

OpenGL 3D

Prof. Dr. Davi Marcondes Rocha

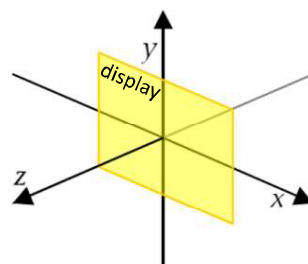
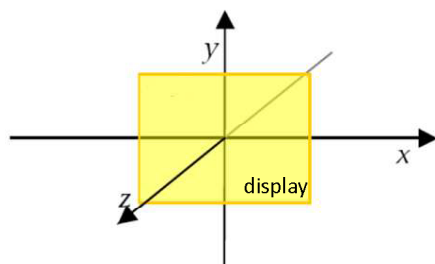
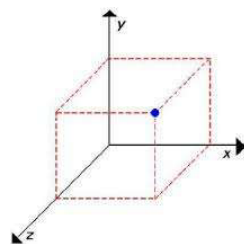
Disciplina: Computação Gráfica

Câmpus Santa Helena



Ministério da
Educação

Coordenadas 3D



Coordenadas 3D

•Exemplo:

```
glVertex3f(3, 3, 4);
```

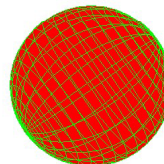
Formas Predefinidas – OpenGL 3D

Esfera:

```
glutSolidSphere(Gldouble radius, Gldouble slices, Gldouble stack)
```

```
glutWireSphere(Gldouble radius, Gldouble slices, Gldouble stack)
```

- Gldouble radius :Raio da Esfera;
- Gldouble slices : Subdivisões ao redor do eixo Z (linhas de longitude);
- Gldouble stack: Subdivisões ao longo do eixo Z (linhas de latitude).



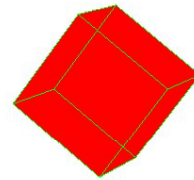
Formas Predefinidas – OpenGL 3D

Cubo:

`glutSolidCube(GLdouble size)`

`glutWireCube(GLdouble size)`

- GLdouble size: Tamanho do cubo



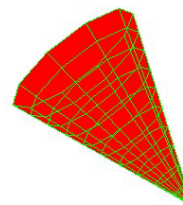
Formas Predefinidas – OpenGL 3D

Cone:

`glutWireCone(GLdouble radius, GLdouble height, GLint slices, GLint stacks);`

`glutSolidCone(GLdouble radius, GLdouble height, GLint slices, GLint stacks)`

- GLdouble radius : Raio da base do cone
- GLdouble height: Altura do Cone
- GLint slices: Subdivisões ao redor do eixo Z (linhas de longitude)
- GLint stacks: Subdivisões ao longo do eixo Z (linhas de latitude).

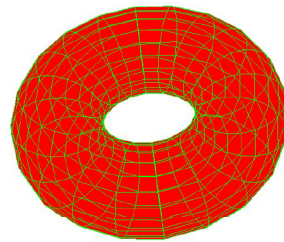


Formas Predefinidas – OpenGL 3D

Toróide:

```
glutWireTorus(GLdouble innerRadius, GLdouble outerRadius, GLint nSides, GLint rings);  
glutSolidTorus(GLdouble innerRadius, GLdouble outerRadius, GLint nSides, GLint rings);
```

- GLdouble innerRadius : Raio interno
- GLdouble outerRadius : Raio externo
- GLint nSides : número de seções que serão usadas para formar o *torus*;
- GLint rings : número de subdivisões para cada seção.

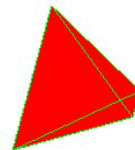


Formas Predefinidas – OpenGL 3D

Tetraedro:

```
glutWireTetrahedron(void);  
glutSolidTetrahedron(void);
```

- renderiza um tetraedro, centrado na origem das coordenadas de modelagem



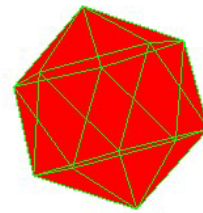
Formas Predefinidas – OpenGL 3D

Icosaedro:

```
glutWireIcosahedron(void);
```

```
glutSolidIcosahedron(void);
```

