No. |
| 4.1 | The purposed (or expected) flow-line of the project | 4 |
| 5.1 | The Image-importing interface  to import the image for the puzzle | 9 |
| 5.2 | Difficulty selection interface to select difficulty level for the game | 9 |
| 5.3 | Shuffled tiles interface showing the image-tiles in a scrambled way | 10 |
| 5.4 | Puzzle solved interface showing winning message | 10 |
| 5.5 | Endgame interface providing option to play again or quit | 11 |

V

1. **INTRODUCTION**

The Puzzle Game project is an interactive and engaging application designed to emulate the classic sliding tile puzzle, offering players a fun and mentally stimulating experience. Developed using Python and the Pygame library, the game challenges players to rearrange scrambled tiles to recreate an image, thereby testing their logical thinking, problem-solving abilities, and spatial awareness. The game begins with a level selection menu, where players can choose from beginner (3x3), intermediate (4x4), or advanced (5x5) difficulty levels. Players can also personalize their gameplay by loading custom images, ensuring a fresh and unique experience with every session.

To enhance user experience, the game includes helpful features such as numbered tiles for guidance, a move counter to track progress, and a miniature preview of the original image for easy reference. The addition of sound effects for actions like tile movement and puzzle completion adds an interactive auditory element, while a celebratory message marks the completion of the puzzle, providing players with a sense of accomplishment.

This project highlights the practical application of core programming concepts such as event handling, graphical rendering, and user interface design. By incorporating creativity with functionality, it demonstrates how coding can be used to create applications that are both educational and entertaining. The Puzzle Game serves as an excellent example of how programming can be utilized to build interactive and meaningful software solutions.

In addition, this project provides a platform for exploring real-world problem-solving and critical thinking. It showcases the potential of Python as a versatile language for developing graphical applications and highlights how programming can merge logic and creativity to deliver an engaging user experience. Whether for players seeking a challenge or developers aiming to hone their skills, the Puzzle Game represents a blend of innovation and technical proficiency.

1. **SCOPE OF THE PROJECT**

The Puzzle Game project has a broad scope that extends beyond being a simple sliding tile game. Its primary focus is to provide an interactive platform for players to test and enhance their logical reasoning and problem-solving abilities. The game is designed to cater to users of varying skill levels, offering three distinct difficulty modes: beginner (3x3 grid), intermediate (4x4 grid), and advanced (5x5 grid). This ensures accessibility for novices while still challenging experienced players. Additionally, the ability to load custom images allows for personalized gameplay, making the experience unique and versatile for every user.

From a technical perspective, the project demonstrates the integration of various programming concepts, such as graphical user interfaces (GUIs), event-driven programming, and multimedia handling. It highlights the potential of Python and its libraries, such as Pygame, for developing visually appealing and functionally rich applications. The inclusion of features like a move counter, original image preview, and celebratory completion message showcases attention to detail and enhances the user experience.

The project also has potential educational applications. It can serve as a learning tool for students interested in understanding programming logic, game development, and user-centric design. Moreover, it offers opportunities for further enhancement, such as incorporating leaderboard systems, multiplayer modes, or additional puzzle mechanics. By bridging the gap between entertainment and education, the Puzzle Game demonstrates how software projects can be both enjoyable and intellectually rewarding, making it suitable for academic, personal, and recreational purposes.

1. **LITERATURE REVIEW**

The development of the Puzzle Game is inspired by a rich history of sliding tile puzzles and the broader field of game design. Sliding tile puzzles, such as the classic "15 Puzzle," have been a staple in cognitive problem-solving games since the late 19th century. They are well-documented in educational and recreational contexts for their ability to challenge spatial reasoning and logical thinking. Research in cognitive psychology highlights the efficacy of puzzles in improving memory, pattern recognition, and mental agility, which formed the foundation for selecting this project theme.

In the field of software development, graphical games have been extensively studied as a means of introducing programming concepts such as graphical user interfaces (GUIs), event-driven programming, and user experience design. Various programming libraries, such as Pygame, have been used in prior works to create interactive, multimedia-based applications. These studies emphasize the importance of user engagement through intuitive controls, visually appealing interfaces, and progressive difficulty levels—all of which have been incorporated into the design of this Puzzle Game.

