Jonathon Moore

CST-310

Professor Citro

5/7/2024

Topic 1 Lab Question 2

To implement a turtle graphics library in OpenGL it would take some thinking on how you want it setup. Usually in a turtle graphics library you have a “turtle” that will draw a line as it moves across the screen, and it is up to the user to control where the turtle moves, how it rotates, and picking its pen up and down for drawing. In the context of OpenGL you could implement these five simple functions to get started:

1. Setting the position:

Using this method will set the current position of the turtle on the screen and move it there instantaneously by changing its x and y coordinates.

Void SetPosition(float x, float y)

{

turtlePosX = x;

turtlePosY = Y;

}

1. Setting the rotation:

Using this method will take in a float as the angle and from there will set the turtles direction to that angle.

Void setRotation(float angle)

{

turtleAngle = angle;

}

1. Moving a certain distance:

Using this method will move the turtle a certain distance denoted “distance” in the current angle of the turtles direction. It will also draw a line if the turtles pen is down.

Void moveForward(float distance)

{

Float rad = turtleAngle \* M\_PI / 180;

Float newX = turtlePosX + distance \* cos(rad);

Float newY = turtlePosY + distance \* sin(rad);

If (penDown)

{

// Draw line from old position to new position

}

turtlePosX = newX;

turtlePosY = newY;

}

1. Setting the turtles pen to up position:

This method will deactivate drawing lines by setting the turtles pen to the up position or false.

Void penUp()

{

penDown = false;

{

1. Setting the turtles pen to down position:

This method will activate drawing lines by setting the turtles pen to the down position or true.

Void penDown()

{

penDown = true;

{

With these five different methods you could have a fully functioning turtle graphics library, although quite trivial it can be used as a building block for much greater applications.