

# Project Title

Tetris Game using Object-Oriented Programming

## Group Members

1. Mohammad Taha Anwer (Roll no: 24K-3033) - Team Lead
2. Syed Sultan
3. Hamza Atif

## Submission Date

April 20, 2025

### 1. Executive Summary

- **Overview:**

This project involves creating a Tetris game using Object-Oriented Programming (OOP). The main tasks include designing the game structure with classes for blocks, grid, and game logic, while also implementing basic features like block movement, scoring, and game over conditions.

- **Key Findings:**

The project was successful in demonstrating the application of OOP principles in game development. Key achievements include a working game with functional block rotation, grid management, and score tracking. We learnt how to use raylib and the compiler needed for its usage.

## 2. Introduction

- **Background:**

Tetris is a classic puzzle game that requires managing falling blocks. This project applies OOP concepts to structure the game's components, making the code modular, maintainable, and easy to extend.

- **Project Objectives:**

- Develop a simple Tetris game using OOP principles.
- Implement core features like block movement, rotation, and collision detection.
- Demonstrate the use of basic game development tools (e.g., Raylib) for graphics and sound.

## 3. Project Description

- **Scope:**

The project includes the creation of blocks, a grid, game logic, and a basic scoring system. Advanced features like multiplayer or online leaderboards were excluded for simplicity.

- **Technical Overview:**

The project was developed using C++ and Raylib for graphics and sound handling. Visual Studio Code was used for coding, and the Raylib library was integrated to handle 2D graphics.

## 4. Methodology

- **Approach:**

The project was developed in iterative phases. The tasks were divided among team members, with regular meetings to integrate each part of the project. The team followed a weekly planning process to ensure timely completion.

- **Roles and Responsibilities:**

- **Taha (Team Lead):** Managed the overall project, implemented the main game loop and block class, and ensured integration.
- **Hamza:** Developed the game logic, sound effects, and block rotation functionality.
- **Sultan:** Worked on the grid class, handling block placements, row clearing, and position management.

## 5. Project Implementation

- **Design and Structure:**

The game was structured around three primary components: blocks, grid, and game logic. The block class handles the shape and movement, the grid class manages the cells and clearing of rows, and the game logic controls the flow and updates.

- **Functionalities Developed:**

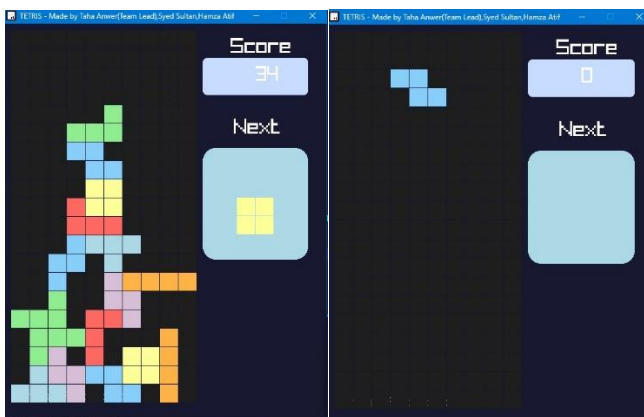
- **Block Movement:** Blocks can move left, right, and down, with rotation functionality.
- **Row Clearing:** The grid detects full rows and removes them, updating the score.
- **Game Over:** A game-over condition is triggered when new blocks cannot fit on the grid.
- **Challenges Faced:**
  - Integrating the game logic with the grid management posed some difficulties, especially with collision detection. The issue was resolved by careful handling of block placement and movement checks.

## 6. Results

- **Project Outcomes:**

The game was successfully implemented with all the core features functioning as expected. The game properly handles block movements, rotations, row clearing, and the score system.

- **Screenshots and Illustrations:**



- **Testing and Validation:**

The game was tested through manual play, checking for issues such as crashes, input lag, or incorrect block placements. All functionalities worked as intended, and the game was deemed stable.

## **7. Conclusion**

- **Summary of Findings:**

The project achieved its goals by creating a functional Tetris game that showcases OOP concepts. The game is modular, making it easy to add new features in the future. We also learnt that to use raylib we must have msys 64 compiler. It doesn't work on minGW compiler.

- **Final Remarks:**

This project provided valuable experience in applying OOP principles to game development. It was a rewarding challenge that enhanced both our programming and teamwork skills. Future improvements could include optimizing performance or adding more advanced game features.