Grid object → Basically a 2D array
→ each [i][j] can have color from the vector<color>
→ Draw() just uses DrawRectangle function m x n times to Produce the game platform & Blocks.

class
Position { row , col }

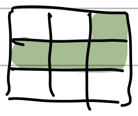
class
Block → cell size, id (for color)

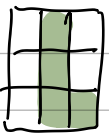
this is Basically a 3x3 grid where some cells are colored to give illusion of shape.

↓
unordered_map <int , vector<Position>> cells .
↳ storing the position of the cells according to the rotation state.

Blocks.cpp → Storing the different shape blocks.

```
E.g.  class LBlock : Public Block {
          id = 2
          cells[0] = {{0,2}, {1,0}, {1,1}, {1,2}}
          cells[1] =
          cells[2] =
          cells[3] =
                                        };
```

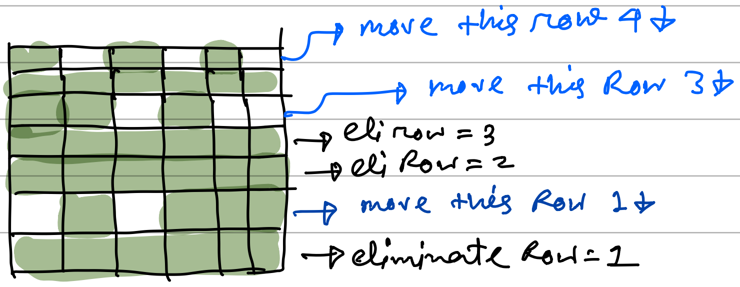


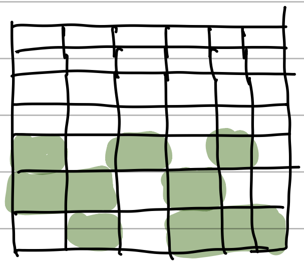


class
Game → handling block spawning, freezing when reached bottom
input handling moving, Rotating,
constraining block inside frame etc.

# Freezing when reached bottom
↳ Color the block's positions permanently
curblock = Nextblock
nextblock = getRandomBlock().

∅ Row elimination logic.





✓ DrawTextEx (Font font, char* txt, Vector2 Position,
float fontsize, spacing, color)

○ <u>RayLib basics</u>: Color { red, green, blue, alpha }.

void DrawRectangle(int posX, PosY, width, height, color)
similarly DrawCircle, DrawPoly, DrawLine etc.