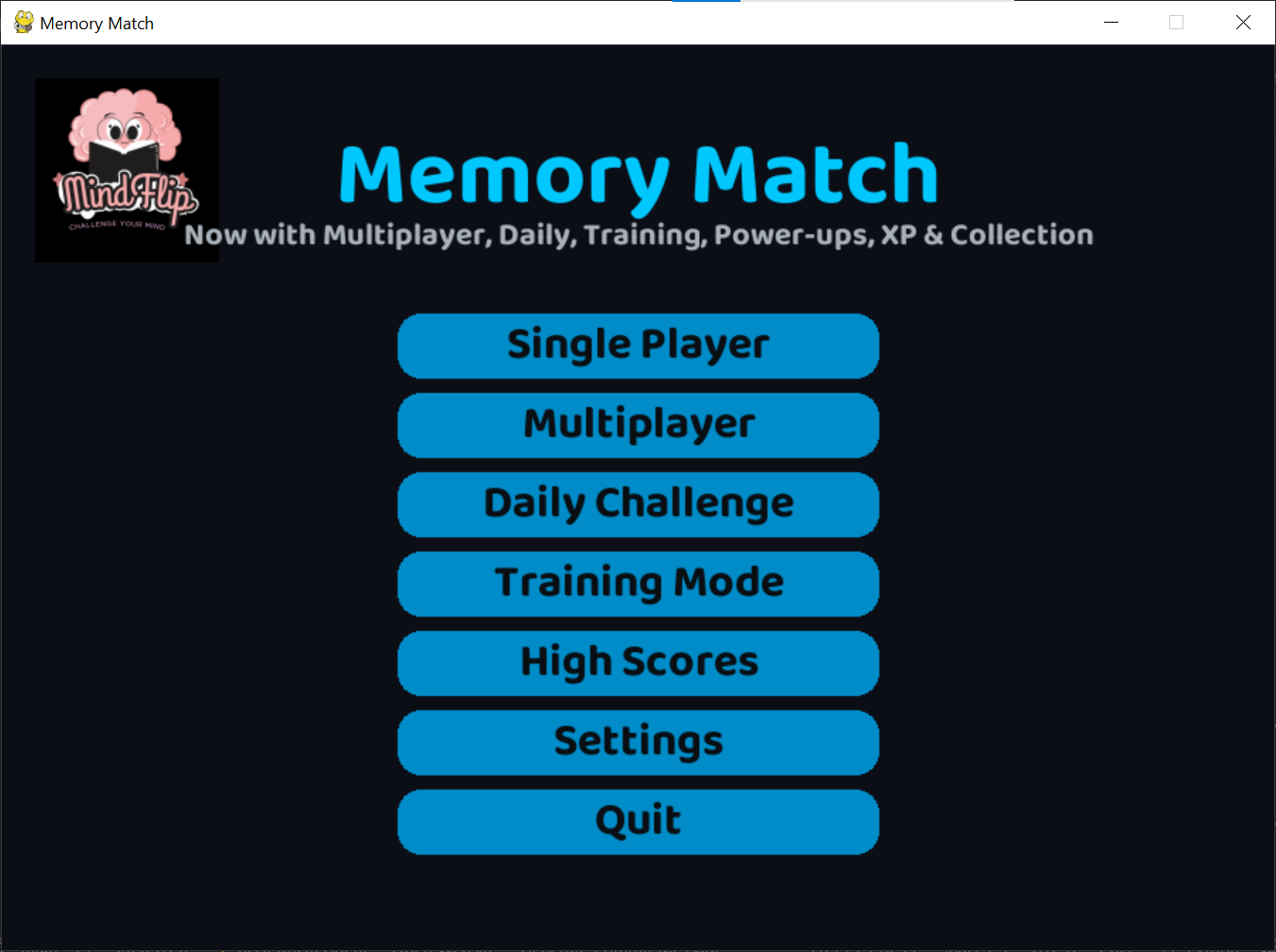
Memory Match Game — MindFlip

Team Members: 5

Course / Department: Information Technology

Date: 8/31/2025



# 1. Introduction

MindFlip — Memory Match is an advanced digital card-matching game developed using Python and Pygame. It is designed to improve memory skills, provide entertainment, and incorporate modern gamification features such as XP, achievements, daily challenges, power-ups, and collections. The game includes multiple modes, multiplayer functionality, and an adaptive layout that scales to different screen sizes.

