

Implementación de una Escena 3D Realista con OpenGL y PyGame

Descripción

Este proyecto implementa una escena 3D realista utilizando OpenGL y PyGame, donde se pueden visualizar y manipular diferentes figuras 3D con transformaciones, iluminación y texturas.

Requisitos

- Python 3.10 o superior
- PyOpenGL
- PyOpenGL_accelerate
- Pygame
- NumPy

Instalación

1. Clonar este repositorio:

```
bash
```

```
git clone https://github.com/su-usuario/practica5-opengl.git  
cd practica5-opengl
```

2. Crear un entorno virtual (opcional pero recomendado):

```
bash
```

```
python -m venv venv  
# En Windows:  
venv\Scripts\activate  
# En macOS/Linux:  
source venv/bin/activate
```

3. Instalar las dependencias:

```
bash
```

```
pip install PyOpenGL PyOpenGL_accelerate pygame numpy
```

Estructura del Proyecto

```
practica5-opengl/
├─ main.py                # Archivo principal
├─ figures/               # Módulos para las figuras 3D
│   ├─ __init__.py
│   ├─ cube.py            # Implementación del cubo
│   ├─ pyramid.py         # Implementación de la pirámide
│   ├─ sphere.py          # Implementación de la esfera
│   ├─ cylinder.py        # Implementación del cilindro
│   └─ superellipsoid.py  # Implementación del superelipsoide
├─ utils/                 # Utilidades
│   ├─ __init__.py
│   └─ texture_loader.py  # Cargador de texturas
└─ textures/              # Directorio para las texturas
    ├─ cube.jpg
    ├─ pyramid.jpg
    ├─ sphere.jpg
    ├─ cylinder.jpg
    └─ superellipsoid.jpg
```

Ejecución

```
bash
```

```
python main.py
```

Controles

Menú Principal

- **1:** Mostrar Cubo
- **2:** Mostrar Pirámide
- **3:** Mostrar Esfera
- **4:** Mostrar Cilindro
- **5:** Mostrar Superelipsoide
- **6:** Salir del programa

Controles de Visualización

- **Flechas** (\leftarrow , \uparrow , \downarrow , \rightarrow): Rotar la figura sobre los ejes X/Y

- **W, S, A, D:** Trasladar la figura en los ejes X/Z
- **+/-:** Escalar la figura
- **R:** Reiniciar transformaciones
- **P:** Alternar entre proyección en perspectiva y paralela
- **T:** Activar/desactivar textura
- **I:** Activar/desactivar iluminación
- **ESC:** Volver al menú principal

Características Implementadas

1. Figuras 3D:

- Cubo
- Pirámide
- Esfera
- Cilindro
- Superelipsoide

2. Transformaciones:

- Traslación
- Rotación
- Escalado

3. Proyecciones:

- Perspectiva
- Paralela (Ortográfica)

4. Efectos de Realismo:

- Iluminación
- Texturizado
- Z-buffer para determinar superficies visibles

5. Interactividad:

- Control de cámara
- Manipulación de objetos
- Menú de selección

Notas sobre la Implementación

- La clase `Cube` implementa un cubo utilizando triángulos (`glBegin/glEnd`).
- La clase `Pyramid` implementa una pirámide de base cuadrada utilizando triángulos.
- La clase `Sphere` utiliza `gluSphere` para una representación eficiente.
- La clase `Cylinder` combina `gluCylinder` y `gluDisk` para el cuerpo y las tapas.
- La clase `Supere ellipsoid` implementa una superficie paramétrica más compleja.

Texturas

Para que el programa funcione correctamente, debe crear un directorio `textures` y colocar imágenes para cada figura con los nombres correspondientes. Si no se encuentran las texturas, se utilizará una textura predeterminada de tablero de ajedrez.

Autor

- [Tu Nombre] - [Tu Correo]

Licencia

Este proyecto se distribuye bajo la Licencia MIT. Consulte el archivo LICENSE para más detalles.