- renderiza um icosaedro, centrado na origem das coordenadas de modelagem



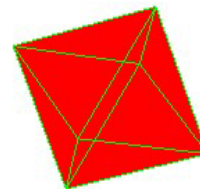
Formas Predefinidas – OpenGL 3D

Octaedro:

```
glutWireOctahedron(void);
```

```
glutSolidOctahedron(void);
```

- renderiza um octaedro, centrado na origem das coordenadas de modelagem



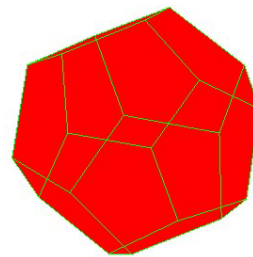
Formas Predefinidas – OpenGL 3D

Dodecaedro:

```
glutWireDodecahedron(void);
```

```
glutSolidDodecahedron(void);
```

- renderiza um dodecaedro, centrado na origem das coordenadas de modelagem



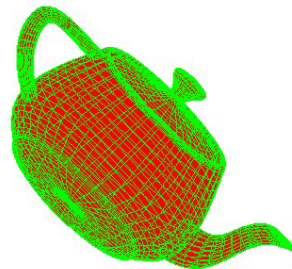
Formas Predefinidas – OpenGL 3D

Chaleira:

```
WireTeapot(GLDouble size);
```

```
SolidTeapot(GLDouble size);
```

- GLDouble size: Tamanho da chaleira a ser renderizada

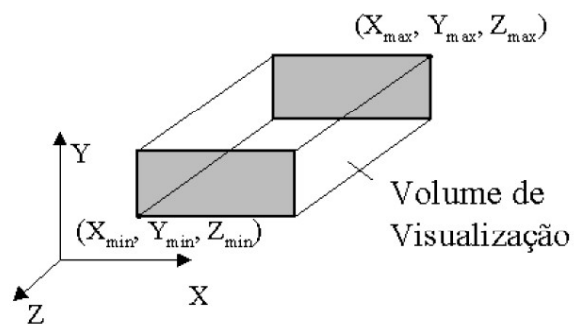


Projeção

Existem dois tipos básicos de projeções possíveis em OpenGL:
Ortográfica e Perspectiva

Projeção Ortográfica

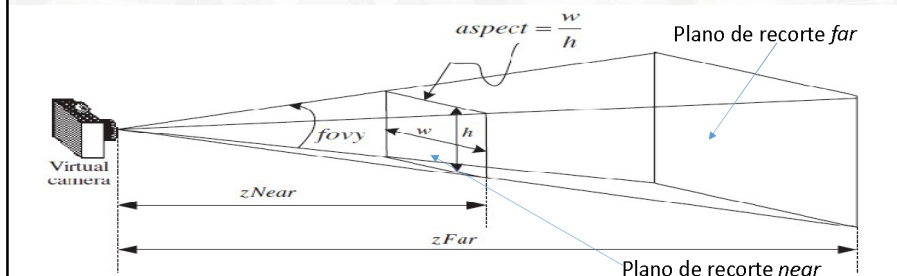
- `glOrtho`



Projeção Perspectiva

gluPerspective(GLdouble fovy, GLdouble aspect, GLdouble zNear, GLdouble zFar);

- GLdouble fovy: Especifica o ângulo do campo de visão, em graus, na direção y.
- GLdouble aspect: Especifica a razão entre a largura e altura da janela (razão da janela x e y)
- GLdouble zNear: Especifica a distância do visualizador ao plano de recorte próximo a câmera (imite perto – em z).
- GLdouble zFar: Especifica a distância do visualizador ao plano de recorte distante da câmera (imite longe – em z).



Projeção Perspectiva

Em conjunto com a perspectiva:

gluLookAt(GLdouble eyeX, GLdouble eyeY, GLdouble eyeZ, GLdouble centerX, GLdouble centerY, GLdouble centerZ, GLdouble upX, GLdouble upY, GLdouble upZ);

Ou seja,

gluLookAt(obsX, obsY, obsZ, alvoX, alvoY, alvoZ, 0,1,0);

Obs-define a posição do observador;

Alvo –posição do ponto para onde o observador esta olhando;

0,1,0 –define o lado de cima do cenário 3D.

