

Universidade da Beira Interior

Departamento de Informática



**Departamento de
Informática**

Computação Gráfica – Relatório do Projeto Prático

Elaborado por:

Rúben Santos nº 41308
Gonçalo Domingos nº 41719
Rui Barata nº 41872

Docente:

Professor Abel João Padrão Gomes

Índice

Índice de figuras	3
1. Introdução	4
2. Motivação.....	5
3. Tecnologias Utilizadas	6
OpenGL:.....	6
GLFW:	6
GLEW:	6
GLM:.....	6
Visual Studio Code:	6
C++:	6
Git:	7
Trello:.....	7
4. Etapas de Desenvolvimento	8
5. Descrição do funcionamento do Software.....	9
6. Trabalho Futuro.....	11
7. Considerações Finais.....	12
Referência Bibliográfica	13

Índice de figuras

Figura I - Menu principal	9
Figura II - Início do jogo	9
Figura III - Ecrã de Game Over	10
Figura IV - Jogo a decorrer	10

1. Introdução

O tema deste projeto é implementar o jogo Tetris em 2D, e para isso utilizámos a linguagem *C++*, e as respetivas bibliotecas: *OpenGL*, *GLFW*, *GLEW* e *GLM*.

Este jogo consiste em empilhar blocos dentro de um tabuleiro retangular, quando uma linha do tabuleiro se forma, esta é eliminada e tudo que se encontra acima desta desce uma linha. Os pontos são calculados consoante o número de linhas eliminadas.

O utilizador tem a possibilidade de visualizar a próxima peça que irá surgir e também a possibilidade de guardar a peça atual no “*HOLD*” para usar posteriormente.

A partida é dada como terminada quando as peças intersetarem o topo do tabuleiro.

2. Motivação

A motivação e a satisfação são variáveis importantes no desenvolver de um trabalho, sendo assim o que nos motivou na realização do mesmo foi o interesse pelo tema abordado por todos os elementos do grupo, também o facto de ser um jogo tradicional e conhecido, de forma geral, pela maioria da população e por ter sido um jogo presente na nossa infância libertou em nós a motivação para a realização do mesmo.

3. Tecnologias Utilizadas

OpenGL:

É o principal ambiente para o desenvolvimento de aplicativos gráficos 2D e 3D portáteis e interativos. Desde a sua introdução em 1992, o *OpenGL* tornou-se a interface de programação de aplicativos gráficos 2D e 3D mais utilizada e suportada da indústria (*API*), trazendo milhares de aplicativos para uma ampla variedade de plataformas de computador. O *OpenGL* promove a inovação e acelera o desenvolvimento de aplicativos ao incorporar um amplo conjunto de renderização, mapeamento de textura, efeitos especiais e outras funções de visualização poderosas.

GLFW:

É uma biblioteca de código aberto e multiplataforma para desenvolvimento *OpenGL* no desktop. Ele fornece uma *API* simples para criar janelas, contextos e superfícies, recebendo entradas e eventos.

GLEW:

É uma biblioteca de carregamento de extensão, *open-source* e *cross-platform*. Fornece mecanismos com tempo de execução eficientes para determinar quais extensões *OpenGL* são suportadas na plataforma de destino.

GLM:

“*OpenGL Mathematics*” é uma extensão, baseado nas convenções de extensão *GLSL*, providenciando mais capacidades como transformações de matriz, quatérnios, tipos baseados em meio, números aleatórios, entre outros.

Visual Studio Code:

É um editor de código-fonte desenvolvido pela *Microsoft* para *Windows*, *Linux* e *macOS*. Ele inclui suporte para depuração, com controlo *Git* incorporado, realce de sintaxe, complementação inteligente de código, *snippets* e refatoração de código.

C++:

É uma linguagem de programação compilada multi-paradigma e de uso geral. Desde os anos 1990 é uma das linguagens comerciais mais populares, sendo bastante usada devido ao seu grande desempenho e base de utilizadores.

Git:

É um sistema de controlo de versões distribuído, usado principalmente no desenvolvimento de *software*, mas pode ser usado para registar o histórico de edições de qualquer tipo de arquivo.