# 2. Objectives

- Create an engaging and interactive memory match game.

- Implement single-player, multiplayer, and daily challenge modes.

- Track player statistics such as score, moves, time, XP, streaks, and achievements.

- Include a card collection system to unlock and view matched cards.

- Implement power-ups (shuffle, bomb, freeze) for advanced gameplay strategies.

- Provide an intuitive user interface (UI) with scalable graphics.

- Enable settings customization, including music and full-screen mode.

# 3. Team Roles & Responsibilities

|  |  |  |
| --- | --- | --- |
| Member | Role | Contribution |
| M J H A P Madushani | Project Leader & Core Developer | Oversaw project execution, managed game logic, card matching, level progression, and power-ups. |
| W K D Bhagya | UI/UX Designer | Designed main menu, HUD, button interactions, and responsive card grid layout. |
| M T Rathnayake | Graphics & Multimedia Developer | Created card images, logo, backgrounds, and sound effects. |
| M K H K Madushani | Data & Backend Developer | Implemented JSON storage for profiles, high scores, achievements, and collection data. |
| M G J Sinty | QA Tester & Documentation | Tested game modes, reported bugs, optimized performance, and prepared project report and documentation. |

# 4. Technologies & Tools

- Programming Language: Python 3.10+

- Game Library: Pygame

- Data Storage: JSON (for profiles, daily scores, collection, achievements)

- Graphics: PNG, JPG images for cards and logo

- Audio: Background music and sound effects (MP3/WAV)

- IDE: VS Code / PyCharm / Any Python IDE

# 5. Game Features

- Single Player Mode: Play through 32 levels with increasing difficulty.

- Multiplayer Mode: Turn-based gameplay, track each player’s score.

- Daily Challenge: Randomized daily layout with score leaderboard.

- Training Mode: Cards are briefly revealed for practice.

- Power-Ups: Shuffle cards, freeze timer, or remove unmatched cards.

- Collection System: Unlock and view matched cards in a gallery.

- Achievements & XP System: Reward players for completing levels, streaks, or flawless matches.

- Settings: Control music, full-screen mode, and gameplay preferences.

- High Scores: Track best times, moves, and daily leaderboard.

# 6. Game Flow

- Launch Screen: Display logo and game title.

- Main Menu: Access all game modes, settings, collection, and high scores.

- Gameplay: Cards laid out in a dynamic grid, players flip cards to match, unmatched cards flip back, power-ups can be used, track moves, time, XP, and streaks.

- Completion: Display score, XP earned, achievements unlocked, and option to replay.

- Exit: Accessible via ESC key or Quit option in the menu.

# 7. System Design

Architecture Overview: Frontend handles UI, game rendering, animations, and user input. Backend manages game logic, card matching, scoring, XP, achievements, and JSON data storage.

Data Structure: Profile JSON stores XP, level, streaks, achievements, power-ups, and settings. Scores JSON stores high scores. Daily Scores JSON stores daily challenge scores.

Card Management: Each card object contains face\_img, back\_img, flipped, matched, and animation state. Cards are dynamically scaled depending on screen resolution.

# 8. Implementation Details

- Python Pygame used for graphics rendering and event handling.

- Dynamic Grid Computation: Determines rows and columns based on the number of cards.

