

# **3-D Simulation of Spider**

**CS352: Computer Graphics  
& Visualization Lab**

**Project Report**

**Course Instructor:**

**Dr. Somnath Dey**

**Submitted By:**

Kanchi Pardhi - 200001032

Neha - 200001051

Saurabh Kumar Singh - 200001071

## ❖ Introduction

### 3-D Simulation of Spider

In this project, we have created a spider and shown its 3-D simulation in a room and on a web.

It can walk forward with four different speeds ( 2, 4, 6 and eight), rotate clockwise and anticlockwise, can zoom in and zoom out with some sound effects also when it hits the boundaries of the web and the room.

While moving it clearly shows all the leg movements. We can see the leg movements separately also for the front left and right leg.

And up-down movement of all the legs simultaneously is also implemented.

For creating its body, we used two spheres and joined them.

For creating each leg we used three cuboidal pieces and three spheres and joined them precisely with each other and the main body i.e. the two spheres.

For texture, we used the SOIL library to render an image on its skin.

For seeing the key controls a separate info button is also provided in the bottom right corner. Along with it, a cross button is also provided for the user to go back to the previous window.

The coordinates of the current position of the spider are also shown on the left side of the window.

We can also change the lighting of the screen using key controls. For now, two effects, light and dark are provided.

We have applied constraints on the movement and scaling of the spider so that it doesn't get out of the view volume.

By dragging the mouse, we can change the view perspective and can have a look of the room and the web from different angles.

In this project, we have implemented both keyboard control and mouse control features. We have also rendered some text on the screen for user's convenience.

We can also change the appearance of the spider i.e. red and black using key control.

In this project nearly all the movements of the spider along with constraints are properly implemented. It is a very user-friendly application. User can see all the controls whenever he wants right there on the screen where he is currently. Overall, the spider looks great and can cover all the window positions efficiently.

## ❖ Specifications

### ➤ Libraries required:

- **iostream**  
Defines the standard input/output stream objects
- **Glut**  
The OpenGL Utility Toolkit (GLUT) is a window system-independent toolkit for creating OpenGL programmes. It implements an OpenGL windowing application programming interface (API).
- **stdlib**  
specifies many utility functions for type conversions, memory allocation, algorithms, and other related use cases.
- **math**  
Defines various mathematical functions
- **unistd**  
Defines a variety of symbolic constants and types, as well as defining a variety of functions.
- **SOIL**  
Simple OpenGL Image Library  
C library for OpenGL texture loading, writing, and processing.
- **irrKlang**  
For adding sound.

### ➤ How to run the project:

- For enabling the display in GLU run the following two commands (if you are using WSL) :
  - export DISPLAY\_NUMBER="0.0"
  - export DISPLAY=\$(grep -m 1 nameserver /etc/resolv.conf | awk '{print \$2}'):DISPLAY\_NUMBER
- For compiling main.cpp, run the following command : (Optional)
  - g++ main.cpp -o spider -lglut -lGLU -lGL -lSOIL -w
- This will create an executable file “spider”.
- Now run the command “make && ./spider”. This will open the GLU display.

### ➤ Details of the Key Controls :

#### Key controls outside home/web view :-

- h : Home (This will open home view)
- w : Web (This will open web view)
- e : Exit (Termination)

#### Key controls inside home/web view :-

- a : third joint inwards of front left leg
- A : third joint outwards of front left leg
- b : first joint inwards of front left leg
- B : first joint outwards of front left leg
- c : second joint inwards of front left leg
- C : second joint outwards of front left leg
- d : third joint inwards of front right leg
- D : third joint outwards of front right leg
- e : first joint inwards of front right leg
- E : first joint outwards of front right leg
- f : second joint inwards of front right leg
- F : second joint outwards of front right leg
- l : switching between light and dark
- u : zoom in
- U : zoom out
- w : spider colour will change to black
- W : spider colour will change to red
- x : all the legs inwards at first joint
- X : all the legs outwards at first joint
- y : all the legs inwards at third joint
- Y : all the legs outwards at third joint
- z : all the legs inwards at second joint
- Z : all the legs outwards at second joint
- UP KEY : move forward
- DOWN KEY : move backward
- LEFT KEY : rotate anticlockwise
- RIGHT KEY : rotate clockwise
- > : for increasing the speed till 8 (initially the speed is 2, then it will become 4,6 and 8 after pressing the button consecutively)
- < : for decreasing the speed till 2 (pressing the button consecutively will decrease the speed by 2, minimum speed is 2)

**> Mouse Controls :**

- We can also click on the home, web and exit button instead of using the allotted keys.
- Mouse controls : we can change the camera view of the web and the room.
- Cross button on the top right corner : to get back to the previous window
- Info button on the bottom right corner : to see the key controls

## ❖ Functionalities Implemented

This project has many functionalities implemented related to movement, colour changing, size changing, environment changing, rotation, camera rotation, constraints, etc with proper keyboard and mouse controls.

