3D Maze

Afonso Guimarães, Vasco Cardoso

***Resumo* – Este projeto foi proposto no contexto da disciplina de Computação Visual e tem com o principal objetico aplicar as técnicas aprendidas e desenvolvidas al longo das aulas de WebGL. O projeto desenvolvivo consiste num Labirinto 3D, com efeitos de iluminação e hipótese de escolher diferentes mapas.**

***Abstract* – This project was proposed in the context of Visual and Computing subject and its main purpose was to apply the techniques, learnt and developed, during the classes. The project consist on a 3D Maze where the user has to find the end and can choose different maps to traverse.**

I. INTRODUÇÃO

Na unidade curricular de Computação visual foi proposto o desenvolvimento de um dos projetos sugeridos pelo docente. O nosso grupo decidiu escolher o projeto do labirinto 3D para demonstrar conhecimentos na área de Computação Visual.

O jogo do Labirinto 3D foi um dos primeiros jogos a serem desenvolvidos e a serem jogados em rede, é um jogo não muito complexo mas nos primórdios da internet, devido ao alto tráfego produzido, teve de ser proibido de jogar.

O jogo apresentado tem como objetivo a descoberta do fim do labirinto, facilmente reconhecível pela sua cor destinta.

I. CONCEÇÃO

Foram usados vários blocos com os quais trabalhamos durante as aulas práticas, e que serviram como base de desenvolvimento do jogo, dos quais o módulo de aplicação de operações sobre matrizes e vetores (maths.js), o módulo de manipulação de focos de luz (lightSources.js) , o módulo de WebGL Utils (webgl-utils.js) e o módulo de inicialização dos shaders (initShaders.js) também foi usadofigures in the middle of the page.

Um dos exemplos das aulas também foi usado para construirmos o módulo de controlo da lógica do jogo e inicialização dos componentes gráficos (3dMaze).

III. ESTRUTURA DE DADOS

De forma a ter o código o mais modular possível e assim conseguir fazer alterações e adicionar novos requisitos foram adicionadas algumas variáveis:

- A estrutura de dados ‘Maze’ foi criada para conter o array que permite o desenho do mapa e permite guardar variáveis sobre o mesmo para obtenção de dados mais eficientemente.
  - Map [] : é um array multidimensional que contém três valor, zero, um e dois. O valor zero significa que nesse local estará um chão. O valor um diz-nos que lá estará uma parede. Uma parede é um rectângulo com altura ajustada a transparecer ser uma parede.
  - Height: é o valor do comprimento do array e é utilizado principalmente para obter o chão.
  - Width: é o valor da largura do array.
  - NOTA: para obtenção do mapa conforme o que foi desenhado, começando de baixo para cima foi escolhido inverter o array e percorrer esse array invertido, assim obtém-se tal e qual o que foi desenhado.
- A estrutura de dados SceneModels contém todos os modelos instanciados para desenho das figuras. Esta estrutura é preenchida através de um array que percorre todas as posições do array ‘Map’ na estrutura ‘Maze’ e constrói consoante o valor ser parede ou fim. Mais detalhes serão encontrados em “Elementos do Jogo”.
- No documento principal do trabalho temos a estrutura de dados que permite alterar a posição do “personagem”. As variáveis GlobalTx, GlobalTy, GlobalTz alteram a sua posição enquanto que o GlobalAngleZZ

altera o seu ângulo de visão. De forma a agilizar a alteração de algumas destas variáveis temos as funções :

- resetPosition(): utilizada para dar reset à posição do "personagem".
- seeFromAbove(): que permite ter uma visualização por cima do mapa mas com a consequência de voltar à posição inicial para assim tornar mais dificil.

IV. ELEMENTOS DO JOGO

Todos os elementos do nosso projeto se baseiam em cubos que podem ou nao sofrer alterações com o intuito de servirem os nossos interesses para o jogo. em seguida segue uma descrição destes elementos:

i) Chão

O chão do nosso labirinto é um único cubo que sofre alterações aos seus lados por forma a preencher toda a base do labirinto, tornando-se, na pratica, num paralelepipedo. sendo os seus lados a largura e profundidade do array que contém a informação do labirinto.

