Tetris Game Documentation

1. MyTetrisGame Class

This is the main class of the game, extending the Game class from LibGDX. It initializes the TetrisScene and GameOverScene, and sets the initial screen to TetrisScene. It also handles the rendering and disposing of the game.

2. TetrisScene Class

This class implements the Screen interface from LibGDX and represents the main game scene. It handles the rendering of the game, user inputs, and game logic such as moving and rotating the blocks, checking for collisions, and deleting completed lines.

2.1 User Inputs

The class captures user inputs for moving and rotating the blocks. The blocks can be moved left, right, or down and can be rotated clockwise or counterclockwise.

2.2 Game Logic

The class handles the game logic, which includes moving and rotating the blocks, checking for collisions with the walls or other blocks, and deleting completed lines.

3. GameOverScene Class

This class also implements the Screen interface from LibGDX and represents the game over scene. It currently only clears the screen with a red color, but it can be extended to display a 'Game Over' message and the player's final score.

4. BlockManager Class

This class manages the blocks in the game. It generates new blocks, checks for collisions, and deletes completed lines. It also keeps track of the current block and all the old blocks that have already been placed.

5. Block Class

This is an abstract class that represents a block in the game. It contains a list of Cube objects that make up the block. It handles the rendering of the block, moving and rotating the block, and checking if the block can move or rotate.

6. Cube Class

This class represents a single cube in a block. It has a position and a texture. It handles the rendering of the cube and moving the cube.

7. GameOverScene Class

This class represents the game over scene. It currently only clears the screen with a red color, but it can be extended to display a 'Game Over' message and the player's final score.

Each of the block types (BlockI, BlockJ, BlockL, BlockO, BlockS, BlockT, BlockZ) would extend the Block class and define their own shape and rotation behavior. These classes are not shown in the provided code, but they would be necessary for the game to work.