Trello:

Ferramenta de colaboração que organiza os projetos em ”*boards*”. O Trello informa o que está a ser realizado e produzido, quem está a trabalhar em qual das tarefas, o que está em processo e o que já foi finalizado.

4. Etapas de Desenvolvimento

Inicialmente começamos por definir as tarefas que cada um iria executar usando a plataforma “*Trello*”.

De seguida decidimos começar por criar a janela para ser usada para mostrar o nosso jogo. Começamos por planear as coordenadas para o tabuleiro de jogo e os quadrados para termos a peça “*NEXT*” e “*HOLD*”. Tratamos de desenhar estes no ecrã.

Criámos dois “*arrays*” que continham as coordenadas para todas as peças e para todas as rotações das mesmas (“*piece_vertex_buffer_data[7][4][4*6*3]*”). Também criámos um “*array*” que contém todas as posições do tabuleiro que estão ocupadas com 1 e as que estão livres com 0, o que nos permite desenhar os lugares ocupados do tabuleiro. Tratamos das colisões, de gerar a peça seguinte e da funcionalidade de termos a peça “*HOLD*”.

Em seguida tratámos da função que verifica se existem linhas completas e se sim apaga-as aumentando a pontuação.

A seguir tratámos do ecrã de menu e finalmente adicionámos texto gráfico.

5. Descrição do funcionamento do Software

Ao iniciarmos o programa deparamo-nos com um menu com o nome do jogo (“*TETRIS*”) e duas opções, “*START*” e “*QUIT*”, onde o utilizador irá fazer uma escolha, utilizando as setas para cima e para baixo e para confirmar a sua escolha basta premir “*ENTER*”.



Figura I - Menu principal

Caso o utilizador escolha a opção "*QUIT*", a janela irá fechar e então o programa termina. Caso contrário é então inicializado o jogo onde o utilizador poderá ver o tabuleiro, a sua pontuação e diferentes peças como a "*NEXT*" que irá a jogo, a "*HOLD*" que consiste na peça que podemos guardar e utilizar mais tarde e por último a peça que está a ser usada no tabuleiro.

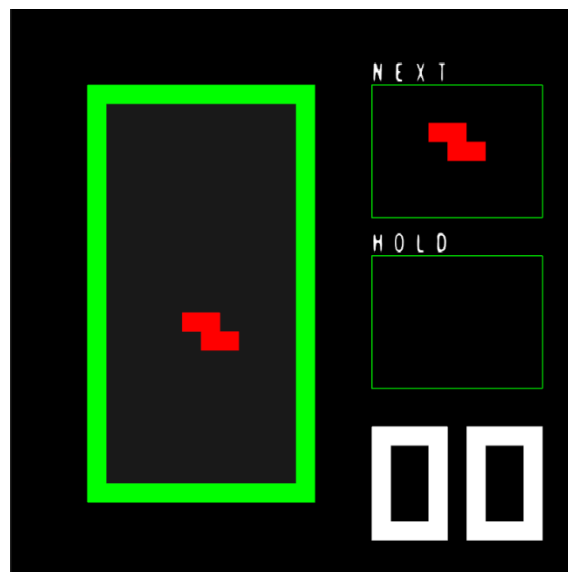


Figura II - Início do jogo

Em relação à peça que está dentro do tabuleiro, o utilizador consegue ainda manipular o seu movimento, como mover para a esquerda e direita utilizando as respetivas setas, rodar em sentido anti-horário clicando na seta para cima, aumentar a velocidade da peça para que esta desça mais rápido pressionando a seta para baixo e trocar a peça atual pela que está em “*HOLD*” através da tecla 'C' do teclado. O jogo acaba quando a peça que está na “*NEXT*” não tiver espaço para ser colocada no tabuleiro, dando assim o jogo como perdido e apresentando a mensagem “*GAME OVER*”.

