Project 6 – Shaders

Manoj Kumar Gummadi
gummadm@oregonstate.edu

December 1st, 2023

Goal: The goal of this project is to create an animated ellipse pattern by using an OpenGL fragment shader. The center of the ellipse will be animated using the KeyTime method you already know. The S-radius and T-radius will be animated using the Time variable and some functions that you invent. Light the object using per-fragment lighting.

Description: Below are the steps below that I used to achieve the project's goal.

1. **Shader Configuration and Variables**

   - Used **pattern.frag** and **pattern.vert** files for shader editing.
   - Controlled ellipse center position using **S0** and **T0**, aliasing them to **uS0** and **uT0**.
   - Managed ellipse radius on semi-major and semi-minor axes with **rX** and **rY**, aliased to **uD** and **uD2** via **Pattern.SetUniformVariable()**.

2. **Ellipse Equation and Color Assignment**

   - Implemented standard ellipse equation in the fragment shader: **((s - uS0) / uD)^2 + ((t - uT0) / uD2)^2 <= 1**.
   - Assigned RGB colors based on vertex position:
     - Inside: Red (1, 0, 0).
     - Outside: Orange (1, 0.5, 0).

3. **Animations Setup**

   - Utilized **key-time animations** for center movement in a triangle-shaped path over 10 seconds.
   - Employed **time-equation**-based animation to control ellipse radii.

4. **Key-Time Animations**

   - Implemented 5 key-time animations for x-coordinate control of the ellipse center.
   - Employed 5 additional key-time animations (with distinct values) for y-coordinate control.
   - Thus, a total of 10 key-time animations regulate the movement and positioning of the ellipse center.

5. **Key-Bindings and Controls**

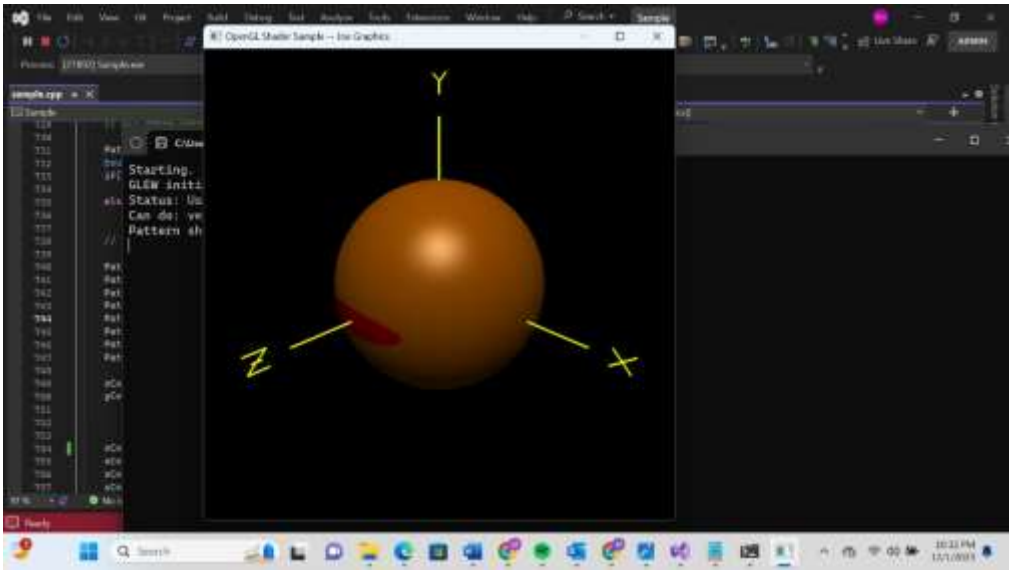
- Key bindings for enabling/disabling animations:
- 't'/'T' for radius animation.
- 'k'/'K' for center movement animation.

Key Time Values:

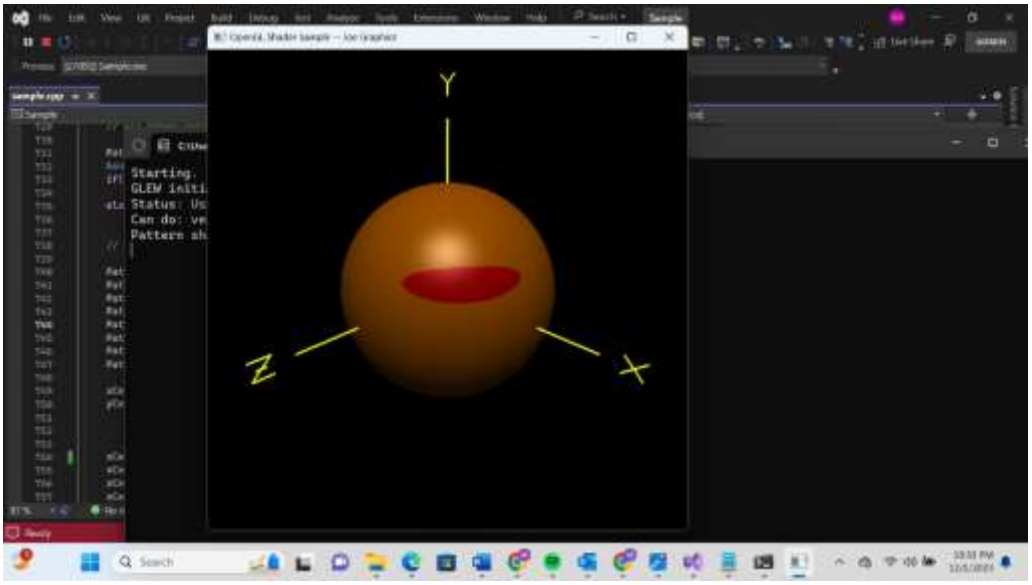
```
xCenter.AddTimeValue(0.0, 0.5);
xCenter.AddTimeValue(2.5, 0.6);
xCenter.AddTimeValue(5.0, 0.75);
xCenter.AddTimeValue(7.5, 0.6);
xCenter.AddTimeValue(10.0, 0.5);

yCenter.AddTimeValue(0.0, 0.5);
yCenter.AddTimeValue(2.5, 0.52);
yCenter.AddTimeValue(5.0, 0.75);
yCenter.AddTimeValue(7.5, 0.6);
yCenter.AddTimeValue(10.0, 0.5);
```

Kaltura Video Link: https://media.oregonstate.edu/media/t/1_m3akf487

Initial Position



Key Time Animation Enabled



Time Equation Animation Enabled