Furthermore, research on gamification highlights the impact of incorporating rewards, feedback, and personalized experiences in enhancing user motivation and satisfaction. This insight influenced the addition of features such as a move counter, celebratory win messages, and the ability to load custom images, aligning with best practices in game design.

Existing literature on educational games also underlines the potential for puzzles to serve as learning tools. By developing a game that integrates cognitive challenges with customizable options, this project bridges the gap between entertainment and educational value. This literature review provided a theoretical framework for the design and implementation of the Puzzle Game, ensuring it is both engaging and intellectually stimulating.

1. **METHODOLOGY**

The methodology for this Puzzle Game project encompasses a systematic approach to planning, designing, implementing, and evaluating the application.

**Requirements analysis**

**Minimum Requirements:**

* Python 3.7 (or higher) : Core programming language used.
* Processor : Any modern processor (Intel i3 or AMD Ryzen 3) of 3.0GHz.
* RAM : 2GB DDR3.
* Storage : 500MB(approx.)
* OS : Windows 7, Linux or MacOS.

**Libraries:**

The main Python-libraries used in this project are:

1. Pygame:

Used for game development, GUI handling, rendering images, playing sound effects, and capturing user input.

1. Tkinter (filedialog):

Specifically, filedialog from the Tkinter library is used to allow users to browse and select custom images from their local storage.

1. Random:

A built-in Python library used for shuffling the puzzle tiles, ensuring randomness in tile placement.

1. OS (Optional):

A built-in Python library that can be used for interacting with the file system when loading images or dealing with file paths.

1. PyInstaller (for distribution):

Not used in the gameplay code but helpful when packaging the script into an executable format for sharing or distribution.

**Framework**

The Python-Based Puzzle Game follows a systematic approach to game development using the Pygame library, which provides the core framework for creating game loops, handling user input, and rendering the graphical interface. The project is structured in a modular way, using object-oriented programming principles to ensure that components such as the puzzle grid, image handling, and user interactions are clearly separated.

**Proposed flow line**

Fig 4.1 : The expected flow-line of the project

**Objectives**

The primary goal of this project is to develop a Python-based puzzle game that challenges players' logical thinking and enhances their ability to solve problems. The main motives of this game is to provide:

* A personalized puzzle experience by allowing users to import their own images.
* A variety of difficulty levels to cater to both casual players and experienced puzzle solvers.
* A platform for students to learn and apply Python programming concepts in a fun and interactive way.

1. **DETAILS OF THE PROJECT**

**Code Description:**

The whole code can be broken into several logical blocks. Each block has been explained in detail below:

1. **Initialization and Setup**

Purpose: Sets up the environment and resources needed for the game.

Code Details:

Pygame Initialization: `pygame.init()` initializes all Pygame modules.

Window Settings: `screen = pygame.display.set\_mode((window\_width, window\_height))` creates the main game window, and `pygame.display.set\_caption()` sets its title.

Colors and Fonts: Defines color constants (e.g., `WHITE`, `BLACK`) and initializes fonts for rendering text.

Sound Effects: Loads sound effects (`move.wav`, `win.wav`). If the files are missing, the exception handling ensures the game doesn't crash.

1. **Helper Functions**

Purpose: Modularizes key functionalities for reusability and readability.

a. `display\_message()`

Renders a message at the center of the screen with optional vertical offsets for positioning. Used for displaying messages like "You Win!" or instructions.

b. `load\_image()`

Opens a file dialog for the user to select an image, then loads the image into Pygame. This ensures user interactivity at the start.

c. `split\_image()`

Splits the loaded image into a grid of tiles based on the selected difficulty level (rows and columns). Each tile is a subsurface of the original image.

d. `draw\_numbers\_on\_tiles()`

Adds visible numbers to each tile (except the empty one) for easier gameplay. The numbers are white with a thin black outline for readability.

e. `shuffle\_tiles()`

Shuffles the tiles by simulating 1000 random movements of the empty tile. This ensures the puzzle starts in a randomized but solvable state.

f. `move\_tile\_with\_mouse()`

Handles tile movement based on mouse input. When a tile adjacent to the empty space is clicked, the two tiles swap places. Plays a sound if available.

g. `is\_solved()`

Checks if the current tile arrangement matches the solved configuration. Used to determine when the game ends.

h.`level\_selection()`

Displays buttons for difficulty level selection (3x3, 4x4, or 5x5 grids). Captures the user's choice and returns the corresponding row and column values.