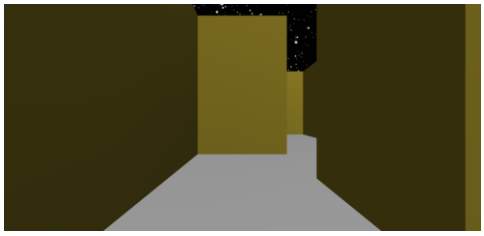
O chão tem os seus valores "KDiff", "KAmb", "KSpec" e "NPhong" alterados de forma a dar características visuais de material cinzento a este elemento.

ii) Paredes

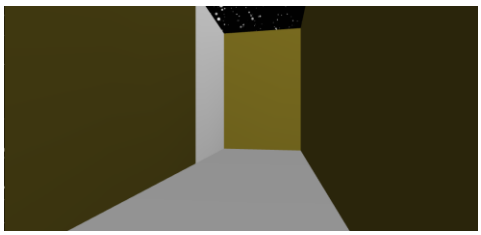
As paredes do nosso labirinto sao geradas individualmente gerando cubos na posição destas paredes aprendida atravez do mesmo array falado no ponto acima e desta vez a sua altura é alterada apenas para criar uma sensação de maior grandiosidade do labirinto para o jogador. Estas têm os seus valores "KDiff", "KAmb", "KSpec" e "NPhong" alterados de forma a dar carateristicas visuais de ouro polido a este elemento.

iii) final

A Parede que representa o final do labirinto é formada da mesma forma e com as mesmas porporções que as paredes do labirinto estando a diferença da sua cor especifica que torna obvio para o jogador que se trata do final do labirinto. Esta tem os seus valores "KDiff" alterados de forma a dar a cor branca a este elemento.



IMG1: ELEMENTOS CHÃO E PAREDES DO LABIRINTO



IMG2: ELEMENTOS CHÃO, PAREDES E FINAL DO LABIRINTO

iv) Colisões

Para obter as colisões e dado que o que muda são as posições globais e não todo o labirinto fizemos o segunte.

1. Calculamos com arrendondamento o bloco em que nos encontramos (dado que cada bloco instaciando traduz uma posição do array) e verificamos se com o movimento pretendido pelo utilizador o local que nos tentamos deslocar é válido. É de realçar que o utilizador se desloca em dois eixos por isso, é calculado o quanto anda consoante as regras de trigonomia. Caso esse movimento seja inválido as posições do "personagem" não se alteram ficando parado.
2. A colisão com o final leva a que seja mostrado um aviso de que isso sucedeu e permite então recomeçar.

V. MODELAÇÃO

Ainda que haja um variedade enorme de blocos que podem ser instaciados com WebGL, como foi dito anteriormente todo o nosso projeto se baseia em cubos e paralelipípedo.

Para que um cubo possa ser instanciado é usada a função *initBuffers()* que contém os buffers das coordenadas e dos vetores normais dos blocos.

VI. TRANSFORMAÇÕES

As transformações aplicadas permitem obter a visualização em primeiro pessoa por parte da personagem e também obter uma visualização global de todo o mapa.

- Para obter as rotações do “personagem” foi feito uma multiplicação da matriz de translação do centro (0,0,0) com a rotação no eixo. De seguida, para o colocar na posição certa em teve de se fazer uma multiplicação dessa matriz obtida anteriormente com a real posição do personagem obtida através da matriz de translação com as posições GlobalTx, GlobalTy e GlobalTz.
- Para obter a visão geral coloca-se o visualizador numa posição mais elevada e com um ângulo adequado, assim da mesma forma são feitas as multiplicações necessárias com a matriz mvMatrix.

