

Faculdade de Ciências e Tecnologia
Computação Gráfica e Interfaces
2013/2014

Autores: 35469 Eduardo Lopes
42038 Ricardo Gaspar

Relatório:
Trabalho Prático 2 – Visualizador de
Objectos 3D

O trabalho realizado consiste num programa em Java Swing/JOGL que permite a visualização de objectos 3D através de várias projecções cujos parâmetros são controlados pelo utilizador.

A programação de aplicações gráficas é baseada em eventos, isto é, quando ocorre um evento de um determinado tipo esse evento tem de ser tratado. Assim, para se utilizar a biblioteca JOGL foi necessário implementar a classe `GLEventListener` à qual se deu o nome de `VisualizadorObj3D`. Por conseguinte, foram implementados os métodos: `init`, `display`, `reshape`, `displayChanged`.

Nesta classe, estão ainda definidos os seguintes métodos auxiliares: `desenhaCubo`, `desenhaObjecto`, `aplicaTextura`, `realizaProjeccao`, `calculaTetaInicial`, `calculaGamaInicial`, `criarPainelControlo`, `criarMenuFicheiro` e `criarMenuVer`.

Além desta, foram ainda definidas as classes `Vertex`, `Face` e `ObjectLoader`. A classe `Vertex` define as coordenadas de um vértice no espaço tridimensional. A classe `Face` define estruturas de dados para guardar os vértices e os vértices-textura que caracterizam uma face do objecto 3D. Já a classe `ObjectLoader` é responsável pelo carregamento do objecto em memória através de estruturas de dados próprias.

Breve descrição dos métodos da classe `VisualizadorObj3D`:

Init – Método que é chamado pelo JOGL na inicialização da informação gráfica aquando da execução inicial do programa. Aqui activou-se o Z-Buffer utilizando a chamada `glEnable(GL_DEPTH_TEST)`.

Display – Método responsável por actualizar o que é mostrado na área de desenho. Actualizar a área de desenho implica apagar tudo e redesenhar. Neste são carregadas as matrizes projecção consoante a projecção a realizar, bem como, as matrizes de modelação. Neste, são chamados os métodos `realizaProjeccao` e `desenhaCubo`. É também aqui que é aplicado o zoom e o ajuste do objecto apresenta à janela. É conseguido através da chamada da função `glOrtho`.

Reshape – Método chamado quando a janela é redimensionada. Quando ocorre o redimensionamento da janela é necessário ajustar o visor. Além disto guardam-se as novas dimensões da janela (largura e altura) para, no método `display`, se poder fazer o ajuste do objecto na área de desenho. Mais concretamente, manter o desenho ao centro e evitar que este seja deformado com o redimensionamento.

DisplayChanged – Método simplesmente implementado para obedecer à interface `GLEventListener`. Não se realiza qualquer operação neste método.

RealizaProjeccao – Consoante a projecção seleccionada pelo utilizador é realizada a projecção correspondente. Existem matrizes já definidas para cada projecção para serem aplicadas à matriz identidade quando se está a tratar da matriz projecção, isto é, após a execução de `gl.glMatrixMode(GL.GL_PROJECTION)` e antes da fase de modelação do objecto.

DesenhaCubo – Quando o utilizador selecciona a opção de boundingbox é desenhado um cubo de aresta 1. Este cubo é desenhado utilizando uma ferramenta do JOGL chamada `GLUT` – (GL Utility Toolkit).

AplicaTextura – Para que uma dada textura seja aplicada é necessário instruir o JOGL para esta tarefa. Este método prepara a textura e indica ao JOGL como se deseja que a textura seja aplicada.

DesenhaObjecto – Modela e desenha o objecto carregado em memória. Este método é executado quando o programa se encontra na fase de modelação, isto é, após a execução da instrução de `gl.glMatrixMode(GL.GL_MODELVIEW)`.

Neste trecho do programa existem várias fases delimitadas pelas variáveis de controlo da boundingbox, wireframe, sólido e textura. Assim, supondo que as variáveis de controlo estão activas, a sequência é: desenho da boundingbox recorrendo à chamada ao método `desenhaCubo`, desenho em sólido, aplicação da textura e por fim desenho da wireframe. De mencionar que a aplicação da textura se sobrepõe ao desenho com cor sólida, assim como a wireframe é aplicada sobre a textura e, consequentemente, sobre o sólido.

Para a aplicação da textura é necessário executar o método auxiliar `aplicaTextura` e no depois de desenhados os vértices, realizar a instrução `gl.glDisable(GL.GL_TEXTURE_2D)` para evitar que a textura seja aplicada à wireframe.

CalculaTetaInicial – Realiza o cálculo para o valor inicial do ângulo teta da projecção axonométrica.

CalculaGamaInicial – Realiza o cálculo para o valor inicial do ângulo gama da projecção axonométrica.

CriarPainelControlo – Cria um painel com abas para a selecção das projecções.

CriarMenuFicheiro – Cria um menu ficheiro com as opções de carregamento de objectos e texturas e ainda a opção de reinício do programa. As operações mencionadas são realizadas apenas quando a selecção dos itens do menu é feita.

CriarMenuVer – Cria um menu com os modos de desenho do objecto: Wireframe – desenha a wireframe; Sólido – desenha o objecto com uma cor sólida; Textura – aplica uma textura sobre o objecto; Boundingbox – desenha um cubo de aresta 1 que delimita o objecto.