PacMan 3D

Computação Visual

Diogo Ferreira 76425 Pedro Martins 76551

Resumo **– Este projeto foi proposto no contexto da disciplina de Computação Visual e tem como principal objetivo aplicar as técnicas 'low level' de WebGL apresentadas durante as aulas. Optámos por desenvolver o famoso jogo PacMan, mas em 3 dimensões. Toda a jogabilidade básica do jogo é portada do original, mas também introduzimos um novo modo "difícil", em que apenas uma área restrita em torno do PacMan é iluminada.**

Abstract **– This project was proposed in the context of Visual Computing subject and its main purpose was to apply multiple techniques of lower level WebGL that were presented through out the classes. We chose to develop the famous game PacMan, in 3 dimensions. All of the basic gameplay remains the same as the classic game, and we also introduced an "hard mode", where we changed the lighting sources so that only a restricted area around PacMan is visible.**

I. INTRODUÇÃO

Na unidade curricular de Computação Visual foi proposto o desenvolvimento de um projeto em WebGL, seguindo a linha do que nos foi apresentando durante as aulas.

O PacMan é um jogo mundialmente conhecido e bastante famoso. Decidimos, portanto, adaptá-lo ao mundo 3D e, para além da jogabilidade clássica com os "famintos" fantasmas e o "super-mode", onde o PacMan se pode também alimentar dos mesmos, adicionar-lhe mais algumas funcionalidades extra, como a leitura de labirintos a partir de um ficheiro fornecido pelo utilizador, portais que teletransportam um personagem para um qualquer outro portal e a inserção de um novo modo de jogo ('hard mode'), onde apenas uma área restrita em torno do PacMan é iluminada e esse mesmo foco segue os seus movimentos. O objetivo do jogo, claro está, é fazer o máximo de pontos possível, alimentando-se das migalhas e fantasmas, quando em 'super-mode', e fugindo dos mesmos caso contrário.

II. 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*) e o módulo de WebGL Utils (*webgl-utils.js*).

O módulo de inicialização dos *shaders* também foi usado, mas foi alterado para permitir a aplicação de texturas em conjunção com iluminação, a modelos (sem o ter que fazer no CPU, aplicando assim a textura e iluminação diretamente na GPU) e a possibilidade de aplicação de um *spotlight*, que apenas é usado no 'hard mode'. Este *spotlight* parte de uma fonte de luz direcional e filtra todos os vértices em que a norma do vetor entre o vértice e o foco ultrapassa um determinado *threshold*.

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

III. ESTRUTURAS DE DADOS DO JOGO

De forma a organizar o código e ser facilmente alterado, optou-se por guardar todos os dados do campo num único objeto, que é acedido por todos os restantes módulos e funções. Este objecto é composto por:

- Estrutura do campo: matriz composta pelos vários tipos de blocos (f para *food*, s para *super-food*, w para *wall* e p para *portal*). Cada bloco é caracterizado pelo seu tipo (anteriormente descrito), coordenadas nos vários eixos (x, y e z) e uma lista de movimentos possíveis, tendo em conta os blocos vizinhos do mesmo;
- Largura e altura do campo/estrutura;
- Tamanho das arestas dos cubos (nos eixos XX e ZZ) que servirão de base para desenhar os vários blocos;
- Velocidade de movimento do PacMan e dos fantasmas.

De igual forma, foi criado um tipo específico para as personagens, já que os comportamentos são semelhantes. Para tal, uma personagem é caracterizada por:

- Coordenadas da personagem (componente X e Z), conforme os tamanhos da matriz da estrutura do campo (que são ajustadas para mapearem o campo de jogo aquando do desenho do modelo) ;
- Direção do movimento para eixos XX e ZZ (e.g., valor 1 no eixo XX significa que a personagem se moverá conforme o eixo cresce, -1 conforme ele decresce e 0 nao se moverá neste eixo);
- Tecla que corresponde a direcção atual do movimento, para que seja mais fácil mapear as direções, usando como identificador o valor da tecla;
- Próxima direção, de maneira a "agendar" movimentos, de forma a que seja mais fácil mover o PacMan na direção desejada;
- Bloco onde o PacMan está atualmente situado. Apenas é atualizado quando a sua posição é múltipla do tamanho da aresta do bloco, dado que isso significa que está paralelo ao bloco (já que os incrementos de movimento podem ser menores do que este);
- Identificador (nome) do personagem;
- Indicador de teletransporte, usado para identificar se o personagem se encontra em teletransporte, para que não esteja a ser novamente teletransportado aquando a sua chegada ao novo portal.

