

# WebGL Maze 3D

Fábio Alves e Sérgio de Aguiar

**Resumo** - O presente trabalho consiste num labirinto 3D, desenvolvido em WebGL, no qual foram utilizados vários conceitos de Computação Visual. Exemplos destes são criação de modelos 3D, aplicação de texturas e iluminação.

Foram também implementadas funcionalidades para permitir a interação com os modelos bem como analisar colisões, realizar mudanças nas texturas dos diversos modelos e ter uma fonte de luz móvel em frente ao jogador (cubo).

Todas as transformações 3D são realizadas através das teclas direcionais do teclado (setas) para movimento, e dos números 1 a 3 para alterar texturas dos vários modelos presentes.

Por fim, foi também implementada uma mudança na cor de fundo ao longo do tempo para aumentar a sensação de imersão do jogador ao simular a passagem do tempo.

**Abstract** - The present assignment consists of a 3D labyrinth, developed in WebGL, in which a variety of Visual Computation concepts were used. Examples of these are the creation of 3D models, texture application and illumination.

Features have also been implemented to allow for interaction with the models as well as analyze collisions, make changes in the textures of the various models and have a mobile light source in front of the player (cube).

All 3D transformations are performed using the keyboard's directional keys (arrows) for movement, and the numbers 1 through 3 to alter the textures of various models present.

Lastly, a change in background color over time has also been implemented to increase the player's sense of immersion by simulating the passage of time.

## I. INTRODUÇÃO

No âmbito da Unidade Curricular de Computação Visual (do 1º semestre do 4º ano) do Mestrado Integrado em Engenharia de Computadores e Telemática, foi-nos pedido que criássemos uma aplicação 3D em WebGL (uma API em JavaScript que permite a renderização 2D e 3D) que demonstrasse a nossa capacidade de aplicação dos conceitos lecionados nas aulas teórico-práticas e desenvolvidos nas aulas práticas.

Face a este requisito, o nosso grupo optou pela realização de um “Labirinto 3D”, opção esta que se encontrava na lista de temas de projetos propostos pelo professor da unidade curricular. Esta aplicação possuía por base alguns requisitos sendo estes a construção do labirinto em si, mecânicas que permitissem controlar o movimento do jogador e deteção das colisões com as paredes do labirinto.

Esta aplicação, envolvendo a navegação de um labirinto, implica a visualização das várias paredes do mesmo de

vários ângulos, sendo assim o tema ideal para um projeto que queira demonstrar o uso de iluminação e o seu efeito em materiais aplicados.

Desta forma, ao longo deste documento iremos especificar os módulos construídos e os adaptados do código fornecido nas aulas práticas acerca de diversos temas (modelos 3D, texturas, interação com os modelos, fontes de luz e materiais), o porquê das escolhas efetuadas, bem como algumas demonstrações visuais dos resultados provenientes do uso desses módulos.

## II. BASE E MODELOS 3D

Na realização deste projeto foram utilizados alguns ficheiros base fornecidos nas aulas práticas de forma a acelerar o desenvolvimento e realizar algumas funções básicas necessárias ao bom funcionamento do projeto como um todo, sendo eles o ficheiro “webgl-utils.js”, que foi usado para não ser necessário definir o contexto manualmente, o ficheiro “math.js”, onde se encontram diversas funções matemáticas usadas ao longo do projeto para os mais diversos cálculos, e ainda o ficheiro “models.js”, onde se encontra a função de cálculo de vetores normais utilizada para permitir o uso de fontes de luz.

A cor aplicada inicialmente ao background é azul, como pode ser visto na figura Fig. 1. Esta, contudo, oscila entre azul e amarelo, processo este descrito nos capítulos seguintes.



Fig. 1 - Posição inicial do jogador.

O modelo de projeção utilizado foi de perspetiva uma vez que este projeto é efetivamente um jogo em primeira pessoa. Este modelo permite aumentar a imersão do jogador ao providenciar uma experiência mais realística.



postos de forma que o centro da matriz se localiza sempre na posição (1.5, -1.5, -16).

Por fim, o "chão" é composto por um modelo de cubo base com alteração na escala do x para 31 e do z para 33, de forma a não só acomodar o array mas também uma unidade fora do array no início e outra no fim, a posição inicial é igual à posição referente ao centro do array e, por fim, o material escolhido foi cobre para intensificar as texturas que possuam bastante vermelho na composição da imagem, tal como visto na figura Fig. 5.



Fig. 5 - Chão do labirinto com textura maioritariamente vermelha.

### III. TEXTURAS

O ficheiro para carregar as texturas foi baseado no código fornecido nas aulas práticas.

Após o carregamento, as texturas são associadas às variáveis "texture" usando TEXTURE0 + n, sendo "n" equivalente ao número de texturas subtraído de 1, facilitando desta forma o carregamento de várias texturas a serem usadas.