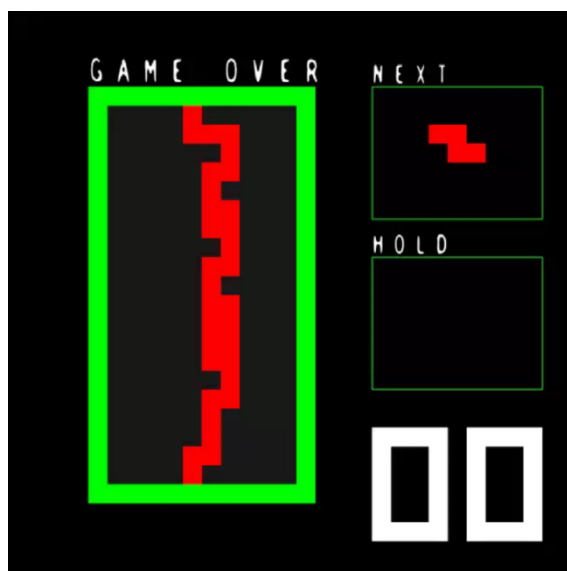


Figura III - Ecrã de Game Over

De seguida o utilizador pode clicar na tecla “*R*” para voltar a jogar, limpando e inicializando o tabuleiro, ficando preparado para jogar novamente, ou clicar em qualquer uma das outras teclas exceto o “*ESC*” e “*R*” para voltar ao menu. Para que o utilizador não perca, este terá de manipular as peças de forma a criar uma linha de peças consecutivas na horizontal, essa linha será eliminada do tabuleiro baixando uma posição tudo o que estava em cima da mesma, o número de linhas limpas vai ser adicionado à pontuação (que estará disponível para visualizar ao longo do jogo).

Caso o utilizador deseje sair do jogo a qualquer momento, basta premir a tecla “*ESC*” do seu teclado que irá fazer com que a janela se feche encerrando assim o programa.

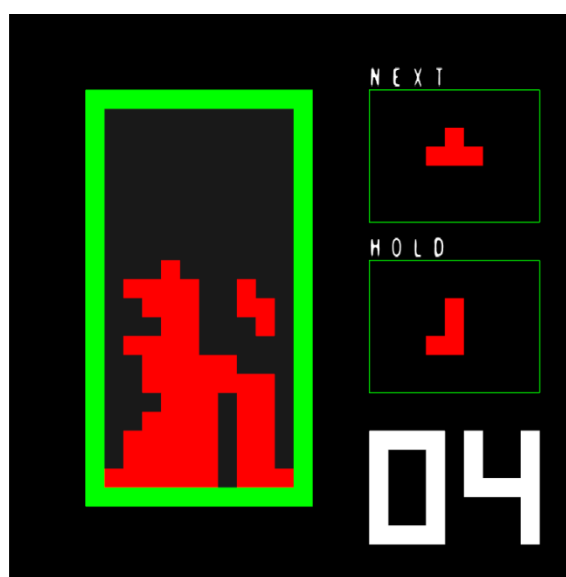


Figura IV - Jogo a decorrer

6. Trabalho Futuro

Futuramente, este jogo poderá ainda sofrer uma série de alterações relativamente às suas funcionalidades existentes, adicionando outras que não foram possíveis de implementar por falta de tempo.

Gostaríamos de ter usado uma fonte de iluminação no jogo, ter conseguido usar mais texturas nas peças de maneira a personalizar uma a uma e também ter conseguido passar o jogo para três dimensões.

Poderia ser pertinente e interessante a inclusão de sons e efeitos sonoros com o decorrer do jogo e também uma nova funcionalidade no menu que fornecesse a possibilidade de consultar o "*High Score*".

7. Considerações Finais

O presente projeto teve como objetivo o desenvolvimento do jogo Tetris 2D, de modo a ser perceptível e de fácil compreensão/utilização para todos os possíveis utilizadores.

O jogo que produzimos vai ao encontro do Tetris tradicional correspondendo a todas as funcionalidades e sensações que esperamos encontrar ao jogar o Tetris. De uma forma geral, é possível considerar que o jogo está funcional e pronto a ser utilizado.

Referência Bibliográfica

Gomes, A. (5 de 12 de 2020). *Computação Gráfica*. Obtido de Abel J.P. Gomes's homepage:
<http://www.di.ubi.pt/~agomes/cg/>
learnopengl. (20 de 12 de 2020). *Shaders*. Obtido de learnopengl.com:
<https://learnopengl.com/Getting-started/Shaders>
media, t. i. (6 de 1 de 2020). *C++ OpenGL Tutorial - #11 2D text*. Obtido de YouTube:
https://www.youtube.com/watch?v=t5V9QtJI6Ao&feature=emb_logo