Com vista a auxiliar a gestão destes atributos, uma personagem tem ao seu dispor os seguintes métodos:

- *updateDirection*: "agenda" um movimento a ser executado quando possível, isto é, guarda-o no atributo da próxima direcção;
- *updatePosition*: atualiza a posição do personagem na matriz que define a estrutura do campo, com base no bloco atual e nas direcções que tomou.

IV. ELEMENTOS DO JOGO

A. Campo

O campo é um dos elementos mais importante do jogo, sendo que vários cálculos são feitos tendo como base a sua estrutura.

Numa fase inicial, a matriz do campo, por si, não nos permite criar um jogo completo, visto que não há noção de movimentos possíveis nem de colisões. Como tal, é necessário fazer uma pré-computação em torno da mesma, de forma a:

1. Tornar cada um dos blocos "endereçável", indicando as suas coordenadas e tipo, calcular o número de migalhas espalhada pelo campo, que permite determinar o término do jogo quando não houver mais comida e identificar quais os portais existentes (método *createFieldStructure*);
2. Identificar, para cada bloco presente no campo, quais os movimentos possíveis de ser executados, tendo em conta os seus vizinhos (método *computeAllMoves*).

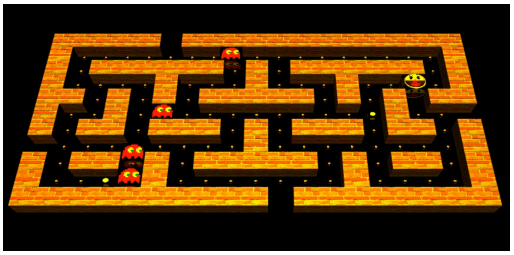

Fig. 1 - Campo de jogo.

B. Personagens

As personagens são os únicos objetos animados no jogo, uma vez que estas são capazes de se movimentarem ao longo do campo. Como tal, é necessário controlar estas movimentações, fazendo com que apenas sejam feitas com base nas direções possíveis. Como referido anteriormente, estes cálculos já foram efetuados previamente.

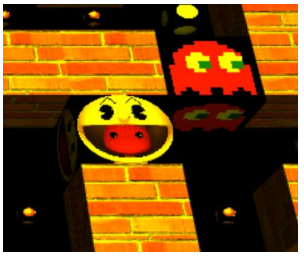

Fig. 2 - Personagens do jogo: PacMan e um fantasma.

C. Comida

A comida é um dos elementos principais que permite ao jogador aumentar a sua pontuação. Há, no entanto, dois tipos de comida:

- *food*, que diz respeito à comida simples e que aumenta 1 ponto no *score*;
- *super-food,* que permite que o PacMan entre em super-mode (aumenta 10 pontos), durante o qual poderá comer os vários fantasmas presentes no campo (cada um aumenta 100 pontos).

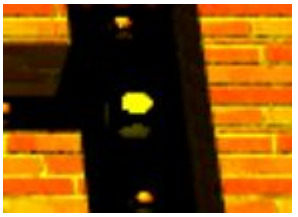

Fig. 3 - *Super-food* rodeada de *regular food*.

D. Portais

Os portais são uma inovação no estilo de jogo clássico do PacMan, sendo que, quando qualquer uma das

personagens estiver num deles, será teletransportado para um outro qualquer portal de forma aleatória, permitindo mudar de forma drástica as movimentações e localizações das personagens.

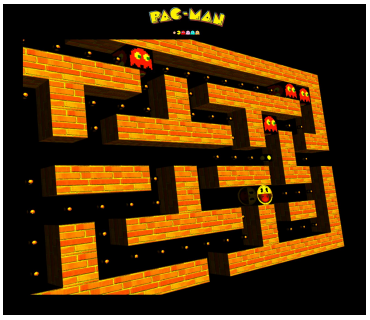

Fig. 4 - Portais (ao fundo à direita e em baixo).

V. MODELAÇÃO

Ainda que haja uma variedade de blocos que definem um campo de jogo, estes podem todos ser construídos com base no mesmo modelo elementar - o cubo.

