

# Relatório do Projeto de Computação Gráfica - Cenário Virtual

## 1. Identificação

### Integrantes do Grupo:

- [Rodrigo Yamaya Gonçalves]
- [Lucas dos Santos Ottvagen]
- [Luiz Felipe Almeida Veloso]

## 2. Instruções de Instalação e Execução

### Bibliotecas Necessárias

Para executar a aplicação, é necessário ter o Python instalado (versão 3.10) e instalar as seguintes dependências:

- **PyGame** (pygame): Responsável pela criação da janela, gerenciamento de contexto OpenGL e captura de entrada (teclado/mouse).
- **PyOpenGL** (PyOpenGL): Bindings do OpenGL para Python.
- **PyGLM** (PyGLM): Biblioteca matemática para cálculos vetoriais e matriciais (essencial para a câmera e transformações).
- **NumPy** (numpy): Manipulação eficiente de arrays de dados.
- **Pillow** (Pillow): Utilizada para carregar texturas de imagem.

### Comando para instalação rápida:

```
pip install pygame PyOpenGL PyGLM numpy Pillow
```

### Script Inicial

O ponto de entrada da aplicação é o arquivo `main.py`. Para iniciar, abra a pasta do projeto e execute no terminal:

```
cd projeto
python -3.10 main.py
```

### 3. Descrição dos Controles (Interação do Usuário)

A aplicação simula uma câmera em primeira pessoa (FPS), onde o usuário controla a visão e o movimento de um observador no cenário.

#### Movimentação

- **Teclas W / S:** Movem a câmera para **frente** e para **trás** no plano horizontal (eixo XZ). A movimentação é travada no chão para evitar que o observador "voe" ou entre na terra.
- **Teclas A / D:** Movem a câmera lateralmente para a **esquerda** e **direita** (Strafe).
- **Tecla SHIFT (Esquerdo):** Ativa a **corrida**, dobrando a velocidade de movimento enquanto pressionada.
- **Tecla ESPAÇO:** Realiza um **pulo**. A física inclui gravidade, trazendo o observador de volta ao chão (altura dos olhos em 1.8 unidades).

#### Orientação (Visão)

- **Mouse:** Controla a direção do olhar.
  - Movimento horizontal: Gira a câmera para os lados (Yaw).
  - Movimento vertical: Olha para cima ou para baixo (Pitch), com limite de 89 graus para evitar inversão da câmera.
- **Cursor:** O cursor do mouse é ocultado e travado na janela para permitir rotação infinita (estilo FPS).

#### Outros Comandos

- **Tecla ESC:** Encerra a aplicação.
- **Setas Direita / Esquerda:** Aceleram ou desaceleram a passagem do tempo (ciclo dia/noite) para fins de demonstração.

### 4. Técnicas de Computação Gráfica Utilizadas

O projeto foi desenvolvido utilizando **OpenGL Moderno (Core Profile)** com shaders programáveis (GLSL). As principais técnicas implementadas foram:

#### 1. Geração de Terreno:

- a. Carregamento de malha irregular a partir de arquivo .OBJ (FBX models/terreno.obj).
- b. Aplicação de textura difusa (grass.png).

## 2. Iluminação Dinâmica (Ciclo Dia/Noite):

- a. Uma fonte de luz direcional simula o Sol, orbitando a cena de Leste para Oeste.
- b. A cor do céu (`glClearColor`) e da luz ambiente é interpolada dinamicamente baseada na altura do sol, criando transições suaves entre amanhecer, dia, entardecer e noite.
- c. Visualização da posição do sol através de uma esfera renderizada no céu.

## 3. Sombras em Tempo Real (Shadow Mapping):

- a. Implementação da técnica de **Shadow Mapping** em dois passos.
- b. **Passo 1:** Renderização da cena do ponto de vista da luz (Sol) para um *Framebuffer* de profundidade (Depth Map).
- c. **Passo 2:** Renderização da cena normal, comparando a profundidade do fragmento com o mapa de sombra para determinar oclusão.
- d. Uso de **PCF (Percentage-Closer Filtering)** nos shaders para suavizar as bordas das sombras.
- e. Correção de artefatos visuais (*Shadow Acne*) utilizando `glCullFace(GL_FRONT)` durante o passo da sombra.

## 4. Neblina Volumétrica (Fog):

- a. Cálculo de neblina exponencial quadrática no Fragment Shader.
- b. A cor da neblina adapta-se automaticamente à cor do céu (ciclo dia/noite), garantindo coesão visual.

## 5. Instancing e Modelos 3D:

- a. Carregamento de múltiplos personagens no formato FBX.
- b. Distribuição aleatória de 100 instâncias pelo terreno, com escala e rotação variadas.
- c. Ajuste automático de altura para garantir que os pés dos personagens toquem o solo corretamente.