

Projeto de Computação Gráfica

Daniel Mahl Gregorini



Roteiro

- Detalhes do projeto
 - Modelagem hierárquica na bicicleta
 - Textura
 - Iluminação
 - Animação dos pedais
 - Input e eventos do teclado e mouse





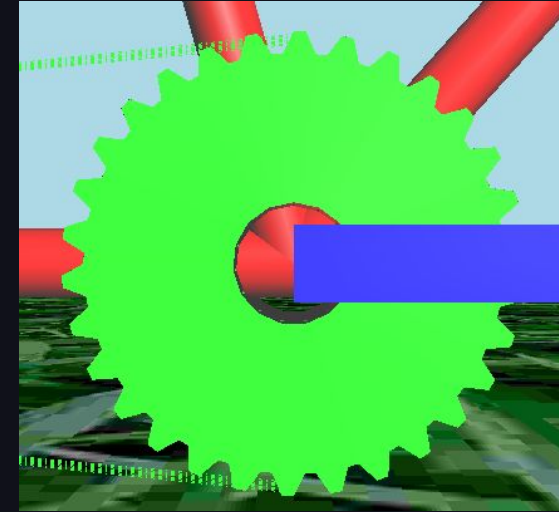
Modelagem hierárquica na bicicleta

```
void desenharQuadro(void);  
void engrenagem(GLfloat inner_radius, GLfloat outer_radius, GLfloat width,  
GLfloat teeth, GLfloat tooth_depth);  
void desenharCorrente(void);  
void desenharPedais(void);  
void desenharPneu(void);  
void desenharAssento(void);  
  
void desenharQuadro(){  
    //engrenagem da frente  
    engrenagem(0.08f, 0.3f, 0.03f, 30, 0.03f);  
    //engrenagem da roda traseira  
    engrenagem(0.03f, 0.15f, 0.03f, 20, 0.03f);  
    desenharPneu();  
  
    desenharPneu();  
  
    desenharAssento();  
}
```



Construção da engrenagem

- **Cálculo de Ângulo:**
 - $\text{angle} = i * 2.0 * \pi / \text{teeth}$
- **Divisão de Arco:**
 - $\text{da} = 2.0 * \pi / \text{teeth} / 4.0;$
- **Coordenadas Polares para Cartesiana:**
 - $x = r * \cos(\text{angle});$
 - $y = r * \sin(\text{angle});$
- **Cálculo do Vetor Normal:**
 - $u = r2 * \cos(\text{angle} + \text{da}) - r1 * \cos(\text{angle});$
 - $v = r2 * \sin(\text{angle} + \text{da}) - r1 * \sin(\text{angle});$
- **Normalização de Vetores:**
 - $\text{len} = \sqrt{u * u + v * v}; u /= \text{len}; v /= \text{len};$





Textura

```
int carregarTexturas () {
    texture = SOIL_load_OGL_texture ( "\\grama.jpg", SOIL_LOAD_RGBA,
SOIL_CREATE_NEW_ID,SOIL_FLAG_INVERT_Y);
    if (texture == 0)
        fprintf(stderr, "Erro ao carregar textura: %s\n",
SOIL_last_result());
        return false;

    glBindTexture(GL_TEXTURE_2D, texture);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glEnable(GL_TEXTURE_2D);
    glShadeModel(GL_SMOOTH); // Habilitar sombreado suave
    glClearColor(0.68f, 0.85f, 0.90f, 1.0f); // Cor de fundo azul claro
    glClearDepth(1.0f); // Configurar buffer de profundidade
    glEnable(GL_DEPTH_TEST); // Habilitar teste de profundidade
    glDepthFunc(GL_LEQUAL); // Configurar o tipo de teste de profundidade
    glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
    return true;
}
```



Iluminação



```
void init()
{
    GLfloat mat specular[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat mat shininess[] = {100.0};
    GLfloat light directional[] = {1.0, 1.0, 1.0, 1.0};
    GLfloat light positional[] = {1.0, 1.0, 1.0, 0.0};
    GLfloat light_diffuse[] = {1.0, 1.0, 1.0};
    reset();
    glShadeModel(GL_SMOOTH);
    glLightfv(GL_LIGHT0, GL_POSITION, light_directional);
    glLightfv(GL_LIGHT0, GL_AMBIENT, light_diffuse);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, light_diffuse);
    glMaterialfv(GL_FRONT, GL_SHININESS, mat_shininess);
    glMaterialfv(GL_FRONT, GL_SPECULAR, mat_specular);
    glColorMaterial(GL_FRONT, GL_DIFFUSE);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0); glEnable(GL_COLOR_MATERIAL);
    glEnable(GL_DEPTH_TEST);
}
```



Animação dos pedais

```
pedalAngle += speed;  
    //não deixa a velocidade da bike menor que 0  
    if (speed < 0.0f)  
        speed = 0.0f;  
    if (pedalAngle < 0.0f)  
        pedalAngle = 0.0f;  
    if (pedalAngle >= 360.0f)  
        pedalAngle -= 360.0f;
```

```
pedalAngle = degrees(angleSum(radians(pedalAngle), speed / RADIUS_WHEEL));
```




Input e eventos do teclado e mouse

```
// Define a função que trata teclas especiais  
glutSpecialFunc(special);  
// Define a função que trata teclas comuns do teclado  
glutKeyboardFunc(teclado);  
// Define a função que trata eventos de clique do mouse  
glutMouseFunc(mouse);
```



Fontes

Gregorini, D. Projeto Final de Computação Gráfica. Disponível em:
<https://github.com/DanielGregorini/projeto-final-computacao-grafica>.
Acesso em: 24/06/2024.