Overview

This project implements a simple Tetris-like game using **C++**. The game runs in a console window and features basic gameplay mechanics such as moving, rotating, and dropping blocks to form complete rows.

Features

- **Game Mechanics:**
  - Blocks can be moved left, right, rotated, or dropped quickly.
  - Completed rows are cleared, and points are awarded.
  - Game ends when blocks stack up to the top of the playfield.
- **Playfield:**
  - A 15x10 grid where the game takes place.
  - Edges and the bottom of the playfield act as boundaries.
- **Blocks:**
  - Four types of blocks (L-block, T-block, square block, and line block).
  - Each block has four possible rotations, stored as binary 3x3 maps.

How the Game Works

1. **Playfield Initialization:**
   - The playfield is represented as a 1D array of size 150 (15 rows × 10 columns).
   - Boundaries (left, right, and bottom edges) are predefined.
2. **Blocks:**
   - Blocks are defined as binary values stored in an array. Each block has four orientations.
   - Example:
     - `56` corresponds to a specific L-block orientation.
3. **Game Loop:**
   - The main game logic is handled in a loop that continues until the game ends.
   - Key components of the loop:
     - **Drawing:** Updates the playfield and displays it in the console.
     - **Movement:** Blocks move down automatically; user input moves them horizontally or rotates them.
     - **Collision Detection:** Checks for collisions with other blocks or boundaries.
     - **Row Clearing:** Full rows are cleared, and the rows above are moved down.
4. **User Input:**
   - **Arrow Keys:**

        - `Right Arrow`: Move block right.
        - `Left Arrow`: Move block left.
        - `Up Arrow`: Rotate block.
        - `Down Arrow`: Drop block faster.
    - **Escape Key:** Quit the game.
5. **Game Over:**
    - The game ends if a block collides at the top of the playfield.

Code Explanation

Key Variables

- **Playfield (`pl[150]`):**
    - Stores the state of each cell (0 = empty, 1 = filled, edges = fixed values).
- **Blocks (`fig[160]`):**
    - Holds all block shapes and their orientations as binary values.
- **Gameplay Variables:**
    - `off`: Current position of the active block.
    - `r`: Game status flag (1 = running, 0 = game over).
    - `pt`: Player's score (points).
    - `spd`: Speed flag for faster dropping of blocks.

How to Run

1. **Environment Setup:**
    - Ensure you are using a Windows environment since this code relies on `windows.h` for console functions.
    - Use a C++ compiler (e.g., **g++** or **Visual Studio**).
2. **Compile the Code:**

    ```
    bash
    Copy code
    g++ main.cpp -o tetris.exe
    ```

3. **Run the Game:**

    ```
    bash
    Copy code
    tetris.exe
    ```

Game Controls

Key	Action
`Up Arrow`	Rotate the block
`Down Arrow`	Speed up the block
`Left Arrow`	Move block left
`Right Arrow`	Move block right
`Escape`	Quit the game

Future Improvements

- Add more block types for greater variety.
- Implement a better rendering system (replace `std::cout` with a graphical interface).
- Add levels with increasing speed and difficulty.
- Introduce sound effects for a more immersive experience.

Acknowledgments

This project is a basic implementation of a Tetris-inspired game and demonstrates:

- Console-based rendering.
- Basic collision detection and handling.
- User input processing in real-time.

Feel free to modify and enhance this code to make it your own! 🎮