Assignment on OpenGL:

Spider animation

Submitted in partial fulfillment of the requirements for

IS F311 - COMPUTER GRAPHICS

PREPARED FOR

Dr. Sundaresan Raman
Department of CSIS

PREPARED BY :

Sanju S: 2021A7PS2537P

Ohiduz Zaman: 2021A7PS2005P

Project Report: Spider Animation

Objective:
The objective of this project was to create an animated spider model using OpenGL libraries such as GLUT, GLU, GL and integrate it into an interior scene. The spider model is designed to have articulated legs that move in a realistic manner, and the scene includes furniture and walls to create a room-like environment. The spider would be controlled by key presses given from the user. Detailed instructions regarding the key mapping are given in the report.

Libraries Used:

- GL: Open Graphics Library
- GLU :OpenGL Utility Library
- GLUT :OpenGL Utility Toolkit
- Stb_image.h : To load textures

Implementation Details:

1. Spider Animation:
    - The spider model consists of a body and articulated legs, designed using hierarchical modeling. The body is represented by a sphere, and each leg is composed of two segments connected by a hinge joint.
    - Leg movement is achieved through a combination of translation and rotation. Each leg moves forward and backward in a coordinated manner to simulate walking.
    - Key functions include `drawSpider()` for drawing the spider model, `animateSpiderLegs()` for animating the leg movement, and keyboard functions for controlling the spider's position and orientation.

2. Interior Scene:
    - The scene includes walls, a floor, and furniture items such as a table, bed, and cupboard.
    - Walls and furniture are represented as cuboid primitives with appropriate scaling and positioning to create the desired layout.
    - Lighting is implemented using ambient light to illuminate the scene uniformly.

3. User Interaction:
The user can control the position and orientation of the spider using keyboard inputs:
    - Movement: Forward, backward, left, and right.
    - Rotation: Clockwise and counterclockwise.
    - Leg Extension: Extending legs forward and backward.
    - Additional keyboard commands are provided for adjusting the camera position and zoom level.

Challenges Faced:

1. **Integration of Ready-Made Spider Model:** Initially, we attempted to import a ready-made spider model of various formats into OpenGL. However, we encountered numerous errors related to integrating and resolving include paths, as well as dependency issues in C++. Despite several attempts, we were unable to resolve these issues satisfactorily. As a result, we decided to design the spider from scratch in OpenGL using simple primitives such as spheres and cylinders.
2. **Animation Realism:** Achieving realistic leg movements for the spider proved to be a significant challenge. We experimented with various combinations of geometric transformations, including rotation, translation, and scaling, to simulate leg movements realistically. This process involved extensive trial and error with different parameters for the transformations until the desired animations were achieved.
3. **Lighting and Coloring:** We encountered difficulties with lighting and coloring, particularly with the `glColor3f` function not producing the desired results. To address this issue, we had to find a workaround by adjusting parameters for diffusion and ambient lighting to provide the intended colors to various components of the spider model.

Techniques Used:

1. **Designing Spider from Scratch:** To overcome the challenges associated with integrating a ready-made spider model, we opted to design the spider entirely from scratch in OpenGL. By using simple geometric primitives such as spheres and cylinders, we were able to bypass the complexities of model importation and achieve greater control over the design and animation of the spider.
2. **Trial and Error for Animation:** We employed a trial-and-error approach to fine-tune the parameters for geometric transformations and achieve realistic leg movements. By iteratively adjusting rotation angles, translation distances, and scaling factors, we were able to refine the animations until they closely resembled natural spider movements.
3. **Adjusting Lighting Parameters:** In response to issues with coloring and lighting, we focused on adjusting parameters for diffusion and ambient lighting. By carefully calibrating these parameters, we were able to achieve the desired colors and enhance the visual quality of the spider model despite the limitations of certain OpenGL functions.

Conclusion:
The project successfully demonstrates the implementation of an animated spider model within an interior scene using OpenGL and GLUT. The spider's movement and interaction with the environment create a visually engaging simulation. Further enhancements could include additional animations, textures, and more detailed models for the environment elements.

Future Scope:
- Integration of texture mapping to all/most components to add realism to the scene.
- Implementation of collision detection to enable interaction between the spider and the environment.
- Optimization for performance and compatibility with different hardware configurations.

Acknowledgments:
The project acknowledges the contributions of the OpenGL and GLUT development communities for providing the necessary tools and resources for graphics programming.

This report summarizes the key aspects of the project, highlighting the objectives, implementation details, user interaction, conclusions, and future scope.

Controls:
- Arrow Keys: Movement of Spider
- Shift +(Left Arrow / Right Arrow) : Rotate Spider
- “H” : Shoot web up and climb
- “L” : Climb down
- “W”: Extend legs forward
- “S” : Extend legs backward
- “A” : Look Left
- “D” : Look Right
- “I” : Zoom In
- “O”: Zoom Out
- “1” : Turn Light On/Off
- ‘F’, ‘C’, ‘V’, ‘B’ : Eye Movement
- ‘J’, ‘N’, ‘M’, ‘,’ : Reference point movement

Run Instructions:
make
./gl