

Departamento de Engenharia Informática

Universidade de Coimbra

Computação Gráfica

Relatório Final

Christopher Jin Liu

Nº 2013150914

Daniela Gomes Loureço

Nº2013143670

Conceito

A ideia inicial foi alterada devido ao feedback recebido da primeira meta. Por isso, decidiu-se implementar um jogo de damas para existir mais interacção entre a aplicação e o utilizador.

Assim, a aplicação reproduz uma sala 3D, na qual se encontram vários objectos e um jogo de tabuleiro. São aplicadas as técnicas de iluminação, modelagem, texturas, animação, reflexão e sistemas de partículas.

Iluminação

O programa tem três fontes de iluminação, que estão definidas na função `initLights`. Uma é do tipo foco, que é posicionada junto do observador e aponta sempre para a frente. A outra é uma luz pontual que se encontra posicionada no centro do tecto, iluminando a sala inteira. Por último, existe também uma luz ambiente.

A luz foco não ilumina as paredes, pois o OpenGL só ilumina vértices, e as paredes só têm vértices nos quadros.

Objectos

Os objectos foram definidos com primitivas de OpenGL, a `GL_QUADS`. A sala tem quadros, uma janela, uma porta, dois suportes, duas latas, um tabuleiro e uma fonte.

Na função `window` são definidos os quadros, e na função `RoomBorder()` são definidas os tipos de parede. Ambas as funções podem ser ajustadas quanto aos parâmetros `x`, `y` e `z` da translação, aos parâmetros ângulo, `x`, `y` e `z` da rotação, e à textura a ser aplicada, para posicionar os quadros e as bordas.

A lata foi desenhada com as funções `gluDisk` e `gluCylinder`.

Os suportes são cúbicos e encontram-se definidos nas funções `Showcaselower()`, que desenha as faces laterais e inferior, e na função `ShowcaselowerTOP()`, que desenha a face superior. Ambas podem ser ajustados quando à sua posição. Os suportes encontram-se implementados em funções separadas devido à reflexão da lata na face superior.

Desenho das paredes da sala

A função `room`, é responsável pelo desenho das paredes laterais e do tecto. Nesta função são chamadas a função `floor`, que desenha o chão, e as funções com os objectos que preenchem as paredes.

O chão é desenhado no plano $y=0$, para posicionar mais facilmente os objectos, e o tecto é desenhado na coordenada $y=60$.

A porta e os quadros são desenhados diretamente sobre as paredes. A ordem pelo quais os objetos estão desenhados, não interessa, pois o z-buffer trata de desenhar os objectos mais próximos do observados.

Texturas

Utilizamos três tipos de texturas para as paredes, seis tipos para os quadros, uma para a porta, outra para chão, tecto e suportes. Em todos os objectos associamos os seus vértices aos da textura, para um correcto mapeamento. A função `criaDefineTexturas()` é responsável pelas definições das texturas.

Animação

Também existem duas animações: uma delas faz com que a lata esteja continuamente a rodar, e a outra, permite que a lata salte. Estas animações são feitas com a função `glutPostRedisplay()` presente na função `display()`, que funciona como um loop.

Transparência

A transparência encontra-se implementada na função `ShowcaseUpper()`. Esta função desenha uma caixa transparente, que permite a visualização da animação da lata que salta.

Reflexão

Implementámos uma reflexão da lata que se encontra em rotação, na função `Reflection`. Para isso, definiu-se no stencil buffer a forma da superfície refletora, que se encontra na função `ShowcaselowerTOP()`. De seguida, desenhou-se a lata refletida usando a máscara do stencil buffer e no final, desenhou-se a superfície refletora com transparência.

Observador

O observador só pode navegar dentro da sala. Os comandos para a sua deslocação do observador são os seguintes:

- setas **up** e **down**: permitem o deslocamento do observador para a frente e para trás, ou seja, o observador muda a sua posição(obs_x, obs_z), mas o ponto para onde está a olhar mantém-se fixo.