These are explained in detail below :

- **Spider structure making**
  - Made two spheres small for spider's head and large for spider's body using gluSphere() function
  - For leg making, used glutSolidCube() to make a bone and added spheres at the joints.
  - Connected the legs to the body
- **Leg movement(bone position changing):**
  - For leg movements we have changed the angle between the bones and the angle at the joint where the leg is connected to the spider body, each leg has 3 bones.
- **Translation movement with different speeds:**
  - For displaying the translational movement, we have changed the x and y coordinates of the spider. And we can change the speed by changing the distance by which the spider should move when we press the required key for translation a single time.
- **Rotation**
  - We are rotating the spider 15 degrees in a clockwise and anticlockwise direction as per the key pressed.
  - glRotatef(rz, 0, 0, 1);
  - Rz is the angle of rotation, here rotation is about the z axis.
- **Overall Movement**
  - Combined leg movement with translation. So when we press the key for forward movement then first the legs move (angle between joint changes) and then spider is translated forward , similar is done for backward movement. For moving in any other direction we can first rotate the spider and then apply forward/backward movement.
- **Different skin textures**
  - We have stored two different textures in the texture array for skin.
  - texture[3] = SOIL\_load\_OGL\_texture("img/skin3.png", SOIL\_LOAD\_AUTO, SOIL\_CREATE\_NEW\_ID, SOIL\_FLAG\_INVERT\_Y);
  - texture[5] = SOIL\_load\_OGL\_texture("img/skin2.png", SOIL\_LOAD\_AUTO, SOIL\_CREATE\_NEW\_ID, SOIL\_FLAG\_INVERT\_Y);
  - Then use glBindTexture(GL\_TEXTURE\_2D, texture[sk]) to apply the texture we are changing the sk value as per key pressed.

- **Speed up down**
  - By default the spider speed is 2; we can change it to 4, 6, and 8 and then can decrease it back to 2 using the ‘>’ and ‘<’ keys respectively.
  - To achieve this we will make the spider translate more coordinates.
- **Lightning**
  - There are two lightning modes i.e. dark and light. We will use the ‘l’ key for switching between these themes. It will change the image that is rendered on the skin of the spider using the SOIL library.
  - The function used is **glDisable(GL\_LIGHTING)** and **glEnable(GL\_LIGHTING);**
- **Scaling up and down the spider’s size**
  - We have scaled up and down the spider using the **glScalef** function.
  - **glScalef(zooms,zooms,zooms)** // initially set zooms to 1
  - For scale-up => **zooms=min(1.15,zooms+0.05);**
  - For scale down => **zooms=max(0.85,zooms-0.05);**
  - Here we have added the upper bound and lower bound to the scaling factor
- **Info button**
  - For user’s convenience we have provided menus in both the web and the home pages. It will show the list of all the key controls and the mouse controls also.
  - There is a button on the bottom right side of the pages, pressing which will take you to the info page.
  - We have used the text rendering on the screen feature i.e.
  - **glutBitmapCharacter()**
- **Close button**
  - There is a cross button on the top right side of the windows using which the user can go back to the previous window.
  - We will set the variables used for the respective windows to false and set the variable used for the previous window to true and will call the display function again.
  - **glutPostRedisplay();**
- **Sound**
  - If the spider tries to get out of the boundaries of the window, it will not be allowed to do that and it will create a sound similar to that is created during collision of two solid objects.
  - For this we have used the library **#include <irrKlang.h>**
  - The code snippet is :

```
ISound* sound=engine->play2D("sound/click.wav", true,false,true);
    std::this_thread::sleep_for(std::chrono::milliseconds(100));
    sound->setIsPaused(true);
    sound->drop();
```
- **Constraints**
  - Spider is restricted to move only within a certain region.
  - To implement this feature we will check the spider’s coordinate and the angle and accordingly will disable the keys that will make the spider move out of the window when pressed.

