



Universidade Federal do Piauí
Centro de Ciências da Natureza
Departamento de Computação

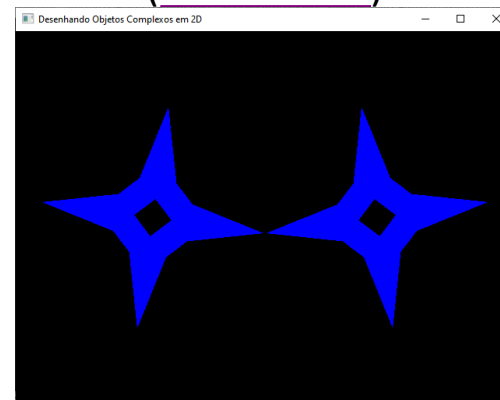


Modelagem Geométrica, Sombreamento e Animação usando OpenGL

Prof. Dr. Laurindo de Sousa Britto Neto

1

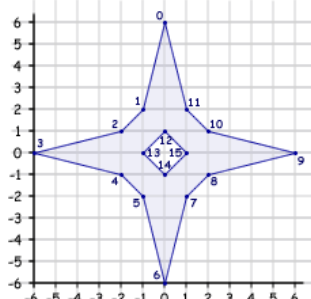
Modelando Objetos 2D (1/4) (desenho2D)



2

2

Modelando Objetos 2D (2/4)

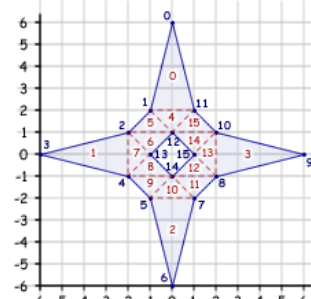


```
// Vertices do objeto
#define N_VERTICES 16
const GLfloat
vertices[N_VERTICES][3] = {
    { 0.0,  6.0,  0.0}, //Vertex 0
    {-1.0,  2.0,  0.0}, //Vertex 1
    {-2.0,  1.0,  0.0}, //Vertex 2
    {-6.0,  0.0,  0.0}, //Vertex 3
    {-2.0, -1.0,  0.0}, //Vertex 4
    {-1.0, -2.0,  0.0}, //Vertex 5
    { 0.0, -6.0,  0.0}, //Vertex 6
    { 1.0, -2.0,  0.0}, //Vertex 7
    { 2.0, -1.0,  0.0}, //Vertex 8
    { 6.0,  0.0,  0.0}, //Vertex 9
    { 2.0,  1.0,  0.0}, //Vertex 10
    { 1.0,  2.0,  0.0}, //Vertex 11
    { 0.0,  1.0,  0.0}, //Vertex 12
    {-1.0,  0.0,  0.0}, //Vertex 13
    { 0.0, -1.0,  0.0}, //Vertex 14
    { 1.0,  0.0,  0.0} //Vertex 15
};
```

3

3

Modelando Objetos 2D (3/4)



```
// Faces do objeto (Triangulos)
#define N_FACES 16
const GLuint faces[N_FACES][3] = {
    { 0, 1, 11}, //Face 0
    { 2, 3, 4}, //Face 1
    { 5, 6, 7}, //Face 2
    {10, 8, 9}, //Face 3
    { 1, 12, 11}, //Face 4
    { 1, 2, 12}, //Face 5
    { 2, 13, 12}, //Face 6
    { 2, 4, 13}, //Face 7
    {13, 4, 14}, //Face 8
    {14, 4, 5}, //Face 9
    {14, 5, 7}, //Face 10
    {14, 7, 8}, //Face 11
    {15, 14, 8}, //Face 12
    {15, 8, 10}, //Face 13
    {12, 15, 10}, //Face 14
    {11, 12, 10} //Face 15
};
```

4

4

Modelando Objetos 2D (4/4) (desenho2D)

```

faces[i][0];      vertice[faces[i][0]];
faces[i][1];      vertice[faces[i][1]];
faces[i][2];      vertice[faces[i][2]];

void draw_object(void){
    int i;

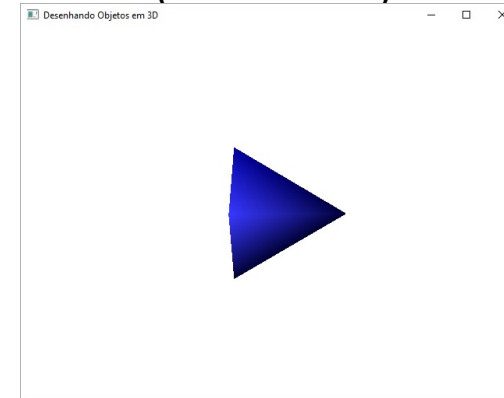
    // Desenha todos os triângulos do objeto
    glBegin(GL_TRIANGLES);
    for (i = 0; i < N_FACES; i++){
        glVertex3fv(vertice[faces[i][0]]);
        glVertex3fv(vertice[faces[i][1]]);
        glVertex3fv(vertice[faces[i][2]]);
    }
    glEnd();
}