A. Buffers

Para que um cubo possa ser instanciado, tendo em conta a biblioteca WebGL, será necessário criar 4 *buffers*:

- *Buffer* de coordenadas, para o qual são passados os vários vértices que permitem definir as faces do cubo;
- *Buffer* de índices, que permite definir as faces do cubo com base nos vértices do *buffer* anterior;
- *Buffer* dos vetores normais, necessário para o cálculo da iluminação da cena;
- *Buffer* de texturas, que permite que um modelo possa ser desenhado com uma determinada textura.

B. Transformações

As várias transformações aplicadas permitem dar o carácter dinâmico ao jogo, dado que as mesmas tornam possível os movimentos, rotações e escalas. No entanto, estas transformações podem ser:

- Locais: aplicadas a um modelo específico. Tanto no caso das personagens como dos blocos do campo, é aplicada uma translação relativa à sua posição (guardada nos seus atributos) ajustada para que este seja desenhado com base no centro do campo (o centro do campo é desenhado na posição (0,0,0)). Aos blocos do campo é ainda aplicada uma escala, dado que a *food* é a mais pequena, seguida da *super-food* e depois as *walls*;
- Globais: aplicadas a todos os modelos. É feita uma translação no eixo ZZ, para ajustar a posição fruto da vista em perspectiva, e são ainda aplicadas rotações, controladas pelo utilizador (via rato ou teclado).

Todo o jogo foi construído numa vista perspetiva e o modelo do campo está construído sobre o plano XoZ, portanto, é aplicada uma transformação inicial (rotação) a todos os modelos para que o campo seja visto frontalmente.

C. Texturas

Com vista a conferir aos modelos uma maior realidade, foram criadas texturas para cada bloco, de forma a refletir visualmente o seu tipo, sendo elas:

- Textura de tijolos, para as *walls;*
- Imagem de uma esfera vermelha, que representa a *food;*
- Imagem de uma esfera amarela, para a *super-food*;
- Imagem do clássico fantasma do PacMan;
- Imagem da face frontal do PacMan.

D. Iluminação

De forma a tornar a experiência de jogo melhor e mais realista, foi ainda adicionada iluminação, em conjunção com as texturas anteriormente referidas. Foram criados dois conjuntos de focos direcionais de iluminação, sendo eles:

- *Sun0* e *Sun1*, fixados x=0, y=5 e z=-1 e z=2, respetivamente. O objetivo destes dois focos é simularem a luz emitida pelo sol, proveniente de diferentes posições (para iluminar a parte lateral e cimeira do campo). Estes focos são apenas utilizados quando o jogo não está em *super-mode*;
- *LightRed*, fixado em x=0, y=5, z=0 que iluminará o campo quando o jogo estiver em *super-mode*, permitindo dar uma informação visual de que este está a decorrer.
- *LightBlue*, fixado na mesma posição do foco anterior e criado com o mesmo objectivo, no entanto este foco estará em constante rotação, conferindo um efeito semelhante ao da luz de um veículo da polícia.

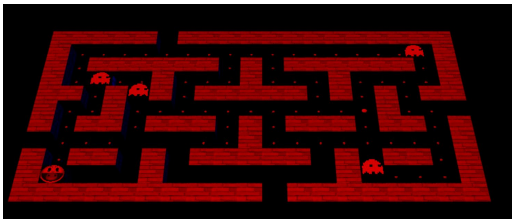

Fig. 5 - Momento do jogo com ‘super-mode’ ativo.

O comportamento dos focos acima descritos é totalmente alterado quando a dificuldade de jogo escolhida pelo utilizador for a mais elevada (‘hard mode’). Nesse caso, o campo de visão será notoriamente mais pequeno por forma a aumentar a dificuldade de detecção de comida e fantasmas, bem como o movimento dos focos

acompanhará o movimento do PacMan. Para alcançar este modo de iluminação, é ignorado a adição do contributo da iluminação em vértices cuja distância ao modelo do PacMan esteja acima de um determinado *threshold* definido de antemão.

É ainda pertinente realçar que todos estes cálculos são efectuados na GPU, diminuindo o tempo necessário para processamento dos cálculos, o que confere uma maior fluidez ao jogo.

Fig. 6 - Momento do jogo em 'hard-mode', onde apenas parte do campo é iluminado.