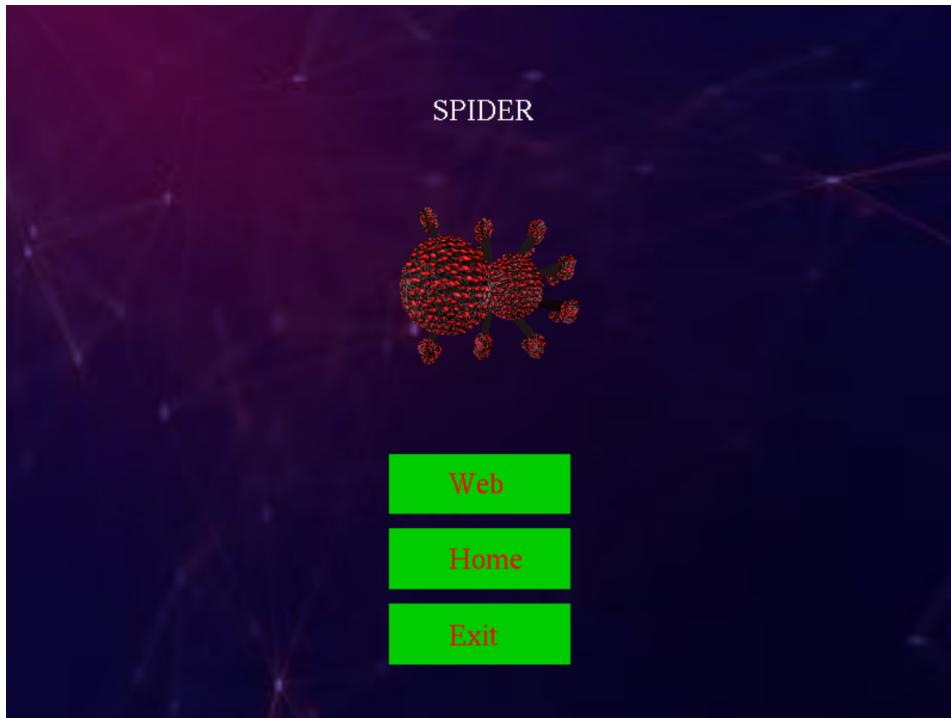
## ❖ References

<https://www.glprogramming.com/red/chapter05.html>

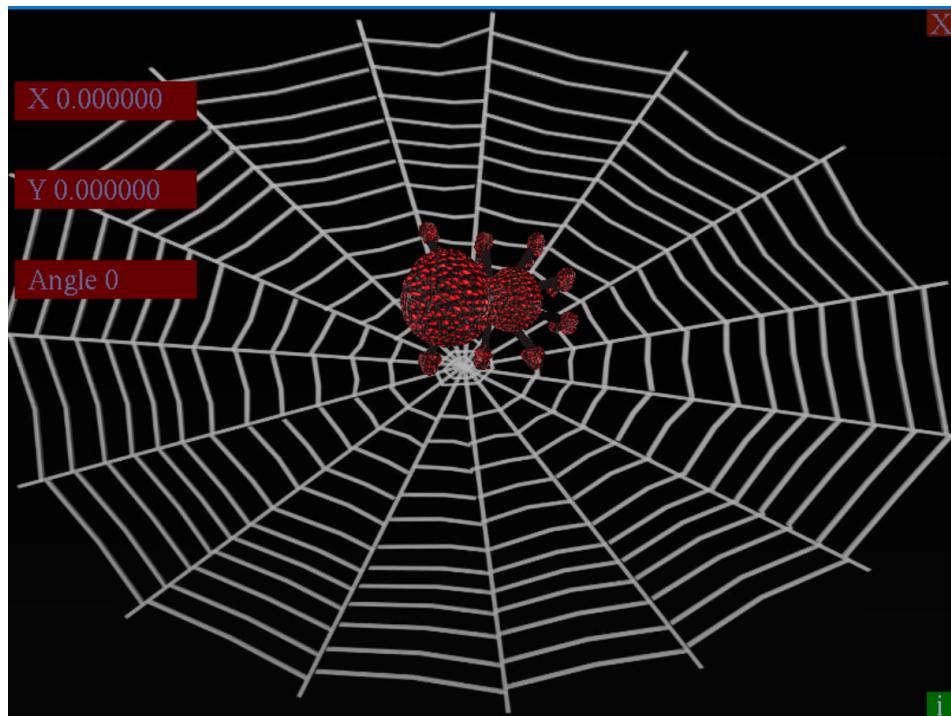
<https://learnopengl.com/In-Practice/2D-Game/Audio#:~:text=OpenGL%20doesn't%20offer%20us,for%20use%20in%20our%20game>