```

5

5

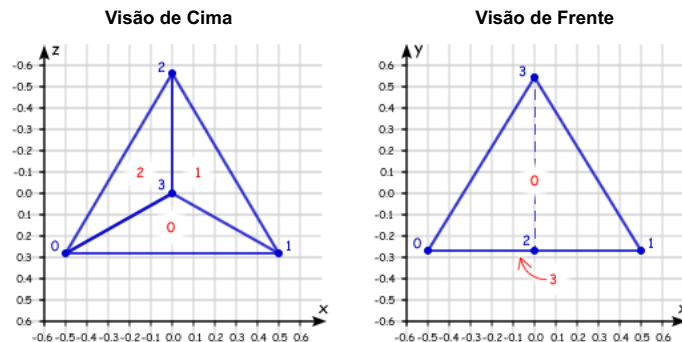
Modelando Objetos 3D (1/8) (desenho3D)



6

6

Modelando Objetos 3D (2/8)



7

7

Modelando Objetos 3D (3/8)

```

// Vertices do objeto
#define N_VERTICES 4
const GLfloat vertices[N_VERTICES][3] = {
    { -0.5, -0.272, 0.289 }, //Vertex 0
    {  0.5, -0.272, 0.289 }, //Vertex 1
    {  0.0, -0.272, -0.577 }, //Vertex 2
    {  0.0,  0.544,  0.000 }  //Vertex 3
};

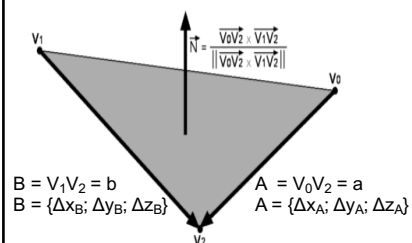
// Faces do objeto (Triangulos)
#define N_FACES 4
const GLuint faces[N_FACES][3] = {
    //sentido anti-horário
    {3, 0, 1}, //Face 0
    {3, 1, 2}, //Face 1
    {3, 2, 0}, //Face 2
    //sentido horário
    {0, 2, 1}  //Face 3
};

```

8

8

Modelando Objetos 3D (4/8)



Calculando a Normal da Face:

```
a.x = x[2] - x[0]; // ΔxA
a.y = y[2] - y[0]; // ΔyA
a.z = z[2] - z[0]; // ΔzA
```

```
b.x = x[2] - x[1]; // ΔxB
b.y = y[2] - y[1]; // ΔyB
b.z = z[2] - z[1]; // ΔzB
```

```
n.x = a.y * b.z - a.z * b.y;
n.y = a.z * b.x - a.x * b.z;
n.z = a.x * b.y - a.y * b.x;
```

$$N_{AB} = \frac{V_{AB}}{\|V_{AB}\|}$$

$$V_{AB} = A \times B = \overrightarrow{V_0V_1} \times \overrightarrow{V_0V_2} = (y_A z_B - y_B z_A, x_B z_A - x_A z_B, x_A y_B - x_B y_A)$$

9

9

Modelando Objetos 3D (5/8)

$$N_{AB} = \frac{V_{AB}}{\|V_{AB}\|}$$

$$\|V_{AB}\| = \sqrt{x_{V_{AB}}^2 + y_{V_{AB}}^2 + z_{V_{AB}}^2}$$

Normalizando vetor normal

```
void normalizar(vertice * n){
    GLfloat length;
    length = (GLfloat) sqrt((n->x * n->x) + (n->y * n->y) + (n->z * n->z));

    if (length == 0.0f) length = 1.0f;

    n->x = n->x / length;
    n->y = n->y / length;
    n->z = n->z / length;
}
```

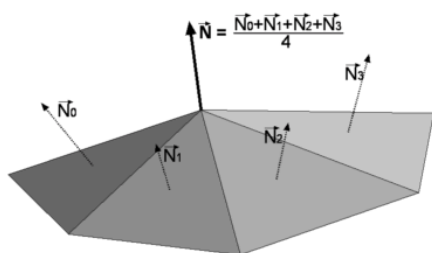
10

10

Modelando Objetos 3D (6/8)

• Calculando Normais do Vértices

- Flat-shading - `glShadeModel(GL_FLAT);`
- Gouraud-shading - `glShadeModel(GL_SMOOTH);`



11

11

Modelando Objetos 3D (7/8)

```
void draw_object_flat(void){
    GLuint i;

    // Desenha todos os triângulos do objeto
    glBegin(GL_TRIANGLES);
    for (i = 0; i < N_FACES; i++){
        glNormal3fv(face_normals[i]);
        glVertex3fv(vertices[faces[i][0]]);
        glVertex3fv(vertices[faces[i][1]]);
        glVertex3fv(vertices[faces[i][2]]);
    }
    glEnd();
}
```

12

12

Modelando Objetos 3D (8/8) (desenho3D)

```
void draw_object_smooth(void){
    GLuint i;

    // Desenha todos os triângulos do objeto
    glBegin(GL_TRIANGLES);
    for (i = 0; i < N_FACES; i++){
        glNormal3fv(vertex_normals[faces[i][0]]);
        glVertex3fv(vertices[faces[i][0]]);

        glNormal3fv(vertex_normals[faces[i][1]]);
        glVertex3fv(vertices[faces[i][1]]);

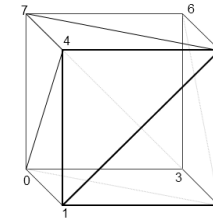
        glNormal3fv(vertex_normals[faces[i][2]]);
        glVertex3fv(vertices[faces[i][2]]);
    }
    glEnd();
}
```

13

13

Tarefa

1. Baseado no código do desenho3D, modele um cubo com malha triangular em OpenGL.



14

14