

Universidade Federal da Paraíba

Ciências da Computação

Introdução à Computação Gráfica

Atividade 2 - ICG

Alunos:

Luciano Pereira - 20190018530

Abraão Homualdo - 20200095558

1 Resumo

Como objetivo desta atividade, foi desenvolvido cálculos por embasamento através de matrizes com o intuito e finalidade de gerar um rendering de um cubo formado por arestas.

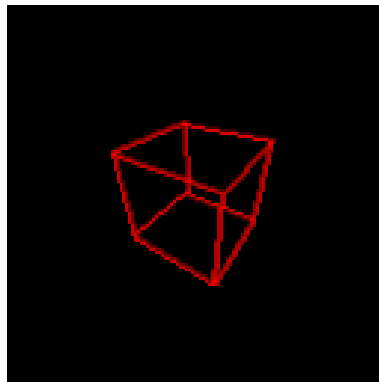


Figura 1: Resultado do Rendering da Atividade

Em princípio, foram alterados todos os valores das matrizes (Model, View, ViewPort, Proj) e com a interpolação da "Homogeneização", como também em virtude da rotação, angulação, forma e projeção do respectivo cubo.

No projeto em si, foi utilizado o framework Three.js, para criação das matrizes 4x4 como modelo descrito, e operações de álgebra linear para a ligação das linhas algoritmas. Abaixo segue algumas das matrizes essenciais para gerar (View, Viewport, Model, Proj).

```
function Matriz_escala(sx, sy, sz) { //T
    let m_escala = new THREE.Matrix4();

    m_escala.set(sx, 0.0, 0.0, 0.0,
                 0.0, sy, 0.0, 0.0,
                 0.0, 0.0, sz, 0.0,
                 0.0, 0.0, 0.0, 1.0);

    return m_escala;
}
```

Figura 2: Matriz da Transformação da escala do CUBO

```

function Matriz_shear(shx, shy, shz) { //Shear
    let m_shear = new THREE.Matrix4(); // Matriz de Shear

    m_shear.set(1.0, shy, shz, 0.0,
                shx, 1.0, shz, 0.0,
                shx, shy, 1.0, 0.0,
                0.0, 0.0, 0.0, 1.0);

    return m_shear;
}

```

Figura 3: Matriz Shear

```

function Matriz_translacao(tx, ty, tz) { //Translação
    let m_translacao = new THREE.Matrix4();

    m_translacao.set(1.0, 0.0, 0.0, tx,
                    0.0, 1.0, 0.0, ty,
                    0.0, 0.0, 1.0, tz,
                    0.0, 0.0, 0.0, 1.0);

    return m_translacao;
}

```

Figura 4: Matriz de Translação

```

function Matriz_rotacao(eixo, angulo) {

    let radiano = angulo * Math.PI / 180.0;

    let cos = Math.cos(radiano);
    let sen = Math.sin(radiano);

    let m_rotacao = new THREE.Matrix4();

    if (eixo === 'X' || eixo === 'x') {
        m_rotacao.set(1.0, 0.0, 0.0, 0.0,
                      0.0, cos, -sen, 0.0,
                      0.0, sen, cos, 0.0,
                      0.0, 0.0, 0.0, 1.0);
    } else if (eixo === 'Y' || eixo === 'y') {
        m_rotacao.set(cos, 0.0, sen, 0.0,
                      0.0, 1.0, 0.0, 0.0,
                      -sen, 0.0, cos, 0.0,
                      0.0, 0.0, 0.0, 1.0);
    } else if (eixo === 'Z' || eixo === 'z') {
        m_rotacao.set(cos, -sen, 0.0, 0.0,
                      sen, cos, 0.0, 0.0,
                      0.0, 0.0, 1.0, 0.0,
                      0.0, 0.0, 0.0, 1.0);
    }

    return m_rotacao;
}

```

Figura 5: Matriz de Rotação(com as devidas condições de estruturação)

A dificuldade que foi encontrada no projeto foi que, a elaboração das matrizes exigia uma certa noção de geometria condicional, onde a alteração dos valores em cadeia fazia com que o cubo ele não tomasse a forma desejada, por fim a rendering sofreu grandes mutações até chegar no seu devido resultado, pois as matrizes devem estar em conjunto numérico e complementar.

Bibliografia

Framework Three.js, <https://github.com/mrdoob/three.js/>

Repositório da Atividade, <https://github.com/luci18530/Atividade2-ICG>