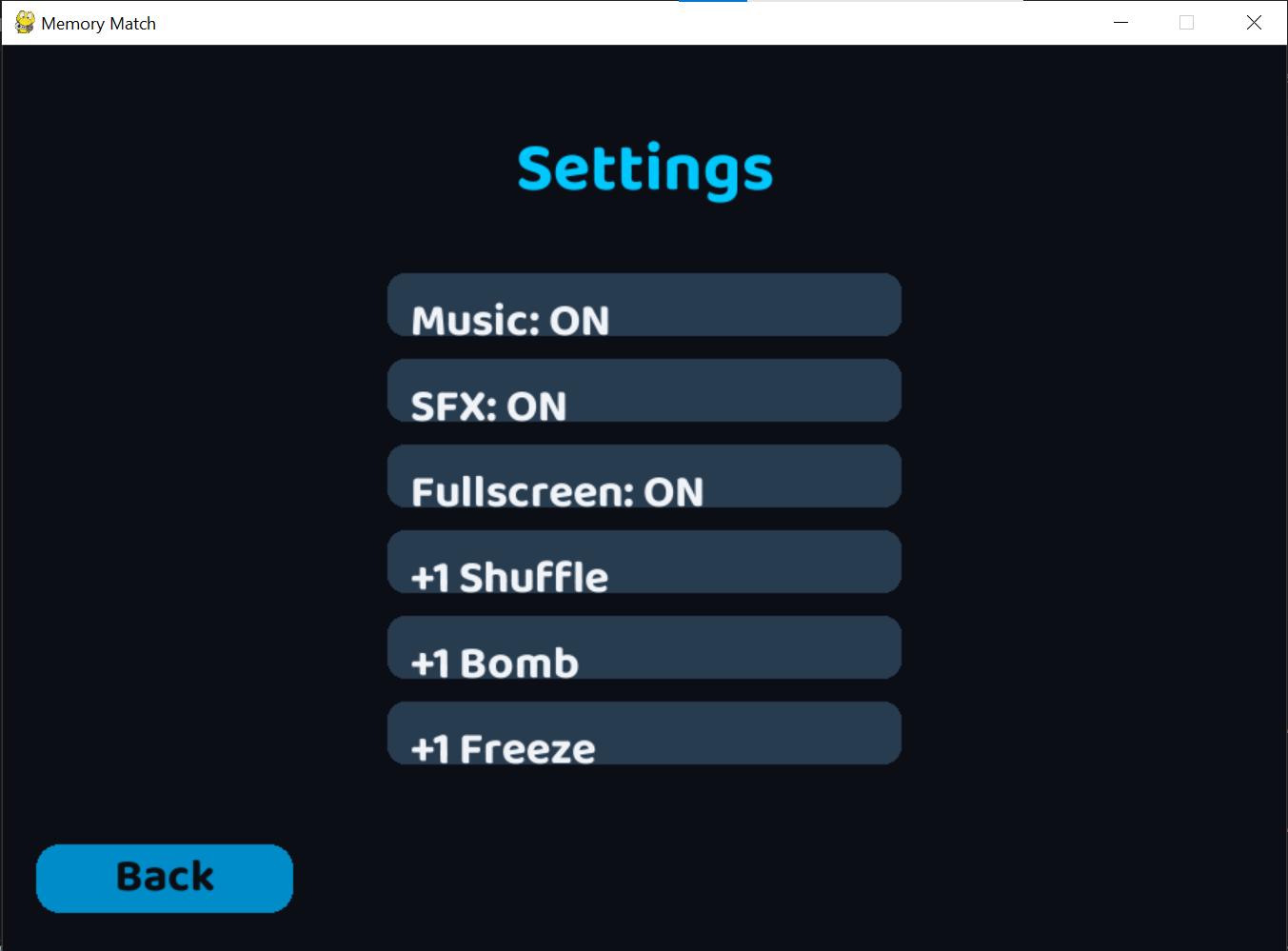
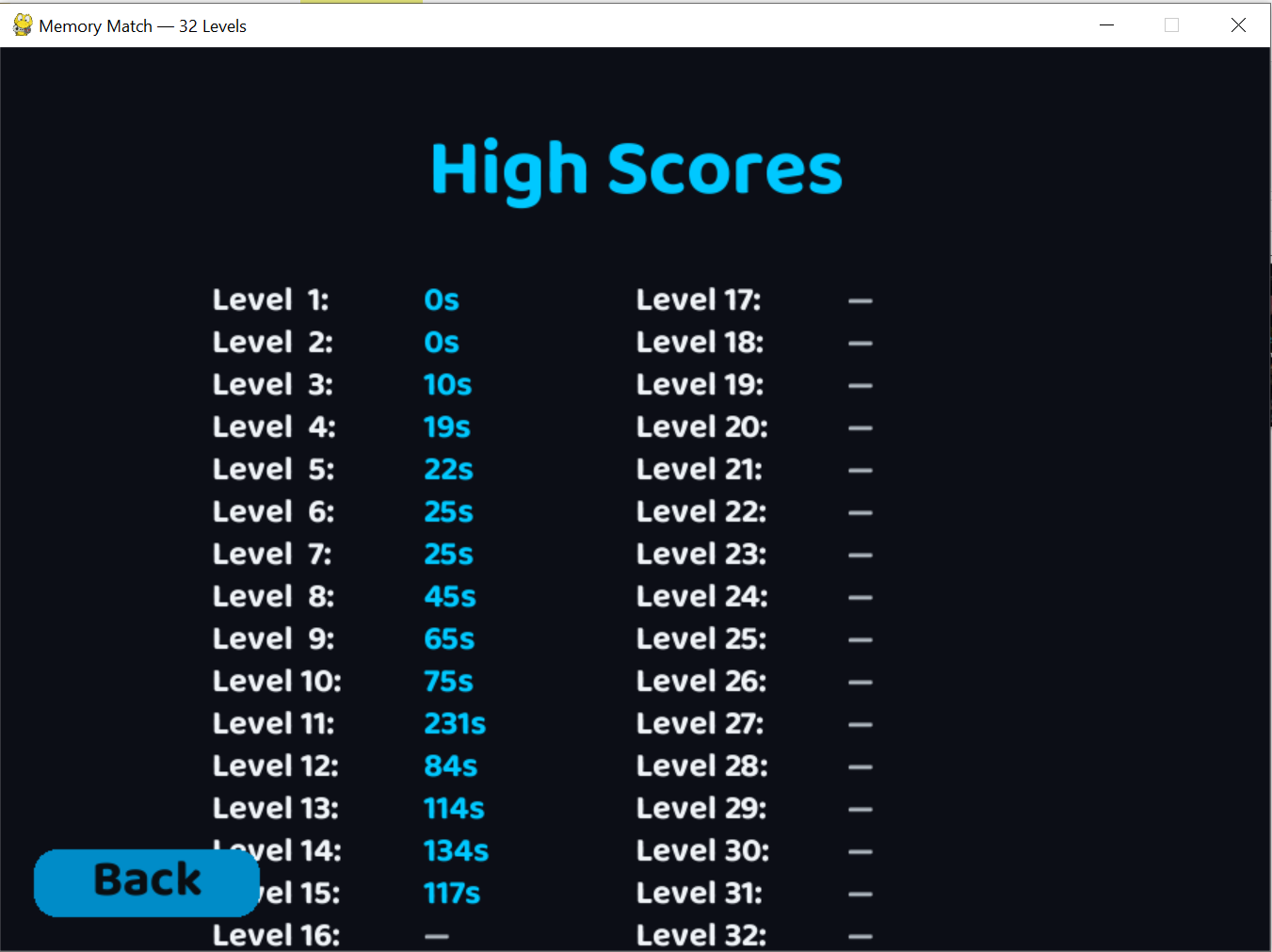
