Oregon State University

Shaders Final Project

Hyuntaek Oh

ohhyun@oregonstate.edu

Due: March 17, 2025

1 Description

1.1 Set up

GLSL and glman are used to render two different dinosaurs; one is a noise-rainbow Argyrosaurus and another is an explosive Triceratops with cube-mapping in the Shaders Final Project.

1.2 Program Description

The program is implemented by these concepts:

- Cube Mapping
- Noise Effect
- Color blending - (Rainbow) *
- Weak Reflection/Refraction *
- Explosive effect * * means new added

Some concepts such as Rainbow, Reflection/Refraction, and Explosive effect are not in the final project's initial proposal. Initially, I planned color blend with just two colors, but it looks too simple and tedious to implement a colorful dinosaur. So, I changed two-color blending to rainbow color blending. Any specific details are in the explanation about each concept.

the number of dinosaurs was reduced to two (initially three) since the downloaded 3-D object files for dino had vertex issues (I did not know what this error, which cannot see, is exactly; maybe that may happen when reading vertex coordinates). What's more, adding multiple objects caused some errors, such as not matching with other files' information in the vertex and fragment shaders.

Figure 1 shows the background images which are used for cube mapping. These images are from the website "Humus - Textures" (`https://www.humus.name/index.php?page=Textures&start=0`).

Figure 1: Cube Mapping

One of the 3-D objects is noise-rainbow blending Argyrosaurus. To implement the noise-rainbow effect, the following code is used:

float rainbow = sin(uNoiseFreq * (vMC.x + vMC.y + vMC.z)) * 0.5 + 0.5;

vec3 rainbowColor = vec3(sin(rainbow * 6.28318),

sin(rainbow * 6.28318 + 2.094), sin(rainbow * 6.28318 + 4.188)); ,

rainbowColor = (rainbowColor + 1.0) *0.5;

color = mix(color, rainbowColor, 0.5);

gl_FragColor = vec4(color, 1.);

These codes can display rainbow color with a noise effect. When uNoiseFreq increases, Argyrosaurus's appearance is changed to the degree of uNoiseFreq. What's more, Argyrosaurus can be transparent or reflective, depending on the value of uMix, which can control the weak reflection/refraction.

Another is an explosive Triceratops consisting of many triangles, which is based on the small points. The implementation of the Triceratops is on a geometric shader. By using the function called "ProduceVertex" in the lecture note, the number of points, the shape of many small triangles, and the degree of distributed points are affected by the variables uLevel, uGravity, uTime, and uVelScale.

It seems not to be an example of cleverness, but rainbow color can be implemented with the mathematical functions mentioned above.

By doing this, I learned that Rainbow color blending can be implemented with mathematical equations such as using the sine function, and the explosive effect is not easy to implement since I do not even understand the meaning of uTime and uGravity. When their values are changed, dino moves up or down with the speed based on the input values (interesting). What's more, I also learned almost all visual effects are based on mathematics.

1.3 URL

Video Link(bitly): `https://bit.ly/4hE8WWC`

Video Link(original):
`https://media.oregonstate.edu/media/t/1_oox1emei`

1.4 Test Result

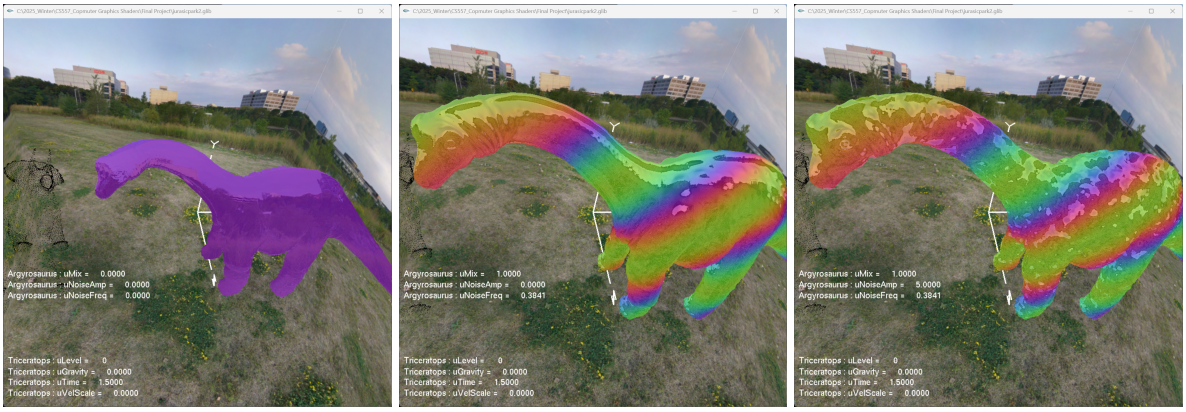

Figure 2: Rainbow Argyrosaurus

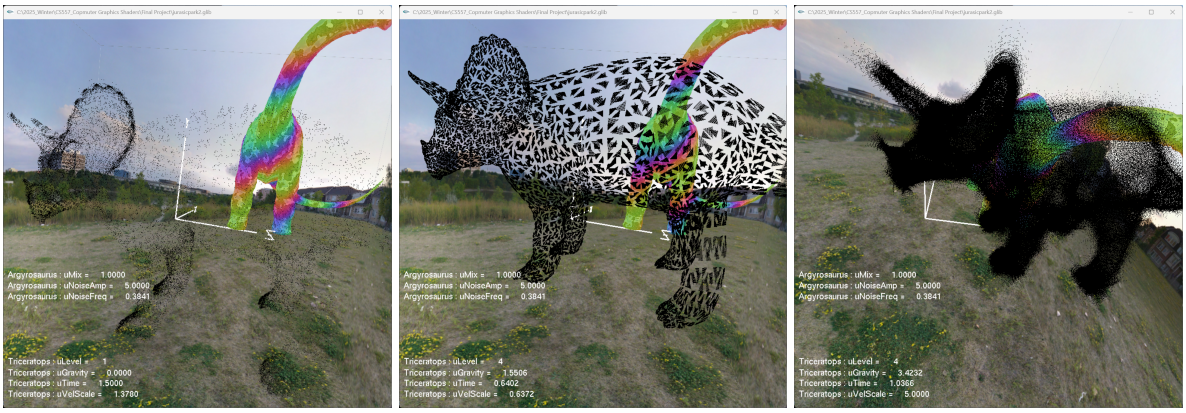

Figure 3: Explosive Triceratops