

# *Modelação 3D em JOGL*

Gil Cunha

[gil.cunha@icloud.com](mailto:gil.cunha@icloud.com)



# Introdução

- *Programação OpenGL em Java: JOGL*
  - O *OpenGL* é a biblioteca gráfica mais utilizada. Tem suporte nas plataformas *desktop* e nas estações de trabalho mais populares [1].
  - O *JOGL* implementa as ligações *Java* para *OpenGL*.
- *Modelos Gráficos*
  - Um objeto real ou imaginário é representado em computador por um modelo que posteriormente é exibido no ecrã como uma imagem.
  - Um modelo é uma descrição abstrata da geometria (vértices) e atributos (cores) do objeto que pode ser usada para encontrar todos os pontos e cores do objeto na área de desenho [1].



# Sumário

- *Modelação de um Paralelepípedo Retangular*
  - Os Vértices
  - As Faces
  - Geometria e Topologia
  - O Desenho





## Os Vértices

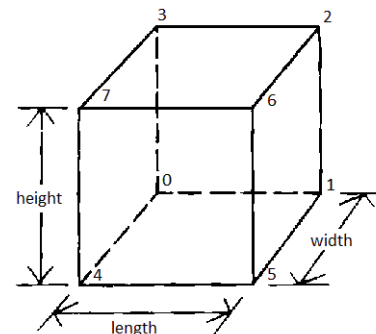
- Conjunto de vértices

```
// Vértices do paralelepípedo
private float[][] vertices;

// Lados do paralelepípedo
private static byte[][] faces = new byte[][][]

// Recebe informação do vértice de posicionamento e as dimensões
public Parallelepiped(float xOri, float yOri, float zOri, float l, float h, float w){
    length = l;
    height = h;
    width = w;
    setVertices(xOri, yOri, zOri);
}

public void setVertices(float xOri, float yOri, float zOri) {
    vertices = new float[][]
    {
        {xOri, yOri, zOri},
        {xOri + length, yOri, zOri},
        {xOri + length, yOri + height, zOri},
        {xOri, yOri + height, zOri},
        {xOri, yOri, zOri + width},
        {xOri + length, yOri, zOri + width},
        {xOri + length, yOri + height, zOri + width},
        {xOri, yOri + height, zOri + width}
    };
}
```





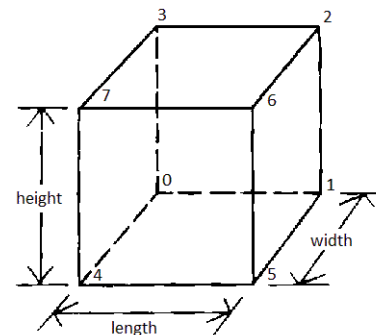
## As Faces

- Conjunto de faces
  - Os Lados Interior e Exterior

```
// Lados do paralelepípedo
private static byte[][] faces = new byte[][]
{
    {0, 3, 2, 1}, // posterior
    {4, 5, 6, 7}, // frontal

    {3, 0, 4, 7}, // esquerdo
    {6, 5, 1, 2}, // direito

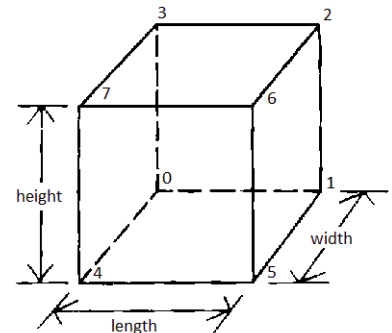
    {5, 4, 0, 1}, // inferior
    {7, 6, 2, 3} // superior
};
```





# Geometria e Topologia

- Geometria
  - Conjunto de Vértices
- Topologia
  - Paralelepípedo Retangular
  - Composto por 6 faces
  - Cada face definida através de 4 vértices ordenados
  - Cada vértice pode ser referenciado indiretamente através do índice.





# O Desenho

- Desenho de acordo com a Topologia

```
// Desenha paralelepípedo
public void drawParallelepiped(GL gl) {
    for(int i = 0; i < faces.length; ++i) {
        gl.glBegin(GL.GL_QUADS);
        for(int j = 0; j < faces[i].length; ++j) {
            gl.glVertex3f(vertices[faces[i][j]][0], vertices[faces[i][j]][1], vertices[faces[i][j]][2]);
        }
        gl.glEnd();
    }
}
```



## Resumo

```
package Geometry;
import javax.media.opengl.GL;

public class Paralelepiped {
    // Dimensões
    private float length, height, width;

    // Vértices do paralelepípedo
    private float[][] vertices;

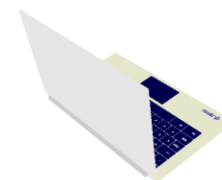
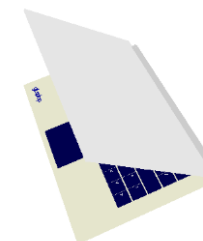
    // Lados do paralelepípedo
    private static byte[][] faces = new byte[][]
    {
        {0, 3, 2, 1}, // posterior
        {4, 5, 6, 7}, // frontal
        {3, 0, 4, 7}, // esquerdo
        {6, 5, 1, 2}, // direito
        {5, 4, 0, 1}, // inferior
        {7, 6, 2, 3} // superior
    };

    // Recebe informação do vértice de posicionamento e as dimensões
    public Paralelepiped(float xOri, float yOri, float zOri, float l, float h, float w){
        length = l;
        height = h;
        width = w;
        setVertices(xOri, yOri, zOri);
    }

    public void setVertices(float xOri, float yOri, float zOri) {}

    public void setVertices(float[][] v) {}
    public float getLength() {}
    public float getHeight() {}
    public float getWidth() {}
    public float[][] getVertices() {}

    // Desenha paralelepípedo
    public void drawParalelepiped(GL gl) {
        for(int i = 0; i < faces.length; ++i) {
            gl.glBegin(GL.GL_QUADS);
            for(int j = 0; j < faces[j].length; ++j) {
                gl.glVertex3f(vertices[faces[i][j]][0], vertices[faces[i][j]][1], vertices[faces[i][j]][2]);
            }
            gl.glEnd();
        }
    }
}
```





# Mais Informações

- [1] Chen, Jim X. e Chen, Chunyang (2008). Foundations of 3D Graphics Programming Using JOGL and Java3D (2ª. ed.). Springer.
- [2] Angel, Edward (2009). Interactive Computer Graphics – A Top-Down Approach Using OpenGL (5ª. ed.). Pearson.