- setas **left** e **right**: permitem que o observador gire, assim o mesmo mantém-se na mesma posição e muda o ponto para onde está a olhar(tx, tz). Neste caso, o ângulo.

- teclas **w** e **s**: o observador gira em torno de um ponto, segundo o eixo dos yy .

- teclas **q** e **a**: permitem um deslocamento vertical (obs_y e ty a variar).

- teclas **e** e **d**: o observador mantém-se na mesma posição, alterando o ponto para o qual está a olhar(ty a variar).

Além disso, o observador também pode activar/desactivar diferentes modos com as seguintes teclas:

- tecla **r**: muda a textura das paredes da sala e activar/desactivar opção da `GL_LIGHTING`.

- tecla **f**: liga/desliga a luz foco.

- tecla **i**: liga/desliga a luz pontual.

- tecla **1**: activa/desactiva a animação do salto da lata.

- tecla **G**: activa/desactiva a câmara de jogo.

Jogo

No centro da sala encontra-se um tabuleiro que permite jogar o típico jogo de damas. Na implementação da movimentação das peças foi utilizado como fonte o site "<http://cjdvs.com/c-opengl-checkers/>", que nos permitiu perceber como seleccionar as latas e realizar a sua posterior deslocação. Assim, simulou-se uma visualização 2D com vista unicamente para o tabuleiro de jogo através da alteração dos parâmetros da função `gluLookAt`, para obter as posições das damas. Estas posições encontram-se numa matriz 8x8, a partir da divisão por 100 e 90, uma vez que o ecrã tem 800x700 pixels, e são obtidas com o botão esquerdo do rato. A matriz é inicializada com valores de -1, 0 e 1, nos quais o valor -1 representa as posições da Pepsi que correspondem às damas brancas; o valor 0 representa células vazias e o valor 1 representa as latas de Coca-Cola, que correspondem às peças pretas. A selecção de objectos também poderia ter sido realizada com o selection buffer ou com raycasting, mas optámos pelo método acima citado pela sua simplicidade e rápida implementação.

Para jogar, é necessário activar o modo da câmara de jogo a partir da tecla G. Para a movimentação das latas é preciso saber qual o turno que se encontra activo e seleccionar uma peça desse turno, com o botão esquerdo do rato e, de seguida, seleccionar a posição

para onde se pretende deslocar a lata. Se a câmara de jogo não estiver activada, não será possível seleccionar as latas. O turno da Pepsi é sempre o primeiro a iniciar o jogo. Caso o utilizador não saiba qual o turno em que se encontra o jogo, poderá sair da câmara do jogo ao clicar na tecla G, e, visualizar os quadros que se encontram no fundo da sala, que mudam de imagem de acordo com o turno de cada utilizador. Posteriormente, foi implementado a opção de Reset, utilizando o botão direito do rato que reinicializa a sala e as variáveis do jogo.

O objectivo do jogo consiste em chegar com pelo menos 4 peças à linha final do tabuleiro, do lado do campo do adversário. O primeiro jogador a concluir esse objectivo será o vencedor.

Sistema de Partículas

Por último, foram implementados três sistemas de partículas idênticos, que simulam uma fonte. A base da fonte é desenhada com vários cubos, já que a base tem a forma de um quadrado. Assim, foi criada a função `baseFonte()` que constrói cada cubo, e, pode ser ajustada quanto às posições, ao tamanho, à largura e textura dos cubos.

As partículas são inicializadas na função `iniParticulas()` e desenhadas na função `showParticulas()`. Para isso implementámos um loop que atualiza cada partícula e reutiliza as partículas. Como as partículas são desenhadas com a primitiva `GL_TRIANGLE_STRIP` em 2D, aplicou-se a técnica de billboard, na qual se roda a textura para estar sempre virada para o observador.