1 2 9 0 FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE D COIMBRA

dei25 departamento de engenharia informática 1995 – 2020

Computação Gráfica

André Perrotta (avperrotta@dei.uc.pt)

Hugo Amaro (hamaro@dei.uc.pt)

TP06: Projeção e viewport

Frequencia 2023

O trecho de código apresentado foi extraído de um programa implementado em OpenFrameworks onde não é feita nenhuma alteração às definições iniciais de visualização e projeção executadas por defeito pelo programa. O tamanho de janela utilizado (definido em main.h) é de 800x800 pixels. Considere que a função **lookat(camX, camY, camZ, lookatX, lookatY, lookatZ, upX, upY, upZ)** implementa o algoritmo de câmera *UVN*, que altera a matriz Modelview para refletir o posicionamento da câmera (conforme utilizada nas aulas PL e TP). Considere também que a função **cube_unit()** desenha um cubo unitário (tamanho das arestas = 1) centrado na origem, e que a função **glOrtho(left, right, bottom, top, near, far)** define um volume de projeção ortogonal. Analise o código e responda às perguntas:

```
void ofApp::draw(){
        glViewport(0, 0, 400, 400);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        glOrtho(-1, 1, -1, 1, -10, 10);
        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();
        lookat(-1, 0, 0, 0, 0, 0, 0, 1, 0);
        glScalef(2, 2, 2);
        cube_unit();
}
```

Em qual parte da janela será desenhado o cubo?

Qual o tamanho final do cubo na tela (em pixels)? (justifique)

Qual face do cubo estamos a visualizar?

Como deveria ser a lookat para vizualizar por trás e centrado no viewport?

Exame 2023

Na conceptualizacção e implementação de uma cena 3D utilizando o pipeline poligonal do OpenGl realizamos operações em 4 sistemas de coordenadas (ou espaços). Sobre estes sistemas, responda de forma clara, objetiva e sucinta as seguintes perguntas:

Quais saõ esses 4 sistemas de coordenadas?

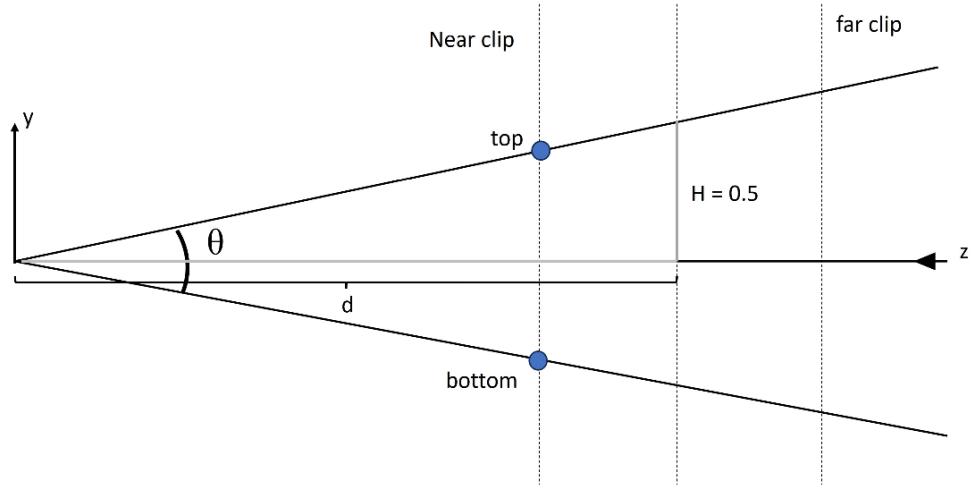
Qual a função de cada um desses sistemas de coordenadas?

Exame 2023

Considere uma aplicação desenvolvida em OpenGl e configurada com uma janela quadrada (largura=altura). Mais, considere que no inicio do programa é configurado o recorte (glViewPort) para toda a tela e que seu volume de projeção é configurado através da função $glFrustum(left, right, bottom, top, near, far)$ com os seguintes valores:

$glFrustum(\frac{-1}{200}, \frac{1}{200}, \frac{-1}{200}, \frac{1}{200}, \frac{\sqrt{3}}{200}, 100\frac{\sqrt{3}}{2})$

A imagem abaixo mostra uma vista lateral do volume de projeção definido.



Qual o valor de theta?

Qual o valor de “d”?

Considere agora que, para além da projeção, é também definida uma vista da cena utilizando o algoritmo UVN implementado na função $lookat(\vec{pos}, \vec{target}, \vec{up})$ com os seguintes valores:

$lookat(0, 0, \frac{\sqrt{3}}{2}, 0, 0, 0, 0, 1, 0)$

E, alteram-se os valores de recorte, utilizando a função $glViewport(x0, y0, width, height)$ com os seguintes valores:

$glViewport(\frac{w}{2}, \frac{h}{2}, \frac{w}{2}, \frac{h}{2})$, onde w e h referem à largura e altura da janela da aplicação.

Após estas definições é então desenhada uma linha entre os vértices $A(\frac{-1}{2}, 0, -\frac{\sqrt{3}}{2})$ e $B(\frac{1}{2}, 0, -\frac{\sqrt{3}}{2})$.

Qual a distância entre os vértices A e B em coordenadas mundo 3D? (justifique)

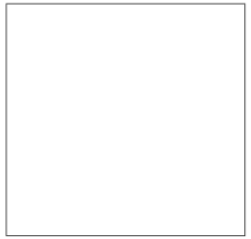
Qual a distância entre os vértices A e B em pixels na janela da aplicação? (justifique)

Q2 (4 valores)

Para cada trecho de código apresentado (Openframeworks/OpenGl), faça um esboço do desenho realizado pela função $axis()$, identificando o "nome" e sentido de cada eixo (x,y,z) quando visíveis. A aplicação foi configurada com uma janela de $(w, h) = (1024, 1024)$ pixels.
(observação: a função $axis()$ é a mesma utilizada no exercício 1.)

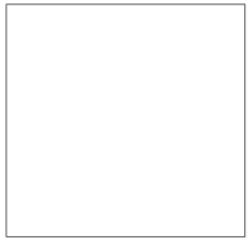
(a)(1 valor):

```
void draw(){
  glViewport(0, 0, w, h);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(-1, 1, -1, 1, -2, 2);
  lookat(0, 0, 1, 0, 0, 0, 0, -1, 0);
  axis();
}
```



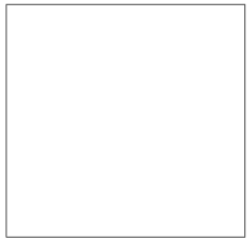
(b)(1 valor):

```
void draw(){
  glViewport(0, 0, w*0.5, h*0.5);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(1, -1, -1, 1, -2, 2);
  lookat(0, 0, -1, 0, 0, 0, 0, -1, 0);
  axis();
}
```



(c)(1 valor):

```
void draw(){
  glViewport(w*0.5, 0, w*0.5, h*0.5);
  perspective(60, 100, 1000);
  lookat(0, h*0.5/tan(PI/6.), 0, 0, 0, 0, 0, 0, -1);
  glScalef(w, w, w);
  axis();
}
```



Melhoria frequencia 2023