As imagens usadas nas texturas, mostradas na figura Fig. 6, quase na sua totalidade tem licença de domínio público e as que não tem, é apenas pedido que sejam referenciadas as fontes nos créditos, o que acontece no final deste documento e no ficheiro "sources" na pasta "Textures".



Fig. 6 - Texturas base[1].

O adição de texturas novas é simples, sendo apenas preciso colocar um ficheiro com extensão ".gif"

com uma imagem, sendo esta imagem quadrada e o seu tamanho uma potência de 2, na pasta "Textures" e no ficheiro "LoadTextures.js" colocar o nome da imagem sem a extensão no array "fileNames".

### IV. ILUMINAÇÃO E ANIMAÇÃO

A cor do background é alterada a cada 250ms e varia entre azul e amarelo. Esta alteração é gradual descendente e ascendente, e varia entre 2 valores predefinidos.

A contagem do tempo é feita na função "countFrames" que foi baseada numa função fornecida nas aulas práticas.

Para a criação de fontes de luz foi usado código baseado também no fornecido nas aulas, porém adaptado para as necessidades do projeto em questão. Desta forma, cada fonte de luz tem 4 propriedades fundamentais sendo estas: posição, cor/intensidade, intensidade ambiente e rotação.

Para este projeto foram utilizados 2 focos de luz: um no centro do labirinto a uma altura de 50 para que todo o chão do labirinto seja iluminado, sendo que esta varia a sua intensidade à medida que a cor de background muda; e outro imediatamente à frente do "player", como pode ser visto nas figuras Fig. 7 e Fig. 8. Este segundo está sempre à frente do "player" independentemente dos movimentos que este faça.

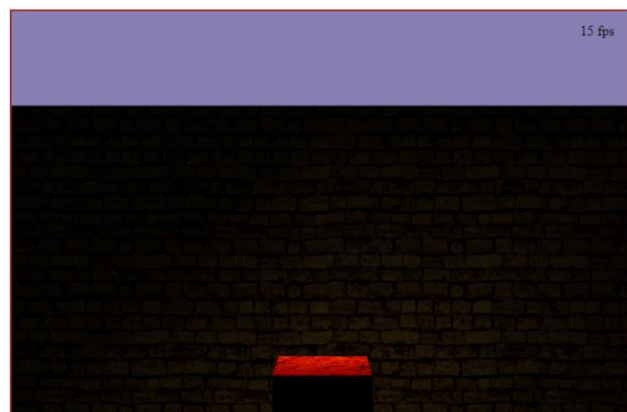


Fig. 7 - Cubo próximo de uma parede.

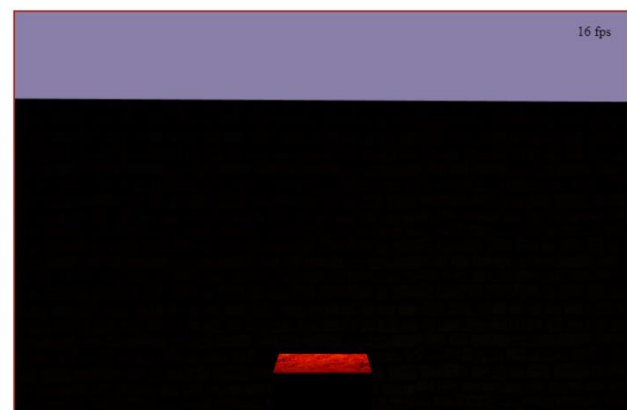


Fig. 8 - Cubo com face dianteira encostada a uma parede.



## V. INTERATIVIDADE / CONTROLO

O movimento do "player" é feito usando translações e rotações globais uma vez que o "player" se encontra fixo na posição central do eixo usado para transformações globais, dando assim a ilusão do player se mexer.

Desta forma, o movimento do player é feito através de uma translação tendo em conta o ângulo global. Por exemplo, se for 90° ocorre uma translação de 0.1 no eixo ZZ; se for 45° a translação será de 0.05 no eixo ZZ e no eixo XX exemplo deste ultimo na figura Fig. 9.

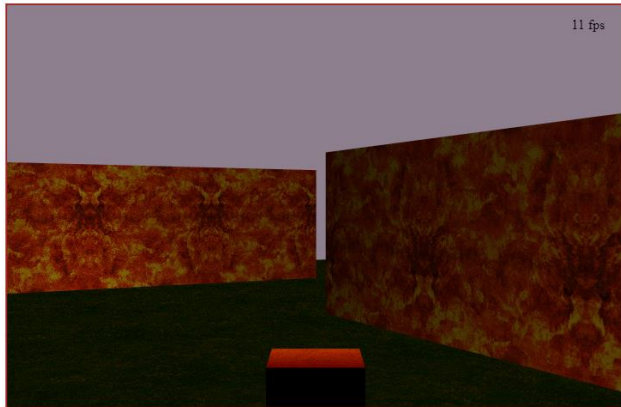


Fig. 9 - Ângulo de movimento 315°.

