

Universidade da Beira Interior

Departamento de Informática



**Departamento de
Informática**

Nº 9 - 2021: *Tetris 3D*

Elaborado por:

Afonso Correia - nº41130

Carlos Martins - nº41968

Rafael Louro - nº41855

Rui Ferreira - nº41064

Docente:

Professor Doutor Abel Gomes

15 de janeiro de 2021

Conteúdo

Conteúdo	i
1 Introdução	1
1.1 Enquadramento	1
1.2 História do jogo	1
1.3 Motivação	2
1.4 Objetivos	2
1.5 Organização do Documento	2
2 Tecnologias Utilizadas	3
3 Desenvolvimento e Implementação	5
3.1 Etapas do desenvolvimento	5
3.2 Descrição do funcionamento do Software	5
4 Conclusões e Trabalho Futuro	9
4.1 Conclusões Principais	9
4.2 Trabalho Futuro	9
Bibliografia	11

Capítulo

1

Introdução

1.1 Enquadramento

De modo a realizar com sucesso o projeto da cadeira de computação gráfica, optámos por escolher o projeto “Tetris 3D”.

1.2 História do jogo

O Tetris é considerado um jogo de quebra-cabeças, desenvolvido por Alexey Pajitnov, enquanto trabalhava no Centro de Informática de Dorodnicyn, em Moscovo.

O seu lançamento ocorreu no dia 6 de junho de 1984, e o seu nome deriva da junção do prefixo grego “tetra” – uma vez que todos os blocos são formados por quatro segmentos – com a palavra Ténis, dado a ser o desporto favorito do programador.

O desenvolvimento do algoritmo do funcionamento do Tetris foi inspirado também num jogo de quebra-cabeças popular, o poliminó.

O mecanismo de funcionamento deste jogo baseia-se em empilhar tetramínos que descem a janela do jogo numa velocidade que aumenta à medida que a dificuldade do jogo também evolui, de forma a completar linhas horizontais.

No momento que se forma uma linha completa, as camadas superiores caem, e consequentemente o jogador ganha pontos. Quando a pilha de peças chega ao topo da tela, o jogo termina.

Desde a sua criação, o Tetris é um dos jogos mais jogados mundialmente.

1.3 Motivação

A principal motivação encontrada para o desenvolvimento deste projeto vai de encontro com o interesse em aplicar na prática os conhecimentos adquiridos ao longo da cadeira de Computação Gráfica, os quais podem vir a ser bastante úteis quer para nossa vida universitária quer para a futura vida profissional.

1.4 Objetivos

Alguns dos objetivos que nos propomos a cumprir vão de encontro a proporcionar uma jogabilidade simples, bem como ser visualmente apelativo, para além dos requisitos necessários para que tudo funcione de acordo com o pretendido.

1.5 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.
2. O segundo capítulo – **Tecnologias Utilizadas** – descreve os conceitos mais importantes no âmbito deste projeto, bem como as tecnologias utilizadas durante do desenvolvimento da aplicação.
3. No terceiro capítulo – **Desenvolvimento e Implementação** – explicamos de forma mais detalhada a gestão do projeto e a maneira como foi implementado o código.
4. No quarto capítulo – **Conclusões** – é feita uma reflexão crítica sobre os objetivos atingidos e não atingidos, bem como o que se poderia melhorar futuramente.

Capítulo

2

Tecnologias Utilizadas

Neste projeto, abordamos e explorámos mais profundamente algumas das tecnologias abordadas no decorrer das aulas.

De modo a atingir um dos principais objetivos do trabalho, recorreremos ao FreeGLUT que nos permitiu criar e gerir a janela de jogo, como também a implementação de funções para ler instruções através do rato ou teclado do computador.

Com base nesta biblioteca, conseguimos redirecionar os vários blocos do Tetris a partir das setas do teclado, mudar as posições dos mesmo aquando o selecionar da tecla espaço. Para o rato definimos também funcionalidades relativas ao zoom da janela, como também à mudança de perspetiva da mesma.

