

DADOS DO RELATÓRIO

Matéria: Computação Gráfica Cod: CMP1170

Relatório Atividade 2

Aluno: Vitor de Almeida Silva

Matrícula: 20161003305497

Requisitos: processing 3.0 ou superior – openGL

ENUNCIADO

Utilizando o Processing, elabore um programa que desenhe um cubo.

Sugestão: Acesse o link com um tutorial sobre processing
(<https://processing.org/tutorials/p3d/>).

DESENVOLVIMENTO

1 Funções:

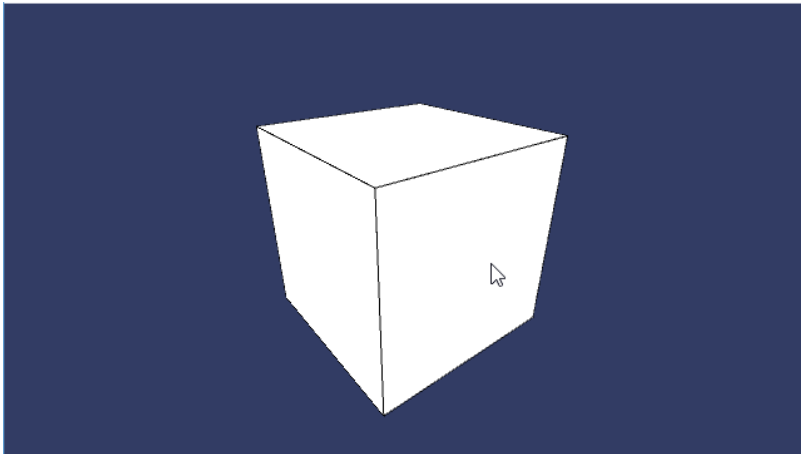
Para realização do exercício foram utilizadas as seguintes funções (as mesmas também foram comentadas no código):

- **size (x, y, dimensão):** indica o tamanho da janela em x, y e a dimensão do desenho;
- **stroke(valor) ou Stroke(R, G, B):** define que o desenho deve ter traços, valor é a cor do traço entre 0 e 250;
- **background(valor) ou background(R, G, B):** define a cor do fundo da figura;
- **translate(x, y, z):** função que realiza a translação do desenho nos eixos do plano;
- **mouseX:** variável que captura o valor da posição do mouse em termos do eixo x;
- **mouseY:** variável que captura o valor da posição do mouse em termos do eixo y;
- **abs(valor):** função que captura o valor absoluto de um número;
- **rotateX(x):** função capaz de rotacionar a figura no eixo x;
- **rotateY(y):** função capaz de rotacionar a figura no eixo y;
- **box(dimensao):** função que cria um cubo de acordo com o tamanho da dimensão definida.
- **mouseWheel(Mouse event):** função que captura a mudança de valor do *scroll* do mouse;

2 Execução:

O código foi desenvolvido de modo que seja possível girar o cubo no eixo x e y. Da mesma forma, através da função *translate* combinada com a função *mouseWheel*, é capaz de realizar zoom no desenho do cubo. A Figura 1 mostra uma imagem da aparência do cubo.

Figura 1: Cubo



Fonte: Autoral

A seguir é exposto o código do cubo em resposta a questão imposta no exercício.

```

float xmag, zmag=0, ymag = 0;

float newXmag,newZmag , newYmag = 0;

void setup() {

    size(640, 360, P3D);

    stroke(0);    //função que par colocar os traços no desenho Stroke(R,G,B);
}

void draw() {

    background(50,60,100); //essa função estando no loop impede o aparecimento de "fantasmas"

    translate(width/2, height/2, newZmag );

    newXmag = mouseX/float(width) * TWO_PI;

    newYmag = mouseY/float(height) * TWO_PI;

    //eixo z é controlado pelo mouse

    float diff = xmag-newXmag; //cálculo realizado fazendo a diferença da posição inicial com a atual

    if (abs(diff) > 0.01){ //para saber o valor do movimento

        xmag -= diff/4.0;

    }

    diff = ymag-newYmag;

    if (abs(diff) > 0.01){

        ymag -= diff/4.0;

    }

    rotateX(-ymag);    //roda a figura de acordo com a posição do ponteiro do mouse

    rotateY(-xmag);

    box(90);

}

//função que captura a mudança de valor do scrool do mouse usei para zoom (eixo z)

void mouseWheel(MouseEvent event) {

    newZmag+= -event.getCount()*6;    //o que ocorre é uma soma de valores vindos do scrool

}

//essa função funciona orientada ao evento do scrool, ela interrompe o processo para atualizar o valor

//retornado por ele

```