Project Overview: Two-Player Tetris Game

Project Description

This project implements a two-player Tetris game where players compete against each other by controlling Tetrimino shapes on separate game boards. The game includes real-time controls, score tracking, and game-over conditions. The objective is to strategically place falling shapes to complete horizontal lines, score points, and prevent the Tetriminos from reaching the top of the board.

Key Components

1. Game Logic

Files: game.cpp, game.h

Description: Contains the core mechanics of the game, including initializing the game, handling player inputs, updating the game board, and determining the game's end conditions. Implements the main game loop, where players can pause, resume, and control their Tetriminos using keyboard inputs.

2. Player Management

Files: Player.cpp, Player.h

Description: Manages the individual player's game board, including initializing the board, validating and executing movements, and updating the game state when a shape is placed. Tracks the player's score and checks for game-over conditions.

3. Shape Handling

Files: shape.cpp

Description: Manages the Tetrimino shapes, including their creation, movement, and rotation on the game board. Handles collisions and interactions with the game board to ensure valid placements and rotations.

4. Point Class

Files: Point.cpp, Point.h

Description: Defines the Point class used to represent coordinates on the game board. Manages the state of each point, including whether it is occupied by a part of a Tetrimino.

5. Main Program

Files: tetris.cpp, tetris.h

Description: Serves as the entry point for the game, initializing the game and starting the main loop. Calls the necessary functions to set up the game, manage players, and handle user inputs.

Features

- Multiplayer Support: Two players can compete against each other on the same machine using separate controls.
- Real-time Gameplay: The game is responsive to player inputs, with real-time shape movement and rotation.
- Score Tracking: Players earn points for completing lines, with the game keeping track of their scores.
- Dynamic Shape Movement: Shapes can be moved, rotated, and placed on the board, with collisions and game rules enforced.

Technologies Used

- Programming Language: C++
- Libraries: Standard C++ Libraries (e.g., <iostream>, <Windows.h>)
- Graphics: Console-based rendering using ASCII characters and cursor manipulation