1. **Main Game Loop**

Purpose: Implements the core game logic, user interactions, and screen updates.

a. Image and Level Setup

The user selects an image using `load\_image()`, and the difficulty level is chosen via `level\_selection()`.

The image is scaled to fit the game window and split into tiles using `split\_image()` and `draw\_numbers\_on\_tiles()`.

b. Game State Initialization

The grid is initialized with sequential numbers representing tile positions. The last position is left empty for movement.

`shuffle\_tiles()` is used to randomize the grid while tracking the empty tile position.

c. Game Loop

Continuously listens for events and updates the screen until the puzzle is solved or the user quits.

Tile Drawing: Displays all tiles in their current positions, skipping the empty tile.

- Mouse Input: Captures mouse clicks and moves tiles if valid.

- Win Check: Calls `is\_solved()` to determine if the puzzle is solved and displays a celebratory message if so.

d. Post-Game Menu

After solving the puzzle, a menu is displayed with "Yes" (play again) or "No" (quit) options. This ensures the user can restart without closing the application.

1. **Replay and Exit Handling**

Purpose: Allows the user to replay the game or exit gracefully.

Code Details:

After solving the puzzle, the replay menu lets users choose to reload the game from the image import step or quit the application.

Graceful exit ensures that resources are released and the program closes without errors.

1. **Main Function**

Purpose: Entry point of the application.

Code Details:

Combines all the above functionality.

Loops back to the start (image selection and level choice) if the user chooses to replay.

This modular structure ensures that each function has a single responsibility, making the code easier to read, debug, and extend.

**Outcomes:**

The final output of the code after successful compilation is given below:

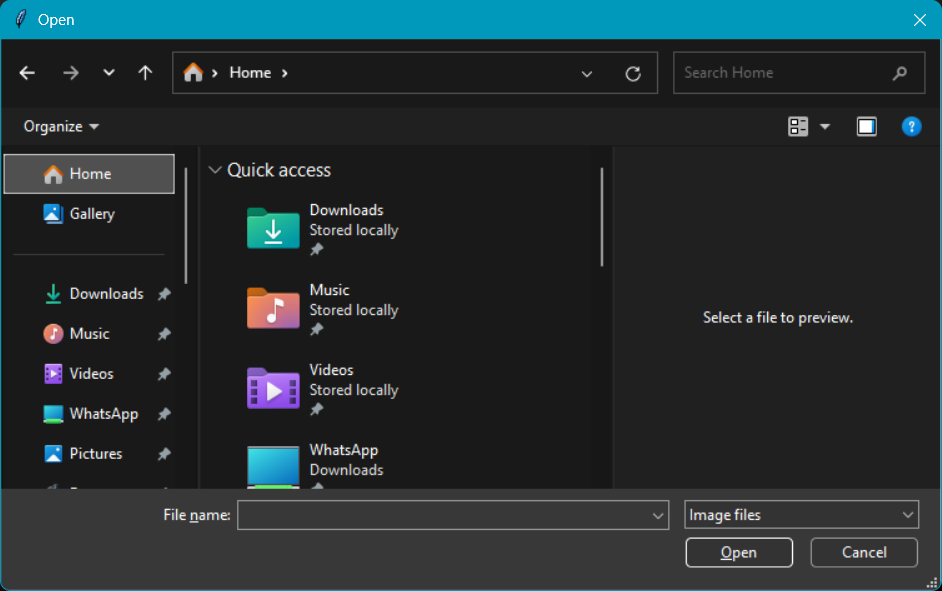
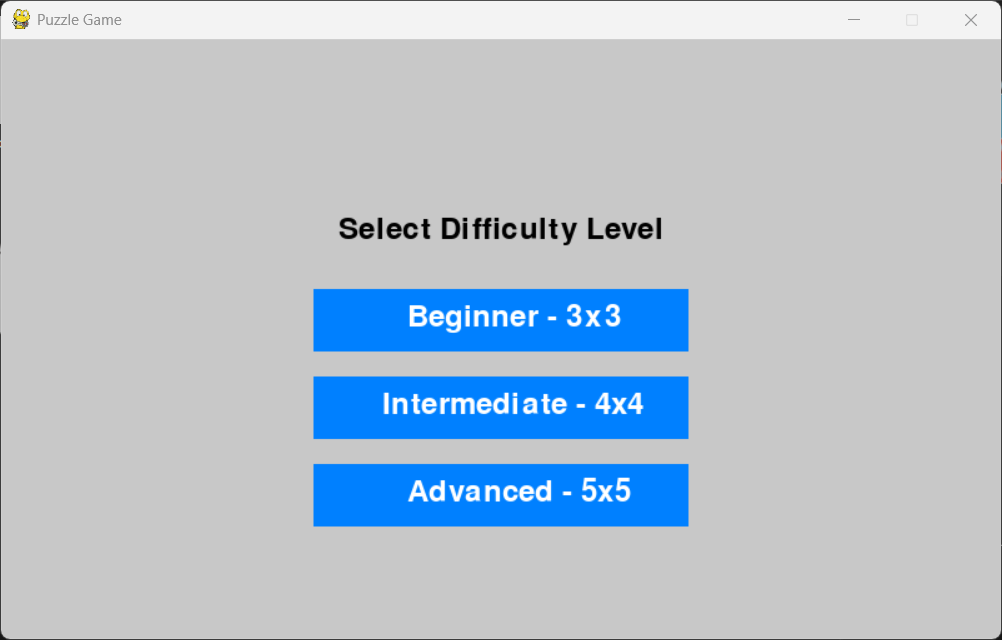