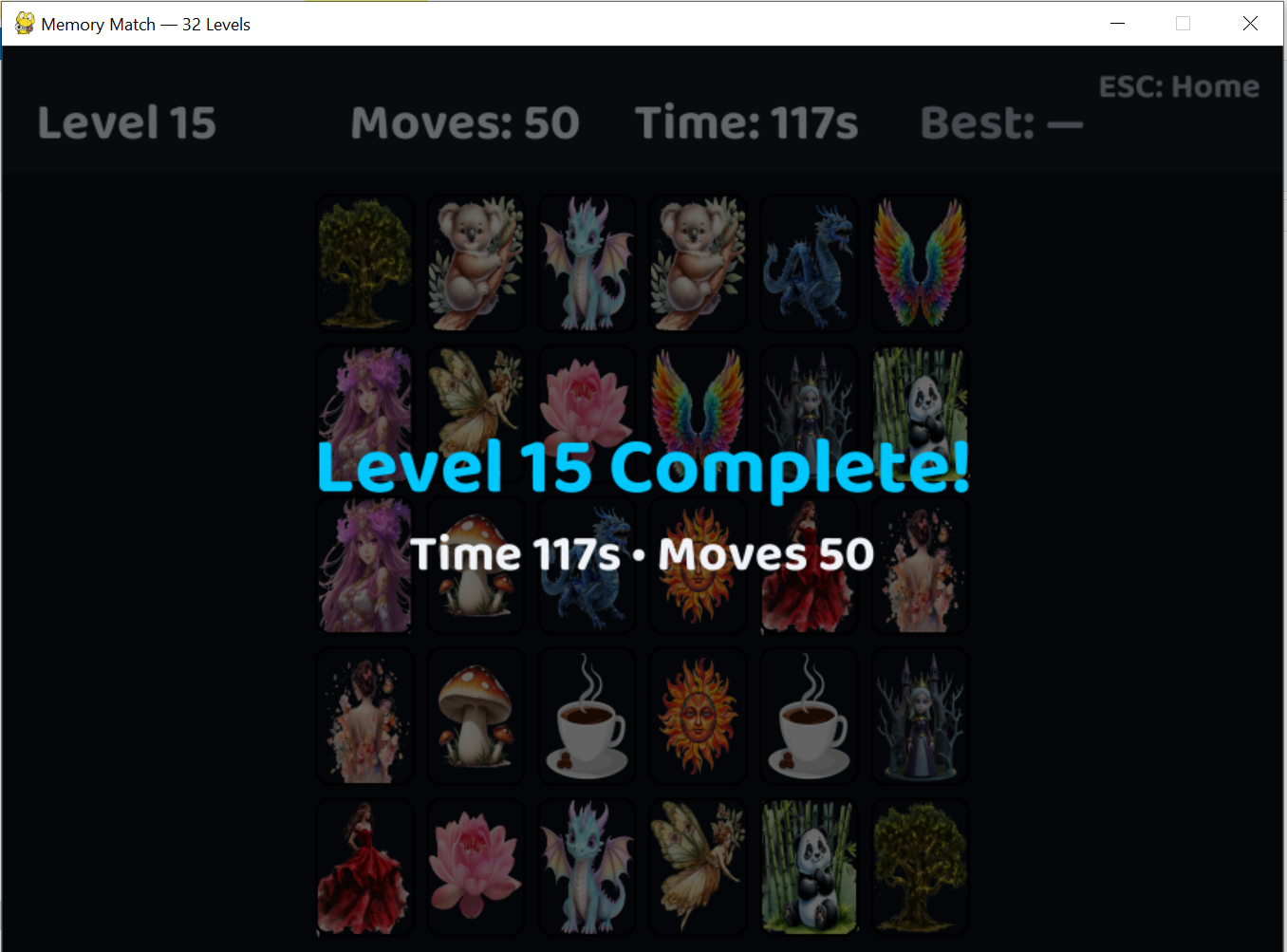
