

# **Universidade da Beira Interior**

## **Departamento de Informática**



**Departamento de  
Informática**

### **Projeto Computação Gráfica Tetris 3D**

Elaborado por:

**Emanuel Pacheco 47712**

**Ricardo Pereira 48554**

**Rodrigo Fonseca 48658**

**Vicente Chã 48160**

Orientador:

**Professor Doutor Abel João Padrão Gomes**

9 de janeiro de 2024



# ***Agradecimentos***

Este trabalho só é possível devido à ajuda do professor orientador da Unidade Curricular de Computação Gráfica, dos nossos familiares e dos amigos, uma vez que nos ofereceram todo o apoio possível para a realização deste trabalho.



# Conteúdo

<b>Conteúdo</b>	<b>iii</b>
<b>1 Motivação</b>	<b>1</b>
1.1 Enquadramento do Trabalho . . . . .	1
1.2 Motivação . . . . .	1
1.3 Organização do Documento . . . . .	1
<b>2 Tecnologias Utilizadas</b>	<b>3</b>
2.1 Introdução . . . . .	3
2.2 Tecnologias utilizadas . . . . .	3
2.2.1 Linguagem de Programação C++ . . . . .	3
2.2.2 OpenGL . . . . .	3
2.2.3 GLFW . . . . .	3
2.2.4 GLAD . . . . .	4
2.2.5 GLM . . . . .	4
2.2.6 L <sup>A</sup> T <sub>E</sub> X . . . . .	4
2.2.7 Discord . . . . .	4
2.3 Conclusões . . . . .	4
<b>3 Etapas de Desenvolvimento</b>	<b>5</b>
3.1 Introdução . . . . .	5
3.2 Funcionamento Geral . . . . .	5
3.2.1 Começo de código . . . . .	5
3.2.2 Parte Principal . . . . .	6
3.3 Funcionalidades Extras . . . . .	7
3.4 Conclusões . . . . .	8
<b>4 Descrição do funcionamento do Software</b>	<b>9</b>
4.1 Introdução . . . . .	9
4.2 Objetivo . . . . .	9
4.3 Obtenção de Pontos . . . . .	9
4.4 Funcionalidades . . . . .	10
4.4.1 Atalhos de Movimentação . . . . .	10

4.4.2	Atalho de Queda . . . . .	10
4.4.3	Atalho de Armazenamento . . . . .	10
4.5	Conclusões . . . . .	10
<b>5</b>	<b>Conclusões e Trabalho Futuro</b>	<b>11</b>
5.1	Conclusão . . . . .	11
5.2	Trabalho Futuro . . . . .	11
5.3	Considerações Finais . . . . .	11
5.4	Referências Bibliográficas . . . . .	12

## **Capítulo**

# 1

## **Motivação**

### **1.1 Enquadramento do Trabalho**

Este documento é um relatório em  $\text{\LaTeX}$  referente ao trabalho prático pertencente à Unidade Curricular de Computação Gráfica com o intuito de obter um jogo baseado em Tetris, sendo um projeto principalmente apoiado nas áreas de Informática, Tecnologia e Programação.

### **1.2 Motivação**

De todos os trabalhos apresentados para escolha o Tetris foi o que despertou mais interesse no grupo, de modo a evitar fazer o Tetris simples já conhecido mundialmente a duas dimensões, o grupo decidiu escolher o Tetris em três dimensões para ser diferente do jogo comum.

Estes problemas devem ser abordados não só para ajudar na nossa compreensão do manuseio e funcionamento de texturas, cores e estruturas, mas também para demonstrar um pouco das funções e operações que os alunos já conseguem efetuar após as aulas práticas e teóricas presenciadas.

### **1.3 Organização do Documento**

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

1. Introdução (Capítulo 1) tem como intuito apresentar o trabalho, dividindo assim a sua apresentação pelas várias secções apresentadas no índice e descritas posteriormente em cada uma das suas partes.

2. Tecnologias Usadas (Capítulo 2) serve para apresentar todos os suportes técnicos e informáticos usados para a execução do trabalho.
3. Etapas de Desenvolvimento (Capítulo 3) denota e descreve a construção das funções e funcionalidades relativas ao trabalho.
4. Descrição do funcionamento de *Software* (Capítulo 4) tem como função apresentar o objetivo, atalhos e como se obtém pontos no Tetris e no *software* desenvolvido para este projeto prático.
5. Trabalho Futuro (Capítulo 5.2) tem como intuito apresentar ideias adicionais ou que faltaram ao programa implementado.
6. Considerações Finais (Capítulo 5.3) tem como intuito refletir sobre o trabalho efetuado e os problemas que neles foram mais complexos.
7. Referências Bibliográficas (Capítulo 5.4) pretende apresentar as referências e *websites* usados para desenvolver o projeto prático.