Fig 5.2 : Difficulty level selection interface

Fig 5.1 : Image-importing interface

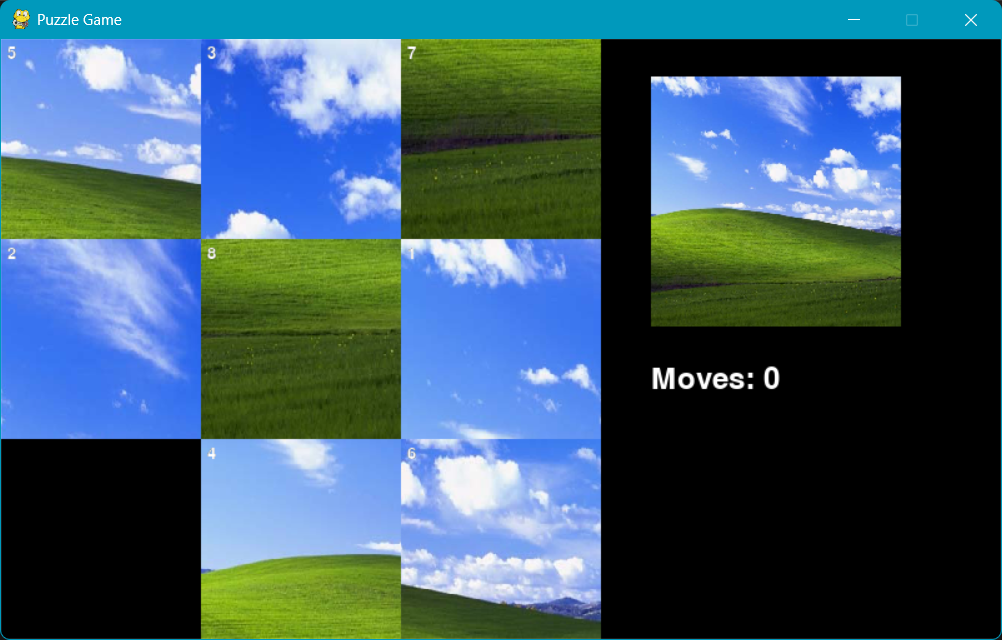


Fig 5.3 : Scrambled tiles at the start of the game

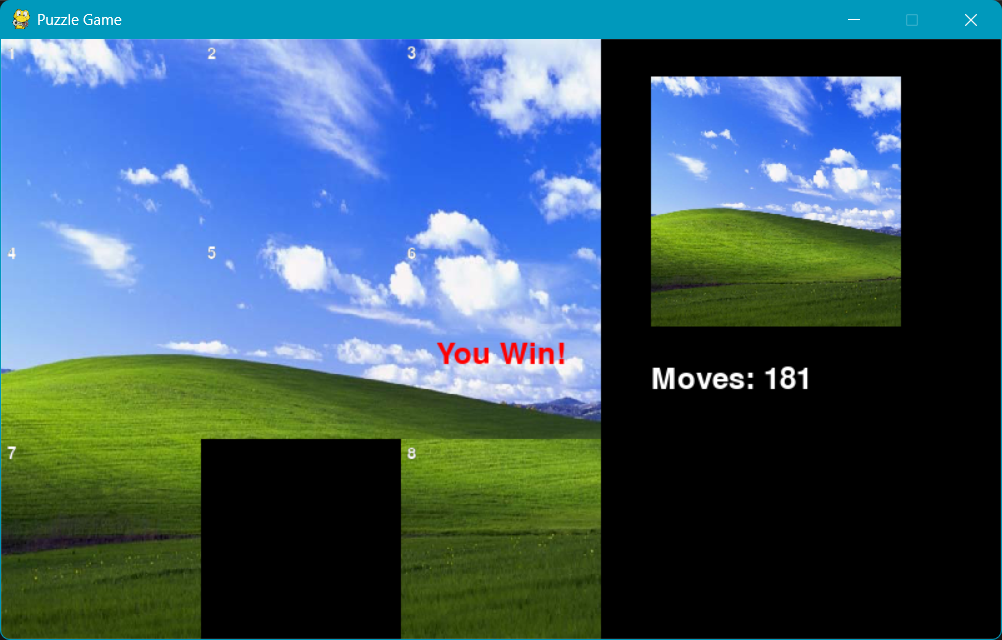


Fig 5.4 : Winning interface after solving the puzzle

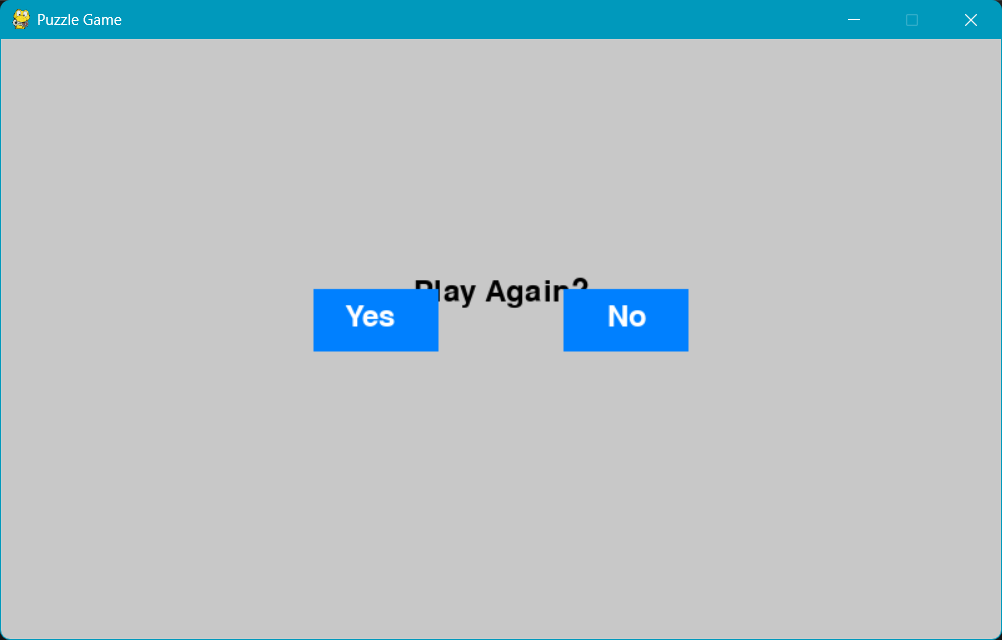


Fig 5.5 : Endgame interface to play again or quit.

1. **CONCLUSION**

The Python-Based Puzzle Game project is an excellent learning opportunity for students interested in game development, GUI design, and problem-solving. It provides a fun and challenging way to apply Python programming skills while offering valuable insights into real-world software development practices. This project also enhances critical thinking and creativity, making it a perfect mini-project for students exploring computer science, programming, or game design.