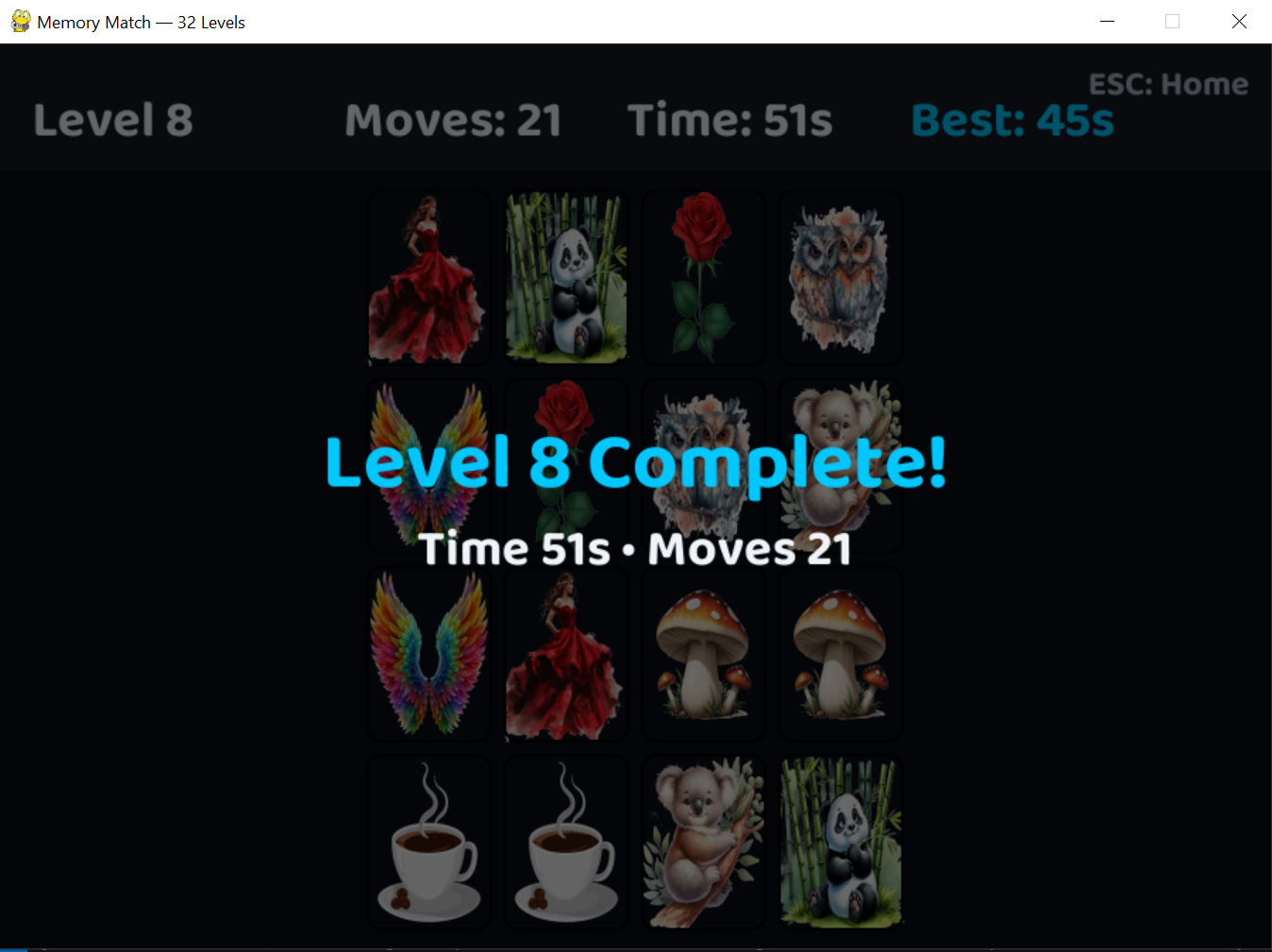
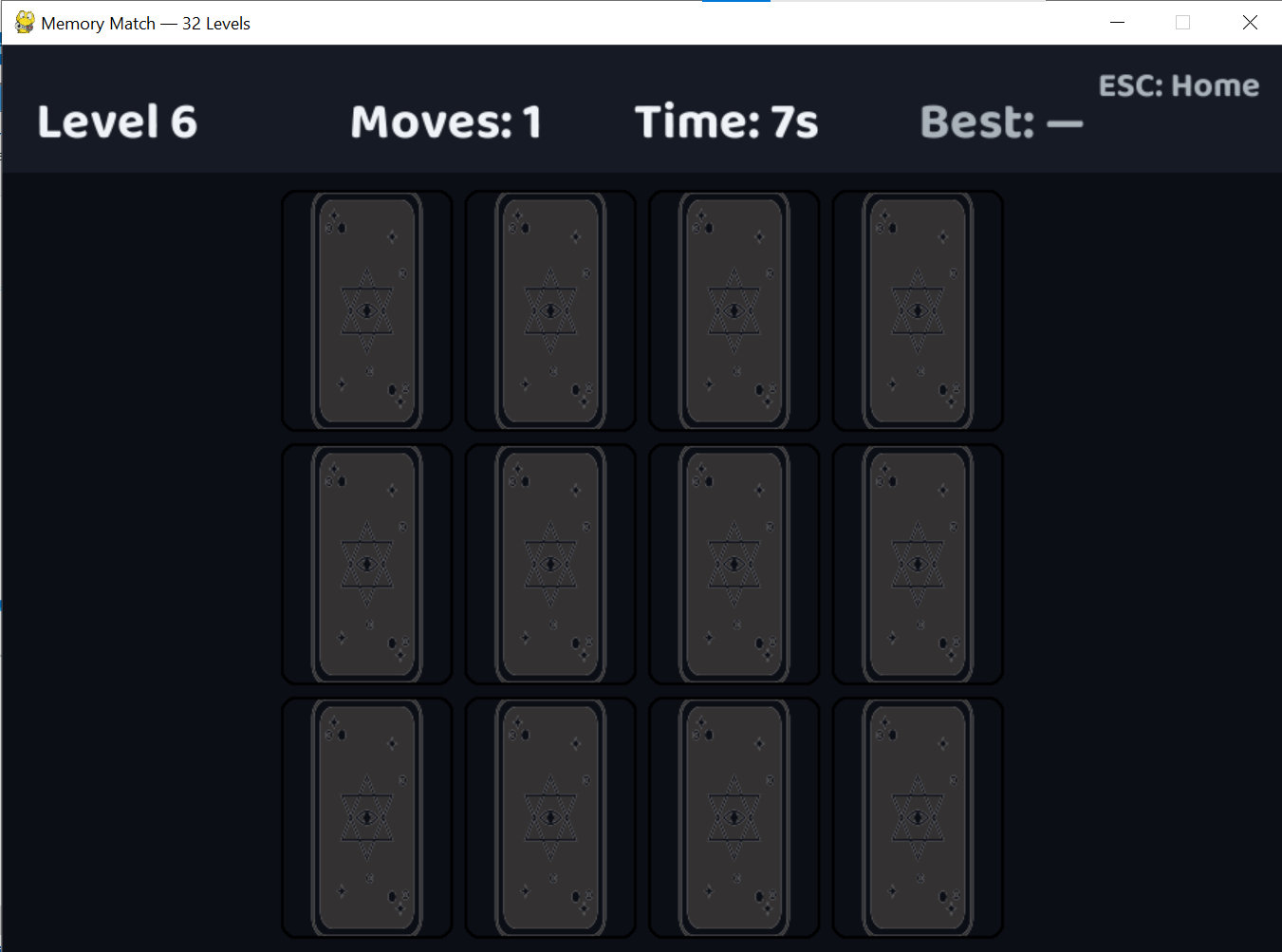