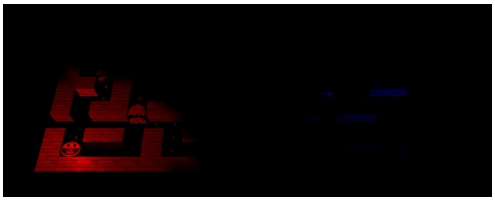

Fig. 7 - Quando selecionado o 'hard-mode' e ativado o 'super-mode', a área iluminada pelos focos aumenta ligeiramente.

V. INTERFACE

A. Instruções e campo personalizado

Ao iniciar o jogo, o utilizador é prontamente confrontado com uma breve apresentação das instruções gerais de jogo, bem como é dada a possibilidade do próprio utilizador fazer *upload* de um ficheiro contendo o campo de jogo. Este deverá conter uma estrutura que vá de encontro aos requisitos apresentados no início deste documento, estrutura essa que vai ser posteriormente validada para que não seja possível desenhar o campo nos moldes normais.

Depois da leitura de instruções e/ou *upload* do ficheiro com o campo, este será prontamente desenhado, bem como inicializadas as personagens em coordenadas aleatórias.

Fig. 8 - Página inicial do jogo, com um estilo bastante "retro".

B. Controlos de jogo

O utilizador poderá mover o PacMan através das setas do teclado, bem como rodar a câmara de jogo com o arrasto do rato ou com as teclas W e S (sobre o eixo XX) e A e D (sobre o eixo ZZ). Depois do jogo terminar (com uma vitória ou derrota), o utilizador poderá sempre reiniciar o jogo, clicando no respetivo botão.

C. Sons

Com vista a melhorar a experiência de jogo, foram ainda adicionados sons que caracterizam o contexto atual, sendo:

- *Intro*, quando o jogo é carregado;
- Ingerir uma *food*;
- Ingerir uma *super-food*;
- Ingerir um fantasma (apenas em *super-mode)*;
- Vitória;
- Derrota.

VI. INFORMAÇÕES EXTRA

O jogo foi testado nos *browsers* Microsoft Edge e Mozilla Firefox, no entanto, recomenda-se o uso deste último.

Como não é possível carregar texturas em alguns *browsers* como Google Chrome ou Apple Safari, não se recomenda o uso dos mesmos.

VI. CONCLUSÃO

O principal objetivo deste trabalho foi atingido, pois foi possível pôr em prática os conhecimentos adquiridos ao longo do semestre sobre computação gráfica, mais propriamente aplicada com a biblioteca WebGL. O facto de ter sido desenvolvido um jogo tão clássico como o PacMan permitiu-nos observar que todos estes conceitos podem ser aplicados sobre modelos do dia-a-dia e/ou reais.

Parte do código utilizado como base para a execução deste projeto foi construído durante as aulas práticas, tendo sido necessária a criação de toda uma base para a gestão do campo e movimentos. Foi ainda desafiante juntar o cálculo da iluminação às cores das texturas, conferindo uma maior realidade aos modelos, bem como o

desenvolvimento de uma lógica de organização das estruturas do jogo, assim como o controlo de colisões durante o mesmo.

VII. REFERÊNCIAS

A estruturação da lógica do jogo e do seu respetivo código, numa fase inicial, foi algo problemático. No entanto, o trabalho do Dionysis Lappas, que é proprietário de um repositório no GitHub (https://github.com/denlap007/pacmaze), foi uma boa ajuda, onde é apresentada uma implementação do PacMan com base também na biblioteca WebGL. A nível de interface com o utilizador, o repositório https://github.com/fpegios/PacMan3D serviu tambem de inspiração (sons e instruções). De notar que não foi copiado qualquer código destas duas implementações, tendo apenas servido de ideia inicial para o arranque do nosso projecto. Foram ainda consultadas algumas outras referências com vista a clarificar dúvidas e problemas que foram surgindo ao longo do desenvolvimento do projecto, nomeadamente:

- http://learningwebgl.com/blog/?page_id=1217
- https://webglfundamentals.org/webgl/lessons/webgl-shaders-and-glsl.html
- http://sweet.ua.pt/jmadeira/WebGL/ (material das aulas práticas)
- http://sweet.ua.pt/jmadeira/CV/index.html (material das aulas teóricas)