## ❖ Output

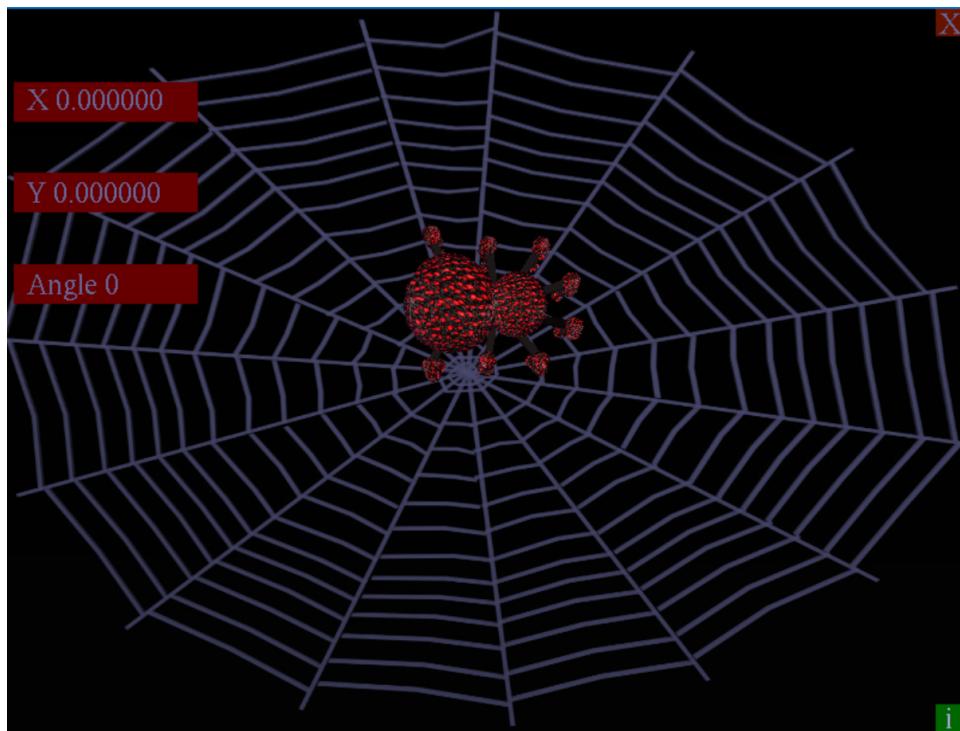
Initial page



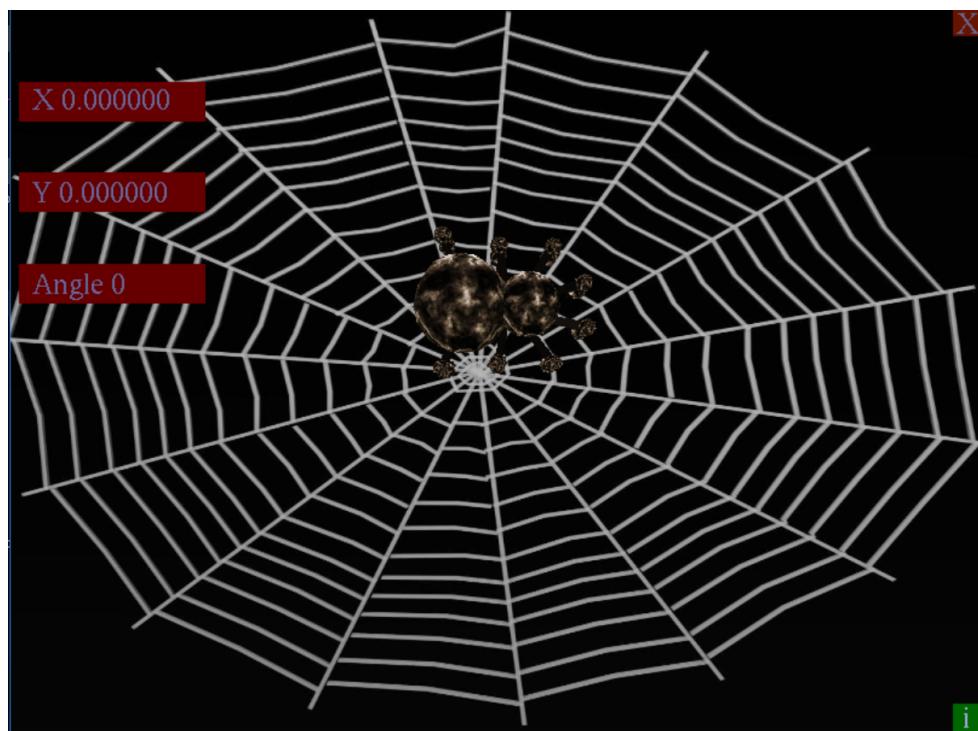
Web (Light Mode)