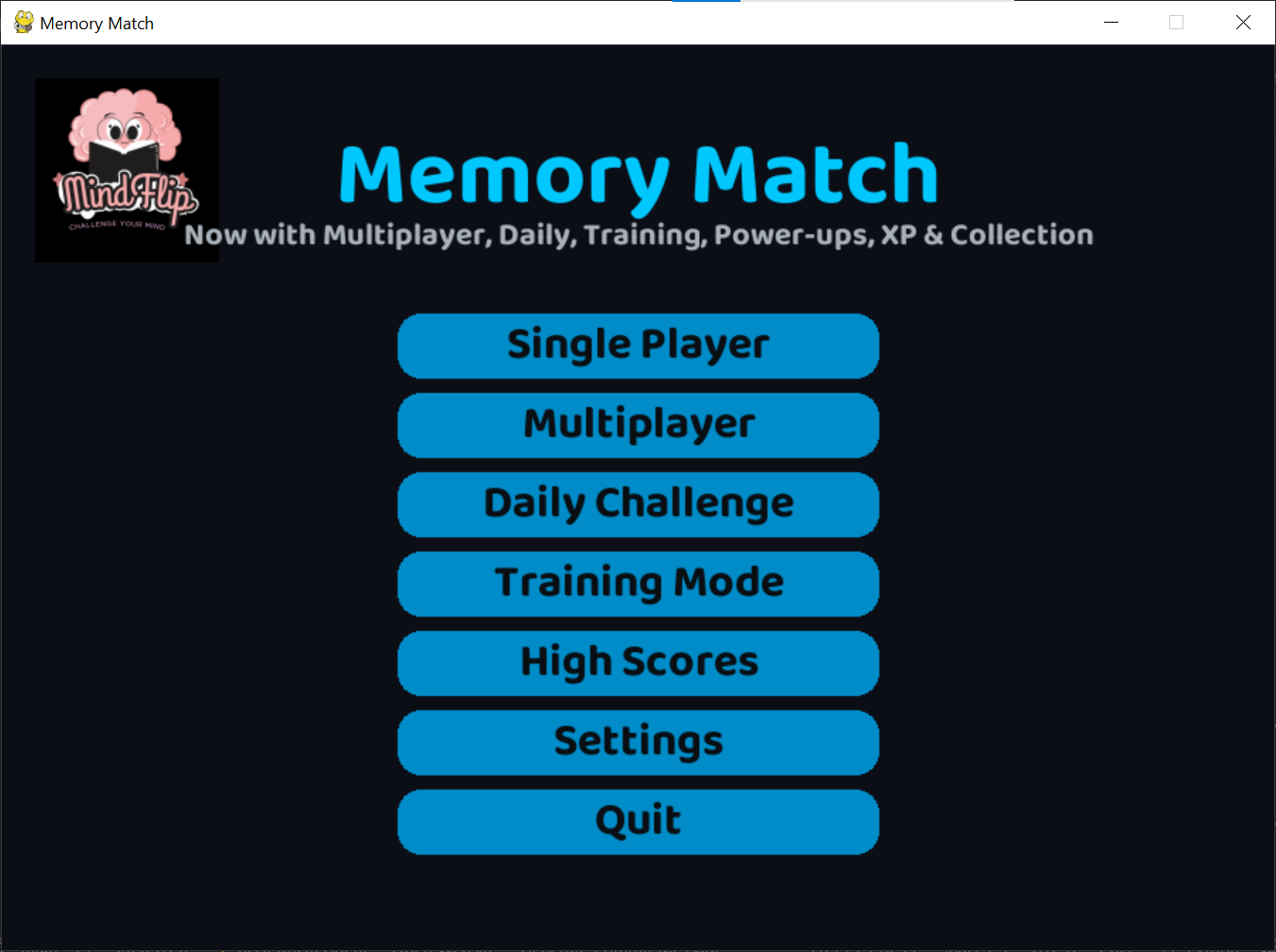
- Scaling Images: Ensures all card images and logo scale properly to window size.

