Hidden Surface Elimination Simulator

❖ Project overview :

This project is a 3D visualization web app built using Streamlit that demonstrates the concept of Hidden Surface Elimination in computer graphics. It allows users to add 3D objects to a scene , manipulate the camera position , and observe how objects appear differently depending on their placement and the viewpoint.

While it does not use real time rendering engine , it offers an educational and interactive way to understand how 3D scenes are viewed and how certain surfaces may be hidden based on the observer's perspective.

❖ Key Features :

- Interactive web interface using Streamlit
- Add 3D objects such as :
  - ★ Cube
  - ★ Sphere
  - ★ Pyramid
  - ★ Cone
  - ★ Cylinder
- Adjust camera position (X, Y, Z) and look at point to change the view
- Scene rendered using matplotlibs's 3D plotting tools
- Simulated hidden surface elimination via matplotlib's internal depth sorting

❖ Tech Stack :

- Python
- Streamlit - web app framework
- NumPy - Numerical computations

- Matplotlib - 3D rendering and plotting

❖ How it works :

- The app initializes a 3D scene and camera.
- Users interact with sliders to add objects and control the camera.
- The Renderer class uses matplotlib to display 3D plots.
- Camera orientation is calculated using atan2 and sqrt to derive azimuth and elevation angels.
- Objects are added as Poly3DCollection , and matplotlib performs basic depth sorting.

❖ How to Run :

1. Clone the Repo

   git clone https://github.com/your-username/hidden-surface-elimination.git
   cd hidden-surface-elimination

2. Install dependencies

   Pip install -r requirements.txt

3. Run the App

   Streamlit run app.py

OR

Click this link to run the web app :

https://hidden-surface-elimination-simulator-gs8fb7qrx3b3nyxywzxvqw.streamlit.app/

❖ Learning Outcomes :

- Understanding the basics of 3D rendering in Python
- Working with camera positioning and projection
- Simulating hidden surface elimination using depth sorting
- Using Streamlit for interactive educational apps

❖ Future Improvements :

Implement real Z - buffer algorithm for pixel accurate visibility

Add back face culling

Export rendered scenes

Animate camera movements