## **Capítulo**

# 2

## ***Tecnologias Utilizadas***

### **2.1 Introdução**

Neste capítulo, será falado sobre as tecnologias utilizadas no desenvolvimento do Tetris 3D. Será explicado porque a linguagem C++ foi escolhida e como as bibliotecas gráficas, OpenGL, GLFW, GLAD e GLM, foram usadas para renderizar gráficos e interagir com a janela e objetos do jogo.

### **2.2 Tecnologias utilizadas**

#### **2.2.1 Linguagem de Programação C++**

A escolha da linguagem de programação C++ para o desenvolvimento do Tetris 3D foi influenciada principalmente pelo seu uso nas aulas práticas. Além disso, C++ é reconhecida pela sua eficiência e controlo sobre recursos de sistemas gráficos, características essenciais para jogos e aplicações gráficas.

#### **2.2.2 OpenGL**

OpenGL foi a principal API gráfica utilizada. Foi essencial para desenhar e transformar as formas geométricas do Tetris em um ambiente 3D.

#### **2.2.3 GLFW**

O GLFW oferece uma ‘interface’ para criar janelas e gerir eventos de entrada como pressionamento de teclas e movimentos do rato, essencial para a interação do utilizador com o programa.

### 2.2.4 GLAD

O GLAD é uma biblioteca que gerencia a extensão e a funcionalidade do OpenGL, foi bastante útil para carregar várias funções do *OpenGL* de forma portátil e fácil.

### 2.2.5 GLM

GLM é uma biblioteca matemática para gráficos baseada na especificação da GLSL(OpenGL Shading Language). Ela fornece funcionalidades matemáticas, especialmente para transformações 3D, como translação, rotação e escala, o que foi crucial para o posicionamento e movimento das peças no Tetris 3D.

### 2.2.6 L<sup>A</sup>T<sub>E</sub>X

Para a realização deste relatório foi usado o L<sup>A</sup>T<sub>E</sub>X para obter um resultado mais profissional e bem estruturado, facilitando a leitura e compreensão do conteúdo.

### 2.2.7 Discord

Para a comunicação e coordenação do grupo durante o desenvolvimento do projeto, foi utilizada a plataforma Discord. O Discord oferece um meio eficaz de comunicação por texto, voz e vídeo, facilitando a colaboração remota entre os membros do grupo.

## 2.3 Conclusões

Com a conclusão deste capítulo percorremos e definimos todas as tecnologias fundamentais usadas como base para o desenvolvimento deste projeto prático.

## Capítulo

# 3

## ***Etapas de Desenvolvimento***

### **3.1 Introdução**

Este capítulo está encarregue por definir o funcionamento interno do *software* com base nas funções implícitas nos ficheiros, funcionalidades essenciais que tornam o *software* único e a definição detalhada das funções das funcionalidades extras.

Maior parte do código já estava escrito através do projeto de *github* fornecido pelo professor da unidade curricular em questão, sendo o dono do código o *user* Andy Khov. Se quiserem verificar a página do projeto, segue-se o *link* do projeto.

### **3.2 Funcionamento Geral**

#### **3.2.1 Começo de código**

A *software* começa por criar um tabuleiro vazia, sendo este onde será armazenado as várias informações do nosso jogo, tal como as posições das peças. São criadas várias variáveis, onde são guardadas as informações da peça atual, e as quatro peças futuras. Também é guardada a posição atual da peça atual.

Cada peça começa no meio do tabuleiro, na linha mais alta possível, sendo esta peça uma das quatro disponíveis do jogo, decidida aleatoriamente. As 4 peças futuras também são decididas aleatoriamente.

É depois definido as variáveis que vão guardar a peça que o *user* quiser guardar, inicializadas a menos um visto que este valor não é usado para as

peças.

### 3.2.2 Parte Principal

Anteriormente criada a aplicação e a janela, é na função *render* dentro da *main* onde é realizado maior parte do jogo.