Outras das bibliotecas usadas para a realização do projeto consiste na GLEW, responsável por fornecer mecanismo de ajuda na consulta e carregamento de extensões OpenGL. Esta biblioteca disponibiliza algoritmos de otimização de tempo de execução para determinais quais extensões de OpenGL são suportadas na plataforma destino. A GLEW está definida no cabeçalho do programa.

Capítulo

3

Desenvolvimento e Implementação

3.1 Etapas do desenvolvimento

Começámos por definir cada cubo individualmente e usámos multiplas translações para criar os limites do jogo. Para fazer cada uma das formas basta juntar um cubo base várias vezes de acordo com a forma desejada

3.2 Descrição do funcionamento do Software

É criada uma peça automaticamente de forma aleatória, reunindo vários cubos, em que através da barra de espaço podemos alterar a sua orientação, bem como mover para a esquerda ou direita através das respetivas setas. Para alterar a velocidade de movimento da peça podem ser usadas as setas para cima e para baixo, sendo que a seta para baixo aumenta a velocidade do movimento ao passo que a seta para cima coloca a peça diretamente no local pretendido, de acordo com a coluna em que se encontra. A cada linha completa, esta é eliminada, e são adicionados 100 pontos à pontuação total. Caso o jogador não consiga completar linhas, se as peças atingirem o topo do painel do jogo, o mesmo irá terminar, uma vez que perdeu o jogo.

A seguinte função permite definir ações às teclas sendo assim possível mover as peças através das setas bem como apresentar o jogo em modo de ecrã inteiro através da tecla f1.

```
void teclas_Setas(int key, int x, int y)
{
    switch (key)
    {
        case GLUT_KEY_F1:
            fullscreen = !fullscreen;
            if (fullscreen) glutFullScreen();
            else glutReshapeWindow(500, 500);
            break;
        case GLUT_KEY_UP:
            game->DropBlock();
            break;
        case GLUT_KEY_DOWN:
            game->IncreaseBlockSpeed();
            break;
        case GLUT_KEY_RIGHT:
            game->MoveBlock(true);
            break;
        case GLUT_KEY_LEFT:
            game->MoveBlock(false);
            break;
        default:
            break;
    }
}
```

Figura 3.1: Função para definir teclas

```
void zoom_rato(int wheel, int direction, int x, int y)
{
    cameraPos[2] = std::min<GLfloat>(MIN_ZOOM, std::max<GLfloat>(MAX_ZOOM, cameraPos[2] - direction * 0.3f));
}
```

Figura 3.2: Função para fazer zoom

Através desta função, é possível fazer zoom com a roda do rato, uma vez que ao deslizar para cima a câmara se aproxima do jogo ao passo que ao deslizar para baixo se afasta do mesmo.

```
void drawBasicBlock(bool withBorder)
{
    defColor(color);

    glEnable(GL_TEXTURE_2D);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
    desenharCubo(1.0f);

    glDisable(GL_TEXTURE_2D);
}
```

Figura 3.3: Função para desenhar um bloco

Esta função serve para criar um bloco, atribuindo-lhe as textura já inicializadas anteriormente.

```
void teclas_letras(unsigned char key, int x, int y)
{
    switch (key)
    {
        case 'm':
            soundPaused = !soundPaused;
            if (!soundPaused)
                PlaySoundTetris(nullptr, nullptr, 0);
            else
                PlaySoundTetris(TEXT("../src/main.wav"), nullptr, SND_LOOP | SND_ASYNC);
            break;
        case 'r':
            cameraPos[0] = 2.0f;
            cameraPos[1] = 3.0f;
            cameraPos[2] = 10.0f;
            lookout[0] = 2.0f;
            lookout[1] = 3.0f;
            lookout[2] = -8.0f;
            break;
        case 'c':
            game->ChangeBlock();
            break;
        case ' ':
            game->RotateActiveBlock();
            break;
        case 27: // ESC
            exit(EXIT_SUCCESS);
            break;
        default:
            break;
    }
}
```

Figura 3.4: Função para atribuir funcionalidades às teclas

Apresentamos nesta função a forma como atribuímos funcionalidades às teclas, sendo que a tecla m(MUTE) faz com que a música pare automaticamente, a tecla r faz com que a câmara volte à sua posição inicial, caso tenha sido feita alguma alteração na mesma.

