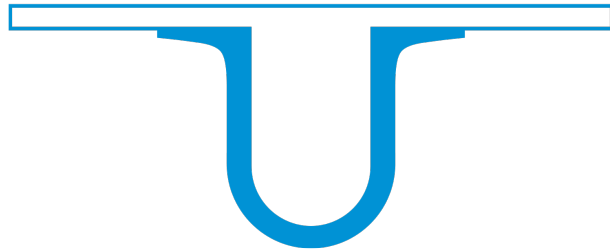


FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE
COIMBRA



Computação Gráfica

Relatório de projeto

2018/2019

Francisco Alves 2012136614 falves@student.dei.uc.pt

Madalena Santos 2016226726 mcsantos@student.dei.uc.pt

Introdução

De forma a cumprir com a proposta que foi feita aos alunos de Computação Gráfica no início do semestre - a aplicação dos conhecimentos adquiridos na cadeira, utilizando a framework OpenGL e o modelo de umas escadas, descrevemos de seguida as opções tomadas e implementadas por nós.

1. Transformações Geométricas

Nesta fase, desenhámos na nossa cena tudo o que seriam os objetos físicos. Desenhámos uma casa com janelas e portas, e o mobiliário existente dentro da casa. Foram criadas algumas funções como `drawCube(width, height, depth)` que cria um paralelepípedo com largura igual a *width*, altura igual a *height* e profundidade igual a *depth*; `desenhaDegrau(altura, comprimento, profundidade, textura, tipo)`, que desenha um degrau com as medidas dadas, e a textura especificada. A nossa casa é rodeada de escadas nas paredes, que são retratíveis. Ao clicar no botão R, estas recolhem, formando uma parede típica, ou então expandem-se, mostrando de novo as escadas.

Tentámos fazer uma animação que mostrasse este movimento, mas sem sucesso. Apenas conseguimos que as escadas estivessem ora na posição aberta, ora na fechada. No entanto, esta animação não seria fácil. O movimento começaria por recolher o primeiro degrau a contar de baixo, até que a sua posição se alinhasse com o degrau diretamente acima. De seguida, as duas primeiras escadas fariam exatamente o mesmo até se alinharem com a 3ª escada. Este processo seria dificultado tendo em conta que existem escadas em 3 lados da casa.

O telhado também é feito de escadas.

Fizémos uma pequena animação num quadro que gira continuamente ao longo do tempo, estabelecendo um ângulo de rotação incrementado ao longo do tempo, e através de translações e rotações do objeto.

2. Projeção e Visualização

Para simular o observador, utilizámos uma visualização em perspetiva, o que torna observação mais próxima do real. A direção do olhar é dada pelo movimento do rato, o que foi estabelecido através da função `glutPassiveMotionFunc()`, que executa uma outra função sempre que o rato é movido. Esta função estabelece o ângulo de visão, que é depois utilizado para calcular, através de fórmulas trigonométricas, os outros parâmetros da visualização. O movimento ao longo do mapa é feito através das teclas A, W, S, D.

3. Cor e Iluminação

Nesta fase, definimos 2 focos de luz na frente da casa, várias luzes pontuais dentro da casa, e uma luz ambiente. Cada objeto foi depois definido em função do material que queríamos representar, e à custa do header `materials.h`, que define as várias componentes para cada tipo de material. Nesta fase criámos ainda uma função para desenhar nevoeiro.

Na maior parte dos objetos que utilizam texturas, utilizámos materiais com uma cor branca, de forma a não influenciar a cor natural da textura, e ao mesmo tempo, tornar as superfícies reativas à luz.

4. Texturas

Utilizámos texturas para o desenho da skysphere, do chão, das escadas e paredes, das duas fontes, do altar, do quadro, do tapete, etc. Todas as texturas utilizam `GL_MODULATE`.

5. Sistemas de partículas

Nesta fase, fizemos uma lareira. Para simular o fogo, criamos várias partículas que têm uma certa posição, uma certa velocidade e um certo tempo máximo de vida, ao fim do qual desaparecem. Para isto, criamos uma estrutura de dados Particula, que guarda estes atributos.

Fizemos uso de funções das funções `initParticulas(Particula *)` e `showParticulas(particulas, x, y, z)`.

6. Transparências

Implementamos transparências nos vidros das janelas na frente da casa. Fizemos uso dos comandos `glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)`, que permite definir as propriedades necessárias, e o comando `glEnable(GL_BLEND)`, de forma a ativar a transparência destas superfícies no início da execução.

7. Reflexão

Nesta fase, foi criado um espelho. Primeiro, renderiza-se o mundo de forma invertida - utilizando uma escala com parâmetros $(-1, 1, 1)$. Fizemos uso do stencil buffer, de forma a limitar os pontos renderizados pelo OpenGL. De seguida é desenhado o objeto do espelho, onde são também definidas as propriedades do material. Depois, renderiza-se a cena "normal", sem ser invertida. Começamos por desenhar o espelho na origem. Para o tirarmos de lá e o colocarmos numa das paredes do interior da casa, teríamos que transladar também a cena, de acordo com a reflexão feita. Tivemos algumas dificuldades em ajustar os parâmetros desta translação de forma a que a reflexão coincidisse com aquilo que era suposto, e portanto decidimos deixar o espelho na origem.