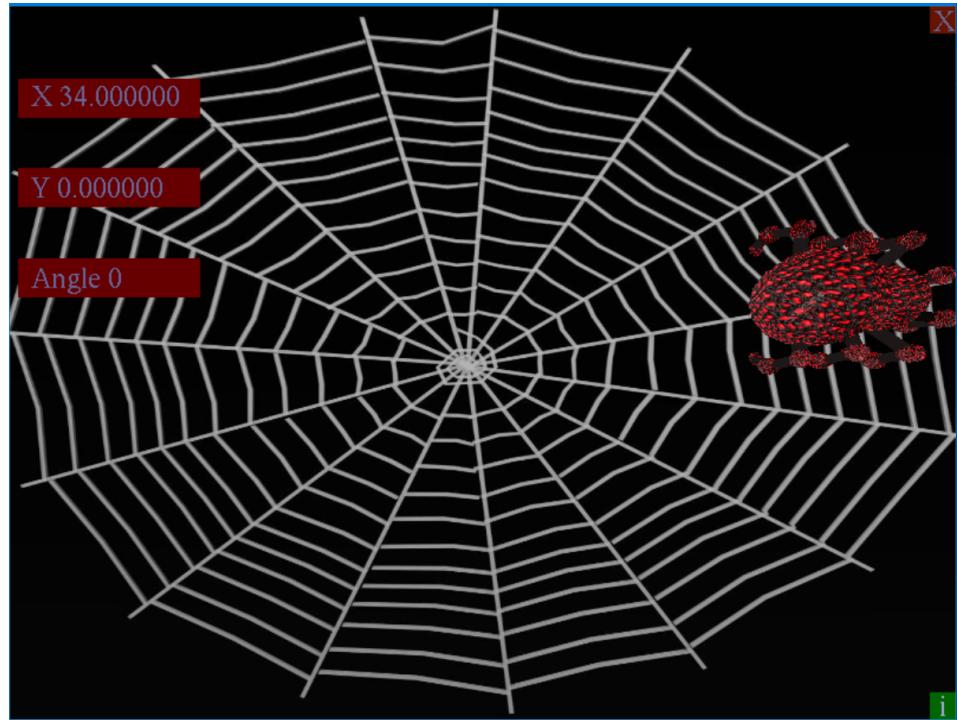
### Web (Dark mode)



### Skin Colour change



## Movement of the spider (constraints set, can't move outside the window)



## Movement along with rotation



## Scaled UP



## Spider in Room (Light Mode)



## Spider in Room (Dark Mode)



## Camera Rotation



## Spider Movement along with Scaled UP



## Scaled Down



## Skin Colour Change



## Info Page

### Instructions

h : Home  
w : Web  
e : Exit  
a : 3rd joint Inwards of front left leg  
A : 3rd joint Outwards of front left leg  
b : 1st joint Inwards of front left leg  
B : 1st joint Outwards of front left leg  
c : 2nd joint Inwards of front left leg  
C : 2nd joint Outwards of front left leg  
d : 3rd joint Inwards of front right leg  
D : 3rd joint Outwards of front right leg  
e : 1st joint Inwards of front right leg  
E : 1st joint Outwards of front right leg  
f : 2nd joint Inwards of front right leg  
F : 2nd joint Outwards of front right leg  
I : switching between light and dark  
u : scale up  
U : scale down  
w : Spider color will change to black  
W : Spider color will change to Red  
x : all the legs Inwards at 1st joint  
X : all the legs Outwards at 1st joint  
y : all the legs Inwards at 3rd joint  
Y : all the legs Outwards at 3rd joint  
z : all the legs Inwards at 2nd joint  
Z : all the legs Outwards at 2nd joint  
UP KEY : Move Forward  
DOWN KEY : Move Backward  
LEFT KEY : Rotate Anticlockwise