Nesta função, é onde são desenhados e caracterizado os vários objetos que constituem as peças, os textos representado à volta do tabuleiro, os vários controlos, e também o nosso *background*, sendo este uma esfera.

Inicialmente, é renderizada a esfera com a textura do *background*, é definido o tamanho de cada cubo usado nas peças e no tabuleiro, e também a posição da câmara.

A seguir, são feitos os vários controlos do jogo, onde cada tecla usada tem uma variável associada, e quando clicada, atribui a essa variável o valor de *True* na função *keyCallback()*. Quando verdadeiras, estas movem-se conforme o suposto, aumentando ou diminuindo a respetiva posição da peça enquanto as teclas são carregadas, em intervalos de 0.12 segundos (para mover as peças, as teclas utilizadas são as setas *down*, *left*, *right*).

Também é verificado se as teclas *spacebar*, *up* e *C key* são pressionadas para fazer os vários controlos extras, sendo o *spacebar* para meter a peça no fundo máximo possível, *up* para rodar a peça atual, e *C key* para guardar a peça atual.

Agora é verificado se a peça atual está em colisão com alguma parede ou com o solo, se estiver em colisão com o solo, essa peça é colocada, guardada no tabuleiro, e então é renderizada a seguinte peça disponível no *stack*.

Antes de colocada a próxima peça, é verificado se a peça anteriormente colocada ultrapassa a linha máxima permitida, se sim é limpo o tabuleiro e é guardado o *score* no *highscore* caso este seja maior que o atual.

Cada peça é constituída por quatro cubos, sendo o formato das peças e as suas rotações definidos no *tetromino.cpp*, e as texturas e as dimensões destes definidos no ficheiro *shape.cpp*. As cores de cada peça, tal como a textura de alguns objetos, é alterado no ficheiro *shader\_fragment\_block.glsl*, na pasta *shaders*, sendo estes desenhados, dependente da variável enviada.

Como dito inicialmente, a peça que aparece no tabuleiro é desenhada no meio, na altitude máxima do tabuleiro, para mais facilidade de compreensão, nas coordenadas do estilo (x,y), sendo x a sua coordenada na horizontal, e y na coordenada da vertical. A seguir, é desenhada o correspondente desta peça, tal como na rotação definida, uma sombra da peça na respetiva coordenada y, mas na altitude mínima do tabuleiro, para simular onde é que a peça vai se encaixar, e como ficará relativamente às restantes peças já anteriormente postas.

A seguir, é renderizada as futuras quatro peças que vão aparecer no tabuleiro, aparecendo a peça mais em cima no tabuleiro quando a peça atual é posta. Ao fim destas, é renderizada também a peça que o jogador está a segurar atualmente.

Ao fim disto, é escrito as várias informações relativas ao jogo em volta do tabuleiro, como o *score*, o *highscore*, etc. Os resultados, tal como o *score* e o *highscore* são desenhados na função *drawScore()*, recebendo esta função vários parâmetros como o resultado atual e o melhor resultado atual, sendo estes desenhados a partir de vários cubos, sendo o formato destes números definidos em matrizes na função *drawNumber()* e também dimensionados, a posição inicial dos números é definida no *drawScore()*.

Depois são escritas as palavras "*hold*" e "*next*" a partir do mesmo raciocínio dos números, sendo o formato destes definidos em matrizes, e dimensionados e posicionados nas respetivas funções (*drawWordNext()* e *drawWordHold()*).

### 3.3 Funcionalidades Extras

Foi implementada uma câmara fornecida pelos vários exercícios do professor ao longo do semestre, que serve para ser possível haver movimentação pelo espaço tridimensional criado.

Foram implementadas várias mudanças do estilo *Quality of Life* que melhoram a jogabilidade, estas sendo:

1. Mostrar os próximos quatro blocos (Só mostrava um único),
2. Correção de *bug* que não permitia certo tipos de blocos rodar quando em colisão com a parede direita,

3. Ficando a pressionar uma tecla, a ação continua, em vez de só se realizar uma vez e parar imediatamente;

Também foi implementado um sistema de *highscore*, que guarda o melhor resultado do programa atual.

### 3.4 Conclusões

Em síntese, o software do jogo Tetris começa criando um tabuleiro vazio e estabelecendo variáveis para as peças. A função principal de renderização desenha os objetos, como peças e texto, enquanto lida com controles e atualizações do jogo. Este capítulo fornece uma visão geral da estrutura básica do código, dando início à compreensão do funcionamento fundamental do jogo Tetris.

