The British College

Documentation

Submitted by

Name: Sujit Kumar Thakur

Student ID: 23085143

Submitted to: Aashish Acharya

Kathmandu, Nepal

April 21, 2025

Documentation:

Match-3 Game

Introduction:

This project is a Match-3 puzzle game built using C++ and the SFML (Simple and Fast Multimedia Library). It features colorful gems, a scoring system, level progression, and game state saving/loading. The game is designed with simple object-oriented programming concepts to help learners understand 2D game development fundamentals.

From a development perspective, this project applies object-oriented programming (OOP) principles to organize code efficiently using classes like Piece and GameObject. It also serves as a beginner-friendly example of 2D game logic, user input handling, grid-based collision detection, inheritance and graphics rendering, making it an excellent learning tool for students and aspiring game developers.

Tools & Technologies Used:

- Programming Language: C++
- Graphics Library: SFML (Simple and Fast Multimedia Library)
- IDE/Compiler: Clion 2025.1
- Assets: background.png, gems.png, font.ttf
- File Handling: Used for saving and loading game state.

Game Logic Overview

1. Grid and Tiles:

- The game grid is 8x8, holding colored gem pieces.
- Each piece is an object with properties like position, type (kind), match flag, and transparency (alpha).

2. Game Mechanics:

- Players can click two adjacent gems to swap them.
- If the swap forms a horizontal or vertical line of 3 or more matching gems, those gems disappear.
- New gems fall from above to replace removed ones.

3. Scoring and Levels:

- Matching 3 gems: +10 points
- Matching 4 gems: +20 points
- Matching 5+ gems: +30 points
- When the score reaches 100 points, the game levels up to Level 2.

4. Save/Load Feature:

- Press K to save the game to save.txt.
- Press L to load the game from the file.

5. Controls:

- Mouse: Select and swap gems.
- W/A/S/D keys: Move the red highlight cursor.
- K key: Save the game.
- L key: Load the saved game.

Graphics and Rendering:

- The background and gem sprites are loaded from image files.
- The red highlight box shows the currently selected tile.
- Gems fade out when matched and animate as they fall into place.

File Handling:

- A simple text file save.txt stores the current score, move count, level, and grid state.
- File I/O is handled using ofstream and ifstream for saving and loading respectively.

Key Features Summary:

- 2D Match-3 gameplay using SFML.
- Object-oriented design (with Piece and GameObject classes).
- Colorful tile animations and fade effects.
- Keyboard and mouse controls.
- Level-up and score system.
- Save and load feature with file handling.

Uses:

- File I/O Operations
- 2D array manipulation
- User input handling
- Game state management
- Basic collision detection

Conclusion:

This Match-3 game project showcases how C++ and SFML can be used to build an engaging 2D puzzle game with intuitive gameplay and smooth visuals. Using object-oriented programming, sprite handling, and simple game mechanics, it provides a solid foundation for understanding core game development concepts. The inclusion of scoring, level progression, keyboard and mouse input, and file handling for saving/loading enhances the experience and encourages players to improve their strategy and performance. This project serves as a great starting point for beginners and can be expanded further with new features like combos, special tiles, sound effects, and more complex levels to create a richer and more dynamic gaming experience.