As rotações são controladas através das teclas direcionais do teclado "ARROW LEFT" e "ARROW RIGHT", e as translações através das "ARROW UP" e "ARROW DOWN".

A manipulação das texturas dentro do jogo são feitas usando as teclas numéricas 1 (floor), 2 (wall) e 3 (player), não sendo permitidas texturas iguais em objetos diferentes.

As colisões são detetadas fazendo uso da matriz "maze" previamente enunciada, vendo em que posição desta o centro do cubo se encontra, através do valor das translações globais, arredondando sempre para baixo. Inicialmente, o "player" começa fora da área coberta pela matriz. Como tal, foram feitas limitações na área que o "player" pode alcançar (valores máximos e mínimos de tx e tz globais) mostrado na Fig. 10 e no momento em que esta entra na matriz os dados desta começam a ser tidos em conta.

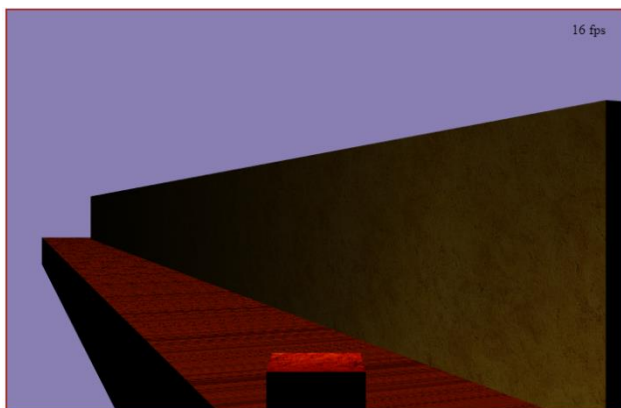


Fig. 10 - Posição limite fora da matriz.

Por fim, foi feito um ajuste para ver se a posição central do cubo +/-50% comprimento deste se encontra dentro de uma parede, evitando assim que o cubo atravessasse uma parede.

## VI. CONCLUSÃO

Com o desenvolvimento deste projeto, pudemos verificar que dezenas de blocos tem um peso computacional elevado, o que no nosso caso levou a que tivéssemos de reduzir o tamanho do labirinto e usar apenas cubos. Mesmo assim, o numero de fps foi relativamente reduzido. Uma possível solução para este problema seria calcular quais objetos estariam no campo visível do jogador, contudo isto poderia trazer outros problema associados a tempos de execução se mal implementado.

O projeto permitiu-nos ainda aprofundar os nossos conhecimentos nas áreas de computação visual e WebGL.

A divisão do trabalho foi 50% para cada um dos membros.

## REFERÊNCIAS

### [1] Wood:

Luke.RUSTLTD – 2012 – Disponível em: <https://opengameart.org/node/8719> . Acesso em: 25 Nov. 2019

### [1] Dirt:

Bart Kelsey – 2012 – Disponível em: <https://opengameart.org/node/11952> . Acesso em: 25 Nov. 2019

### [1] Grass:

Bart Kelsey – 2012 – Disponível em: <https://opengameart.org/node/7320> . Acesso em: 25 Nov. 2019

### [1] Red Sand:

Iwan Gabovitch – 2012 – Disponível em: <https://opengameart.org/node/7999> . Acesso em: 25 Nov. 2019

### [1] Galvanized Blue:

Bart Kelsey – 2012 – Disponível em: <https://opengameart.org/node/7477> . Acesso em: 25 Nov. 2019

### [1] Yellow Rock:

Clint Bellanger – 2012 – Disponível em: <https://opengameart.org/node/7887> . Acesso em: 25 Nov. 2019

### [1] Grass1:

Bart Kelsey – 2012 – Disponível em: <https://opengameart.org/node/7539> . Acesso em: 25 Nov. 2019

**[1] Lava:**

Bart Kelsey – 2012 – Disponível em:  
<https://opengameart.org/node/7763> . Acesso em: 25 Nov.  
2019

**[1] Wood floor:**

Bart Kelsey – 2012 – Disponível em:  
<https://opengameart.org/node/11950> . Acesso em: 25 Nov.  
2019

**[1] Brick:**

Bart Kelsey – 2012 – Disponível em:  
<https://opengameart.org/node/7506> . Acesso em: 25 Nov.  
2019

**[1] Brick1:**

Bart Kelsey – 2012 – Disponível em:  
<https://opengameart.org/node/7276> . Acesso em: 25 Nov.  
2019

**[1] Water:**

Iwan Gabovitch – 2012 – Disponível em:  
<https://opengameart.org/node/10593> . Acesso em: 25 Nov.  
2019

Joaquim Madeira - 2019 - Disponível em:  
<http://sweet.ua.pt/jmadeira/WebGL/> . Acesso em: 1º  
semestre, ano letivo 2019/2020.