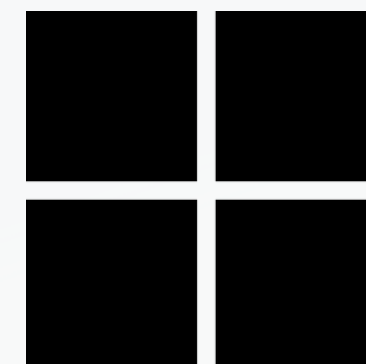




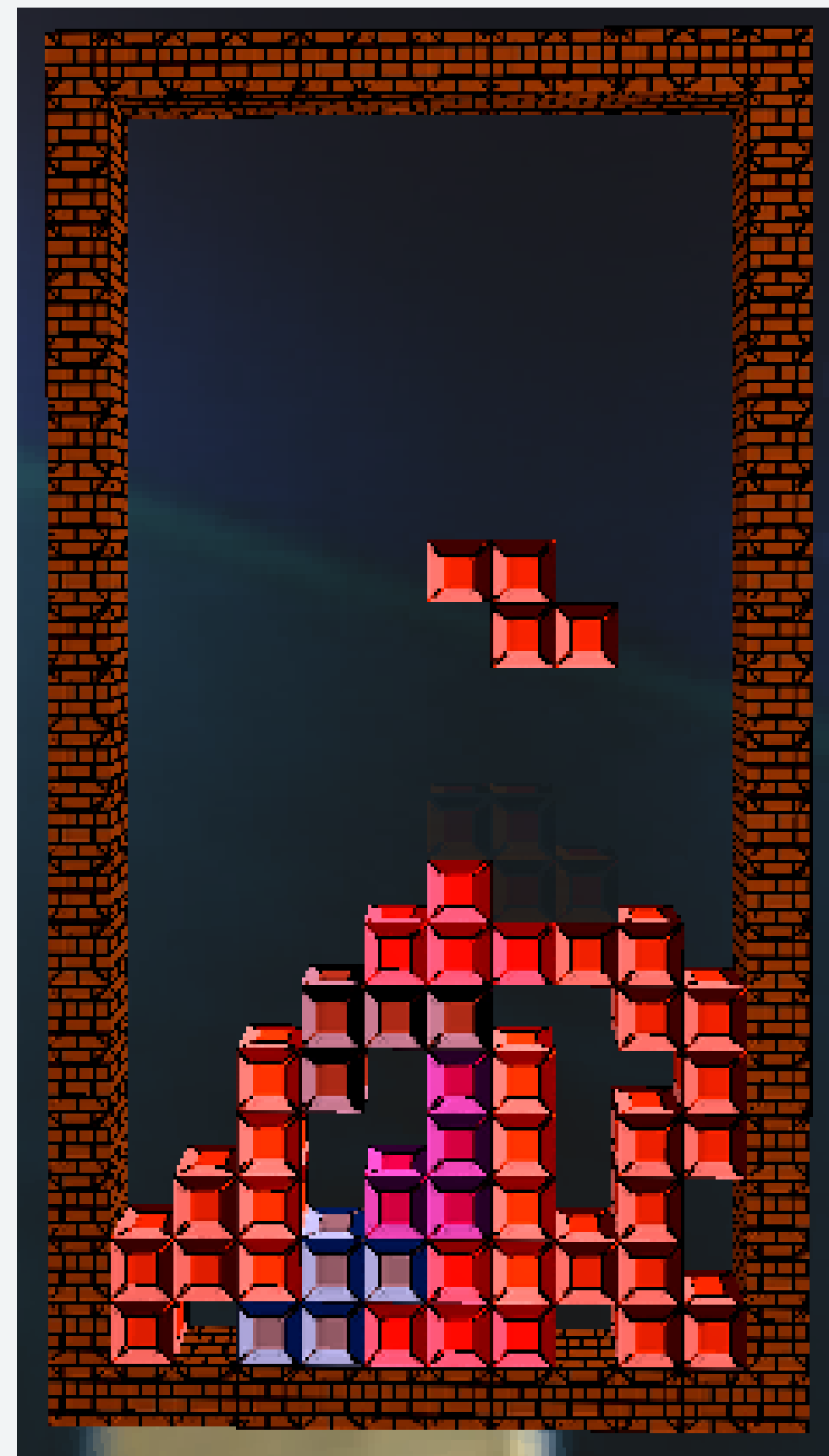
Tetris 3D

PROJETO COMPUTAÇÃO GRÁFICA



Desenvolvimento

Modelação das peças do jogo



Desenvolvimento

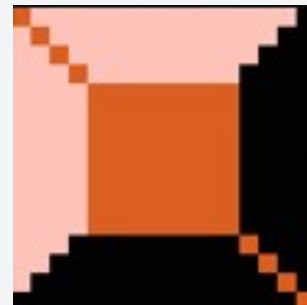
Implementação dos movimentos descendentes das peças com respectivas translações e rotações

```
// check left, right and rotate
if (leftPress)
{
    if (isPastInterval(0.12))
    {
        pos->col--;
        if (board->isCollision(*currentTetromino, *pos))
            pos->col++;
    }
}
else if (rightPress)
{
    if (isPastInterval(0.12))
    {
        pos->col++;
        if (board->isCollision(*currentTetromino, *pos))
            pos->col--;
    }
}
else if (downPress)
{
    if (isPastInterval(0.12))
    {
        pos->row++;
    }
}
```

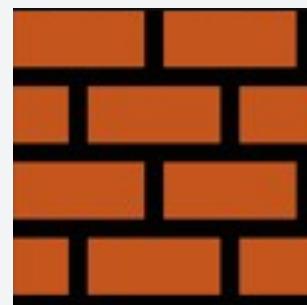
```
else if (rotate)
{
    currentTetromino->rotation = (currentTetromino->rotation + 1) % 4;
    if (board->isCollision(*currentTetromino, *pos))
    {
        pos->col--;
    }
    rotate = false;
}
```

Desenvolvimento

Texturização das peças



Textura base das peças



Textura da borda do jogo

Desenvolvimento

Utilização do teclado, rato para interagir com o jogo.

```
void handleKeyboardInput(GLFWwindow *window)
{
    if (glfwGetKey(window, GLFW_KEY_W) == GLFW_PRESS)
        camera.ProcessKeyboard(FORWARD, deltaTime);
    if (glfwGetKey(window, GLFW_KEY_S) == GLFW_PRESS)
        camera.ProcessKeyboard(BACKWARD, deltaTime);
    if (glfwGetKey(window, GLFW_KEY_A) == GLFW_PRESS)
        camera.ProcessKeyboard(LEFT, deltaTime);
    if (glfwGetKey(window, GLFW_KEY_D) == GLFW_PRESS)
        camera.ProcessKeyboard(RIGHT, deltaTime);
    // Add more controls as needed
}
```

```
void mouse_callback(GLFWwindow *window, double xpos, double ypos)
{
    if (glfwGetMouseButton(window, GLFW_MOUSE_BUTTON_LEFT) == GLFW_PRESS)
    {
        if (firstMouse)
        {
            lastX = xpos;
            lastY = ypos;
            firstMouse = false;
        }

        float xoffset = xpos - lastX;
        float yoffset = lastY - ypos; // reversed since y-coordinates go from bottom to top

        xoffset *= sensitivity;
        yoffset *= sensitivity;

        lastX = xpos;
        lastY = ypos;

        camera.ProcessMouseMovement(xoffset, yoffset);
    }
}
```

EXTRAS

Highscore



EXTRAS

Próximas 4 peças



RESULTADO FINAL