A tecla c altera a ultima peça que é apresentada no ecrã, a tecla SPACE efetua uma rotação de 45° sobre peça que o utilizador está a controlar, no sentido contrário aos ponteiros do relógio. Caso deseje sair do jogo basta clicar na tecla ESC.

```
void initTextures()
{
    glEnable(GL_TEXTURE_2D);
    glGenTextures(numTextures, textureName);

    const char* filename[numTextures] = { "../src/brick1.bmp" };

    // Carregamos la textura
    glBindTexture(GL_TEXTURE_2D, textureName[0]);
    RgbImage texture(filename[0]);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, texture.GetNumCols(), texture.GetNumRows(), 0, GL_RGB, GL_UNSIGNED_BYTE, texture.ImageData());
    glBuild2DMipmaps(GL_TEXTURE_2D, 3, texture.GetNumCols(), texture.GetNumRows(), GL_RGB, GL_UNSIGNED_BYTE, texture.ImageData());

    // Configuramos la textura
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_BLEND);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
}
```

Figura 3.5: Função para inicializar as texturas

Usámos esta função que irá inicializar as texturas para posteriormente podermos atribui-las às peças através das funções destinadas a desenhar as peças do jogo.

Capítulo

4

Conclusões e Trabalho Futuro

4.1 Conclusões Principais

Elaborando uma reflexão crítica relativamente aos objetivos propostos versus os objetivos atingidos, concluímos que estamos satisfeitos com o trabalho elaborado e as aprendizagens adquiridas e solidificadas com o desenvolvimento do projeto.

Com a execução deste trabalho, não só conseguimos aprofundar os nossos conhecimentos no OpenGL, como na exploração das diversas funcionalidades existentes.

Paralelamente, adquirimos também conhecimento sobre uma linguagem que pouco que tínhamos utilizado até então, e que pode ser útil quer na vida académica quer na futura vida profissional.

Relativamente à dinâmica do funcionamento do grupo, tentámos distribuir as tarefas de forma equivalente de modo a que todos tivéssemos a mesma carga de trabalho, contudo fomos estabelecendo contacto regularmente com o intuito de cooperar na elaboração e resolução dos problemas que apareciam com o desenvolvimento do projeto.

Resumidamente, podemos concluir que tivemos uma dinâmica bastante positiva.

4.2 Trabalho Futuro

Futuramente, um dos aspetos a melhorar no projeto desenvolvido assenta na criação de um menu na fase inicial do jogo, a partir do qual é possível seleccionar o nível de dificuldade que se pretende jogar, passando a existir três níveis (nível iniciante, nível intermédio, nível avançado). Outro tema a abor-

dar para o melhoramento do trabalho desenvolvido, foca-se na elaboração de um menu no final de cada jogo, dando ao jogador a hipótese de voltar a jogar novamente.

Bibliografia

Learn OpenGL, extensive tutorial resource for learning Modern OpenGL(último acesso 12 janeiro).

GitHub - andykhv/Tetris3D: Tetris powered by OpenGL(último acesso 13 janeiro)

GitHub - carlos93/Tetris3D: Game based on Tetris. Uses OpenGL for 3D design.(último acesso 13 janeiro)

Displaying Text with OpenGL – Jérôme Belleman (jeromebelleman.gitlab.io)(último acesso 2 janeiro)