## **Capítulo**

# 4

## ***Descrição do funcionamento do Software***

### **4.1 Introdução**

Neste capítulo, é explorado o funcionamento do jogo desenvolvido neste projeto (baseado em Tetris). Aqui, encontrará uma descrição detalhada das ações possíveis a serem efetuadas no jogo, o objetivo do mesmo e a maneira como poderá alcançar o seu objetivo mais eficientemente.

### **4.2 Objetivo**

O objetivo no Tetris visa completar linhas horizontais com peças tetraminó (formadas por quatro blocos cada) sem deixar espaços vazios, evitando que essas linhas cheguem ao topo da grade do jogo. O jogo continua, e as peças descem cada vez mais rápido consoante o desenrolar do jogo e conforme o tempo e pontuação que o jogador obtém.

### **4.3 Obtenção de Pontos**

Quando uma linha é completada ela desaparece e o jogador ganha pontos, esses pontos variam consoante o número de linhas que o jogador consegue completar. Alguns sistemas de pontos em modos de jogo e/ou *websites* e jogos que se baseiam no Tetris, conforme o jogador é mais rápido, maior será a sua pontuação ao efetuar uma ou mais linhas.

No caso do *software* desenvolvido para o projeto prático, o *highscore* é inicializado a 0 sempre que o programa inicia e sempre que é batido esse *highscore* a variável é atualizada no fim de cada jogo.

## 4.4 Funcionalidades

### 4.4.1 Atalhos de Movimentação

Os controlos do jogo do Tetris consistem em: mover a peça para a esquerda, direita ou para baixo, move-a instantaneamente para o fundo do mapa e rodá-la num ângulo de 90°.

### 4.4.2 Atalho de Queda

De modo a possibilitar uma descida instantânea nas peças com o intuito de poupar tempo e tornar o jogo menos chato para alguns jogadores, o Tetris tem geralmente a tecla **Espaço** para possibilitar essa descida dos tetraminós.

### 4.4.3 Atalho de Armazenamento

Para guardar uma peça que poderá ser inoportuna no momento que o jogo se está a desenrolar ou para permitir guardar a peça para um uso futuro, o Tetris tem geralmente a tecla **C** para possibilitar o armazenamento temporário de uma dada peça que o jogador queira, podendo apenas ser possível a troca de uma peça por jogada.

## 4.5 Conclusões

Como conclusão do capítulo, foram apresentados os princípios-base do Tetris, o seu objetivo e como é possível a obtenção de pontos e qual a melhor maneira de obter esses pontos de modo a guiar qualquer utilizador que pretenda utilizar o *software* desenvolvido ou até mesmo o Tetris em si.



## Capítulo

# 5

## ***Conclusões e Trabalho Futuro***

### **5.1 Conclusão**

A conclusão deste trabalho permitiu expandir o conhecimento sobre o OpenGL, programação e modelação de figuras geométricas, aplicação de cor, texturas e iluminação nestas, bem como a programação em geral.

### **5.2 Trabalho Futuro**

Neste projeto todos os requisitos foram feitos, porem, poderiam ter sido adicionadas mais implementações para que o trabalho ficasse mais completo, como, por exemplo, a adição de sons aos vários elementos do jogo, fazer alguns mini jogos além do Tetris original para o jogador ter mais escolhas, implementar uma opção para ser possível jogarem 2 pessoas ao mesmo tempo, e criar uma *interface* simples e de fácil compreensão para juntar todas as funcionalidades a cima mencionadas.

### **5.3 Considerações Finais**

Neste trabalho, é refletida a jornada de desenvolvimento do projeto de Tetris 3D, explorando as suas realizações. Durante o seu desenvolvimento, foi demonstrado as complexidades da renderização tridimensional e as vastas formas de como foi utilizado o OpenGL e as suas bibliotecas, implementando sempre todas as funcionalidades com otimização e uma boa prática de programação em mente.

## 5.4 Referências Bibliográficas

1. **Enunciado**

URL: <https://www.di.ubi.pt/agomes/cg/projeto/05tetris.pdf>

2. **Website da UC**

URL: <https://www.di.ubi.pt/agomes/cg/>

3. **Código Fonte**

URL: <https://github.com/andykhv/Tetris3D>