- Safe Asset Loading: Handles missing images or sound files without crashing.

- Animations: Smooth flip animations with bump effect when a match occurs.

# 9. Screenshots / Mockups

Include screenshots after testing the game. Example sections: Main Menu, Gameplay, Achievements & Collection.



# 10. Challenges & Solutions

- Dynamic Card Scaling: Solved using aspect ratio calculations.

- Full-Screen Compatibility: Implemented toggle via Pygame flags.

- Data Persistence: JSON files store profiles and scores for future sessions.

- UI Responsiveness: Button hover and click animations implemented for better UX.

- Multiplayer Sync: Turn-based logic ensures proper score tracking for two players.

# 11. Future Improvements

- Add animated backgrounds and themes.

- Integrate online multiplayer leaderboard.

- Add more power-ups and custom challenges.

- Implement AI opponents for single-player mode.

- Add mobile compatibility for Android/iOS via Kivy or PyGame Subset for Android.

# 12. Conclusion

MindFlip — Memory Match is a fully functional, engaging game that demonstrates teamwork, Python programming skills, game design principles, and user interface development. It offers multiple game modes, achievements, and a collection system, creating a rich and replayable gaming experience.

**Project Repository:**  
The complete source code, assets, and instructions for running the Memory Match game can be found on GitHub:  
<https://github.com/Piyumi2025/MemoryMatchGame>

# 13. References

- Python Official Documentation — https://docs.python.org/3/

- Pygame Documentation — https://www.pygame.org/docs/

- JSON Data Handling in Python — https://docs.python.org/3/library/json.html