VII. SONS

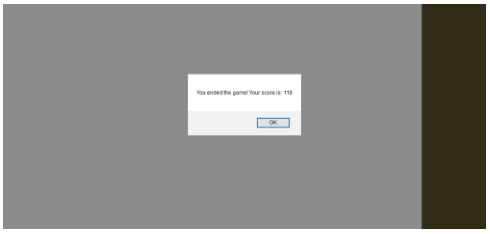
No projeto decidimos introduzir uma musica para criar ambiente enquanto o jogador precorre o labirinto, esta musica intitula-se de “Dark Clouds” do autor -MAZE-. Em seguida pomos um link com a musica:
HTTPS://WWW.JAMENDO.COM/TRACK/1053845/DARK-CLOUDS

VIII. PONTUAÇÃO

A pontuação no nosso jogo é mais alta quanto mais rapido o jogador terminar o labirinto, com isto um jogador que se engane menos no percurso do labirinto é recompensado em relação a um que se engane duranrte o processo de procura do fim do labirinto. Esta pontuação é demonstarda ao jogador enquanto este precorre o labirinto e a pontuação final é apresentada ao jogador atraves de um pop-up quando este entra no bloco do final do labirinto.

Enquanto este precorre o labirinto e a pontuação final é apresentada ao jogador atraves de um pop-up quando este entra no bloco do final do labirinto.

IMG3: PONTUAÇÃO APRESENTADA AO JOGADOR NO DECORRER DO JOGO(RODEADA A VERMELHO)



IMG4: PONTUAÇÃO APRESENTADA AO JOGADOR NO FINAL DO JOGO (ATRAVES DO POP-UP)

IX.CONTROLOS DE JOGO

No nosso projeto o “personagem” é controlado através das teclas “W”, ”A”, ”S”, ”D”, sendo as “W” e “S” para locomoção, permitindo-nos mover em frente e para trás respetivamente, a tecla “A” permite-nos rodar para a esquerda e a “D” para a direita, podendo através destas teclas percorrer todo o labirinto.

X.CONTRIBUIÇÃO INDIVIDUAL

O trabalho foi executado igualmente por ambos os membros do grupo, sendo que estiveram presentes em todas as fases de desenvolvimento do projeto, resultando assim numa participação de 50% para cada.

XI.INFORMAÇÃO EXTRA

O nosso projeto permite a alteração do labirinto, no entanto apenas conseguimos fazer isto alterando o valor no código, tentamos implementar esta feature no entanto surgiram alguns problemas e não o conseguimos implementar como prentendiamos, permitindo ao utilizador selecionar qual o labirinto que pretendia percorrer.

É de realçar que, por parte dos executantes do projeto, o labirinto corre com mais frames em Linux do que em Windows. Por isso sugerimos vivamente que o faça.

Caso tenha interesse em ver um vídeo exemplificador do trabalho siga o link:
HTTPS://WWW.YOUTUBE.COM/WATCH?V=-_FYH1DQTVU

XII.CONCLUSÃO

A nosso ver, os objetivos principais do projeto foram atingidos pois conseguimos por em prática os conhecimentos obtidos nas aulas de WebGL de forma a construir o Labirinto e permitir a navegação do utilizador neste mesmo que eram os dois grandes e principais objetivos do projeto.

Parte do código utilizado como base para a execução deste projeto foi construído durante as aulas práticas, tendo sido necessária a adaptação de certos excertos e elementos de código para que este cumprisse os nossos desejos e ambições para o projeto. Resumindo pensamos ter cumprido o pretendido para este tema e assim realizamos um bom trabalho.

XII. REFERÊNCIAS

Música utilizada como background no jogo:
HTTPS://WWW.JAMENDO.COM/TRACK/1053845/DARK-CLOUDS
Exercicio “new” da aula 8 (Prof. Joaquim madeira).
HTTP://SWEET.UA.PT/JMADEIRA/WEBGL/WEBGL_AULA_08.ZIP