

Project Proposal

Project Title

C++ Tetris Game with raylib

Group Members

- ALISHBA AMIR – 24K-0572
- FABIHA SIDDIQUI – 24K-1022
- JITESH KUMAR– 24K-0871

Submission Date

April 24, 2025

1. Executive Summary

Overview:

This project involves developing a classic Tetris game using C++ and the raylib library. The primary goal was to apply Object-Oriented Programming (OOP) principles to create a modular and maintainable codebase. The game is designed to run on multiple platforms, including macOS, and Linux.

Key Findings:

The project successfully demonstrates the use of OOP concepts in game development. By leveraging raylib, we achieved efficient graphics rendering, input handling, and audio integration. The final product is a fully functional Tetris game that can serve as a foundation for further enhancements.

2. Introduction

Background:

Tetris is a timeless puzzle game that offers an excellent opportunity to explore game development fundamentals. Implementing Tetris using C++ and raylib allows for a hands-on application of OOP concepts, such as classes, inheritance, and encapsulation.

Project Objectives:

- Develop a cross-platform Tetris game using C++ and raylib.
- Apply OOP principles to structure the game's architecture.
- Implement core Tetris functionalities, including block movement, rotation, line clearing,

and scoring.

- Enhance the game with audio effects and a user-friendly interface.

3. Project Description

Scope:

The project encompasses the development of a Tetris game featuring:

- A grid-based game board.
- Seven distinct tetromino shapes with rotation capabilities.
- Real-time user input for block control.
- Line detection and clearing mechanisms.
- Score tracking and display.
- Audio feedback for game events.

Technical Overview:

The game is developed using:

- Programming Language: C++
- Graphics and Audio Library: raylib
- Development Environment: Visual Studio Code
- Build System: Makefile

4. Methodology

Approach:

We adopted an iterative development approach, dividing the project into weekly milestones:

- Week 1: Set up the development environment and familiarize ourselves with raylib.
- Week 2: Design the game architecture using OOP principles.
- Week 3: Implement core game mechanics, including tetromino movement and rotation.
- Week 4: Integrate audio effects and refine the user interface.
- Week 5: Conduct testing, debugging, and final optimizations.

Roles and Responsibilities:

- FABIHA SIDDIQUI : Project lead, responsible for overall architecture and integration.
- ALISHBA AMIR : Focused on implementing game logic and mechanics.
- JITESH KUMAR : Handled graphics rendering and audio integration.

5. Project Implementation

Design and Structure:

The game's architecture is modular, with separate classes for:

- Grid: Manages the game board and line clearing.

- Tetromino: Represents individual blocks, handling shape, rotation, and movement.
- Game: Controls the game loop, user input, and overall state management.

Functionalities Developed:

- Real-time tetromino movement and rotation.
- Collision detection and boundary checks.
- Line completion detection and clearing.
- Score calculation and display.
- Audio feedback for actions like line clearing and game over.

Challenges Faced:

Implementing accurate collision detection required careful consideration of grid boundaries and existing blocks.

Ensuring smooth and responsive user input handling across different platforms.

Integrating audio without causing performance issues.

6. Results

Project Outcomes:

The final product is a functional Tetris game that meets all outlined objectives. The game runs smoothly on multiple platforms and provides an engaging user experience.

7. Conclusion

Summary of Findings:

The project successfully demonstrates the application of OOP principles in game development. By leveraging raylib, we created a cross-platform Tetris game with modular architecture and engaging gameplay.

Final Remarks:

This project provided valuable insights into game development and the practical application of OOP concepts. Future enhancements could include adding levels, implementing a high-score system, and introducing new game modes.