PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA DE CIÊNCIAS EXATAS E DA COMPUTAÇÃO

GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO



**TRANSFORMAÇÕES: AUMENTO DE ESCALA E ROTAÇÃO - FOGUETE**

LUCAS MACEDO DA SILVA

GOIÂNIA

2020

**Aplicação das transformações de rotação e aumento de escala no objeto criado com o processing**

Foi desenvolvido um objeto tridimensional, a partir de três objetos primitivos, usando o software Processing.

O objeto escolhido foi um foguete. Depois foram aplicadas as transformações de rotação e aumento de escala no objeto usando o seguinte código. As operações foram utilizadas de forma interativa, isto é, ao manipular o mouse o objeto muda conforma a operação. Ao passar o mouse perto do objeto, realiza-se a operação de rotação e ao rolar o scroll do mouse, realiza-se a operação de aumento de escala

**Autores:**

* Higor Alves Ferreira;
* Lucas Macedo da Silva;
* Vitor de Almeida Silva.

|  |
| --- |
| /\*  Foguete 3D com Processing  Alunos: Higor Alves Ferreira  Lucas Macedo da Silva  Vitor de Almeida Silva;  Função: desenha\_cilindro adapatada de Reas e Fry (2007)  \*/  PImage imagem;  float newZmag=0;  void setup() {  size(500,500,P3D);  imagem = loadImage("image.jpg");    }  void draw(){    background(imagem); // fundo preto  lights(); // Luzes  translate(width/2, height/2, newZmag); // Mudando a origem do sistema de coordenadas para o meio da tela  // A variável newZmag sofre efeito do scrool do mouse alterando o valor de posicionamento  //do objeto no eixo z, desse modo provocando o efeito de zoom  // ---------------------- Rotação ---------------------------------------  // Rotaciona a imagem de forma interativa  /\*rotate -> Rotaciona o sistema de coordenadas para que as formas sejam desenhadas na tela em angulos\*/  rotateY(map(mouseX, 0, width, 0, PI)); // Rotaciona no eixo Y  rotateZ(map(mouseY, 0, height, 0, -PI)); // Rotaciona no eixo Z  // ---------------------- Rotação ---------------------------------------        noStroke();    fill(0, 0, 180); // Cor da forma  // Propursor do foguete  translate(0, -40, 0);  desenha\_cilindro(70, 100, 200, 4);  fill(255, 255, 255); // Cor da forma  // Cilindro do meio do foguete  translate(0, -40, 0);  desenha\_cilindro(70, 70, 200, 64);    //Cone de cima do foguete  translate(0, -90, 0);  desenha\_cilindro(0, 70, 90, 64);  }  void desenha\_cilindro(float raio\_cima, float raio\_baixo, float altura, int qtd\_lados){  /\*  Desenha um cilindro ou outra forma dependendo da quantidade de lados.  Para isso, usa a função beginShape, vertex e endShape.  \*/  float angulo = 0;  float angulo\_incremento = TWO\_PI / qtd\_lados;    beginShape(QUAD\_STRIP); // Método usado para iniciar uma forma personalizada    // Desenha os vertices para cada lado das figuras da base e do topo  for (int i = 0; i < qtd\_lados + 1; i ++){  //vertex(x, y, z)  vertex(raio\_cima \* cos(angulo), 0, raio\_cima \* sin(angulo));  vertex(raio\_baixo \* cos(angulo), altura, raio\_baixo \* sin(angulo));  angulo += angulo\_incremento;  }    endShape();    if (raio\_cima != 0){  angulo = 0;    beginShape(TRIANGLE\_FAN);  vertex(0, 0, 0);  for (int i = 0; i < qtd\_lados + 1; i ++){  vertex(raio\_cima \* cos(angulo), 0, raio\_cima \* sin(angulo));  angulo += angulo\_incremento;  }  endShape();    }    if (raio\_baixo != 0){  angulo = 0;    beginShape(TRIANGLE\_FAN);  vertex(0, altura, 0);  for (int i = 0; i < qtd\_lados + 1; i ++){  vertex(raio\_baixo \* cos(angulo), altura, raio\_baixo \* sin(angulo));  angulo += angulo\_incremento;  }  endShape();    }    }  // ---------------------------------- Aumento de escala ------------------------------  // Aumento de escala de forma interativa  // Função que captura a mudança de valor do sccroll do mouse usada para zoom (aumento de escala) no (eixo z)  void mouseWheel(MouseEvent event) {  newZmag+= -event.getCount()\*6; // O que ocorre é uma soma de valores vindos do sccroll  }  // Essa função funciona orientada ao evento do sccroll, ela interrompe o processo para atualizar o valor  // retornado por ele  /\*  Referencias:  REAS, Casey; FRY, Ben. Processing: a programming handbook for visual designers and artists. Massachusetts: The Mit Press, 2007.  \*/ |