PROJECT PROPOSAL

Project Title

Tetris Game using Object-Oriented Programming

Group Members

1. Mohammad Taha Anwer (Roll no: 24K-3033) - Team Lead
2. Syed Sultan(Roll no: 24K-0585)
3. Hamza Atif(Roll no: 24K-0594)

1. Introduction

- Background:
  This project is focused on developing a simple yet engaging Tetris game using Object-Oriented Programming (OOP). The project will demonstrate the use of OOP concepts like classes, inheritance, encapsulation, and polymorphism. The primary goal is to build a working game that can showcase how OOP helps in creating scalable and maintainable software systems.
- Problem Statement:
  The project aims to address the problem of managing the Tetris game's core components—such as blocks, grid, and game logic—through an OOP-based approach. The key challenge is implementing the game's functionality while keeping the code clean and modular, making it easy to extend or modify in the future.

- Objectives:
  - Develop a fully functional Tetris game with features such as block movement, rotation, and collision detection.
  - Demonstrate the use of basic OOP concepts like classes, objects, and methods.
  - Build a simple user interface to display the game state, including the score and the next block.
  - Implement game over conditions and basic sound effects.

2. Scope of the Project

- Inclusions:
  - The game will include basic features such as moving, rotating, and locking blocks.
  - The grid will manage the placement and removal of blocks.
  - Sound effects will be added for block rotation and line clearance.
  - A scoring system will track the player's progress.
  - A simple game over condition will be triggered when no more blocks can fit.
- Exclusions:
  - Advanced game features like multiplayer support or online leaderboards.
  - High-level optimization for performance.

    - Complex graphical effects or animations.

3. Project Description

- Overview:
  The project involves building a Tetris game using OOP principles. The core components of the game—blocks, grid, and game logic—will be modeled using classes and objects. Each block will be represented as an object, and the grid will be responsible for managing block placements. The game loop will handle user input, block movement, and updating the game state.
- Technical Requirements:
  - C++ for programming.
  - Raylib library for graphics and sound handling.
  - A basic text editor or IDE (e.g., Visual Studio Code or Microsoft Visual Studio).
- Project Phases:
  - Research: Understand the rules of Tetris and the relevant OOP concepts.
  - Planning: Design the architecture of the game, identifying the key classes and their relationships.
  - Design: Implement the classes for blocks, grid, game logic, and handle input/output.
  - Implementation: Write the code for the game mechanics, graphics, and sound.

- Testing: Ensure that the game runs smoothly, with no crashes or bugs.

4. Methodology

- Approach:
  The project will be developed in short, iterative steps, with each member focusing on a specific part of the project. Taha, as the team lead, will ensure smooth communication and assign tasks to the group members. Each member will complete their individual tasks, and then the code will be integrated and tested together.
- Team Responsibilities:
  - Taha (Team Lead): Responsible for managing the project, making the main game loop, and implementing the block class. He will also handle integration and overall testing.
  - Hamza: Will work on implementing the game logic, including sound effects and the rotation of blocks.
  - Sultan: Will focus on the grid class, handling block placement, row clearing, and position management. Sultan will also work on managing the cell structure and checking for full rows.

5. Expected Outcomes

- Deliverables:

  - A fully functional Tetris game that includes a working score system and game-over condition.
  - A short report describing the project, its architecture, and the challenges faced during implementation.
  - User instructions for running the game and interacting with it.
- Relevance:
  The project demonstrates the importance of object-oriented design in organizing complex systems like games. The project also ties into key ICT topics such as data organization (grid management), simple programming (block handling and movement), and user interaction.

6. Resources Needed

- Software:
  - C++ IDE (e.g., Visual Studio Code, Visual Studio).
  - Raylib graphics and sound library.
  - Text editor for writing the documentation (e.g., MS Word, Google Docs).
- Other Resources:
  - Online tutorials for Raylib and Tetris game design.
  - Support and guidance from the seniors for advanced technical issues like raylib not working