O **objetivo desta aula** é aprender como trabalhar com a visualização 3D em OpenGL. Para isto, é necessário compreender como são utilizadas as funções de projeção e manipulação da câmera em OpenGL. Aqui aplicaremos na prática os conceitos apresentados nas aulas teóricas referente a camera virtual e transformação de visualização e projecções.

A aula será guiada de um conjunto de questões que devem ser respondidas e submetidas no format pdf no espaço **submissão de questões prática laboratorial 2**, existente na plataforma classroom.

Questão 1) Analise o código fonte e descreva o que programa faz.

A função gluLookAt(0,80,200, 0,0,0, 0,1,0); define a câmera, isto é, através dos seus argumentos é possível indicar a posição da câmera e para onde ela está direcionada. Seu protótipo é: void gluLookAt(GLdouble eyex, GLdouble eyey, GLdouble eyez, GLdouble centerx, GLdouble centery, GLdouble centerz, GLdouble upx, GLdouble upy, GLdouble upz);. Os parâmetros: eyex, eyey e eyez são usados para definir as coordenadas x, y e z, respectivamente, da posição da câmera (ou observador); centerx, centery e centerz são usados para definir as coordenadas x, y e z, respectivamente, da posição do alvo, isto é para onde o observador está olhando (normalmente, o centro da cena); upx, upy e upz são as coordenadas x, y e z, que estabelecem o vetor up (indica o "lado de cima" de uma cena 3D) [Wright 2000].

Questão 2) Altere o parâmetro *upy* para -1 e escreva o que aconteceu com o *figura* Depois, atribua o valor 1 para *upy* novamente.

Altere o parâmetro *eyez* para 100, compile e visualize o resultado. Depois altere este mesmo parâmetro para 300, compile e visualize o resultado. Agora passe o valor 200 novamente para este parâmetro.

Altere o parâmetro *centerx* para 20, compile e execute o programa. Agora altere este mesmo parâmentro para -20, compile e execute o programa.

Questão 3) Considerando a teoria estudada em aula, responda em que consiste alterar os parâmetros *eyex*, *eyey*, *eyez*, *centerx*, *centery* e *centerz*.

Agora inclua a função abaixo no código fonte

```
// Callback para gerenciar eventos do teclado para teclas especiais (F1, PgDn, entre outras)
void SpecialKeys(int key, int x, int y)
    {
            switch (key) {
                    case GLUT_KEY_LEFT :
                                                    obsX -=10;
                                                    break;
                    case GLUT_KEY_RIGHT :
                                                    obsX +=10;
                                                    break;
                    case GLUT_KEY_UP :
                                                    obsY +=10;
                                                    break;
                    case GLUT_KEY_DOWN :
                                                    obsY -=10;
                                                    break;
                    case GLUT_KEY_HOME :
                                                    obsZ +=10;
                                                    break;
                    case GLUT_KEY_END :
                                                    obsZ -=10;
                                                    break;
            }
            glLoadIdentity();
            gluLookAt(obsX,obsY,obsZ, 0,0,0, 0,1,0);
```

```
        glutPostRedisplay();
    }
```

No início do programa, acrescente as variáveis globais que aparecem abaixo.

```
#include <gl/glut.h>
GLdouble obsX=0, obsY=0, obsZ=200; //acrescente esta linha
```

Para que o *zoom* continue funcionando corretamente, na função *EspecificaParametrosVisualizacao* troque a linha de código *gluLookAt(0,80,200, 0,0,0, 0,1,0);* para *gluLookAt(obsX,obsY,obsZ, 0,0,0, 0,1,0);*.

Na função *main* inclua a seguinte chamada de função *glutSpecialFunc(SpecialKeys);* antes da chamada para a função *Inicializa();*. Compile e execute o programa.

> Questão 4) Qual o objetivo das últimas alterações realizadas no código fonte?

A função gluPerspective(angle,fAspect,0.1,500); estabelece os parâmetros da Projeção Perspectiva, atualizando a matriz de projeção perpectiva. Seu protótipo é: void gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar);. Descrição dos parâmetros: fovy é o ângulo, em graus, na direção y (usada para determinar a "altura" do volume de visualização); aspect é a razão de aspecto que determina a área de visualização na direção x, e seu valor é a razão em x (largura) e y (altura); zNear, que sempre tem que ter um valor positivo maior do que zero, é a distância do observador até o plano de corte mais próximo (em z); zFar, que também sempre tem que ter um valor positivo maior do que zero, é a distância do observador até o plano de corte mais afastado (em z). Esta função sempre deve ser definida ANTES da função gluLookAt, e no modo GL_PROJECTION [Wright 2000].

Altere o parâmetro *zNear* da função gluPerspective para 0.5 (as vezes pode haver um problema na iluminação se o valor for muito baixo, tal como 0.1). Agora coloque 2 no lugar de *fAspect*, compile e execute para ver o que acontece. Troque o valor 2 por 0.5, compile e execute novamente. Coloque

novamente a variável *fAspect*no segundo parâmetro da *gluPerspective*. Com estes testes, foi possível verificar as conseqüências de alterar a razão de aspecto.

Agora altere o parâmetro *zNear* da função gluPerspective para 210, compile e execute para ver o que acontece. Coloque novamente o valor 0.5 para *zNear* e troque o valor de *zFar* para 200, compile e execute. Coloque novamente o valor 500 para *zFar*.

> Questão 5) Explique como foi implementado o *zoom* neste programa e o que acontece quando os valores dos parâmetros *zNear* e *zFar* são alterados.