Hao Truong
truongh@oregonstate.edu

Shaders Final Project

Copy of Proposal:

For the final project, I plan to create a dynamic Aurora Borealis simulation. This will involve developing eye-catching effects that can perfectly portray the magnificent phenomenon. The goal is to create realistic animations and colors that can perfectly replicate the auroras' appearance in the sky.

My approach will include several key components:

- Noise Generation: I'll be using Perlin or Simplex noise to craft the organic, flowing patterns that are characteristic of the aurora.
- Color Gradients: I aim to dynamically create color transitions that emulate the natural light displays of auroras, complete with their varying intensities and hues.
- Vertex Manipulation: By adjusting vertex positions in real-time, I plan to simulate the undulating motion of the aurora across the sky, adding a layer of depth and realism to the simulation.

In order to add interactivity and realism to the simulation, I intend to use a controllable source of light. This will allow me to adjust the lighting conditions and observe the subtleties of the aurora.

The simulation will render onto a simple geometric shape, such as a quad, which provides a wide canvas to best display the complex interplay of light and color. The project will primarily focus on:

- Noise Maps: These will generate the base patterns that dictate the movement and shape of the aurora
- Color Maps: Used to define the aurora's varying colors, potentially enhanced through texture mapping to deepen the realism of the color transitions.

In the event that time permits, I plan on adding elements from the environment to the simulation, such as stars and other celestial objects. This can complement the effect of the Northern Lights by adding a realistic sky background. In addition, I'm also considering creating a weather system, such as clouds or fog, that can dynamically affect the simulation

Description of what I did:

In order to create the aurora effect, I implemented a simple perlin-like noise function along side a simple hash function to generate a pseudo-random noise patterns. This create a shimmering, fluid-like appearance for the aurora. I also created a sine wave for dynamic patterns. My code used sine function to create wave patterns across the aurora.

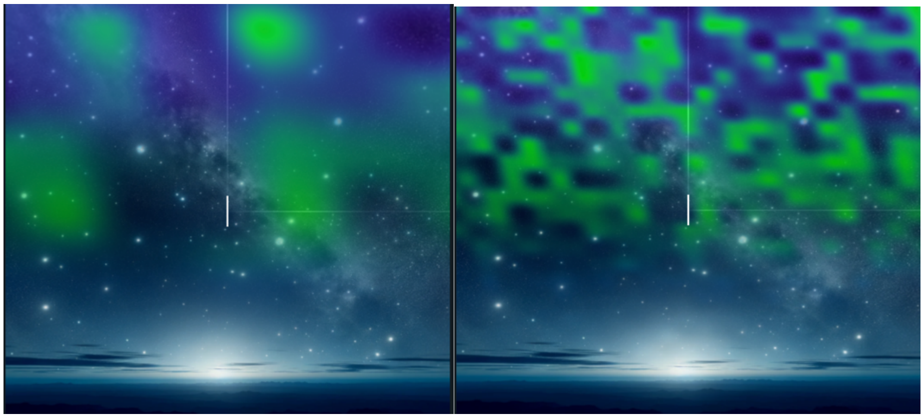
Hao Truong
truongh@oregonstate.edu

The color gradients are dynamically generated by the shader using the noise value and the vertical position (vST.y). They are then used to smoothen the transitions between the different colors in the Aurora. It achieves this by blending different tones such as purple, blue, and green depending on their vertical position and noise to enhance the aurora's visual complexity.

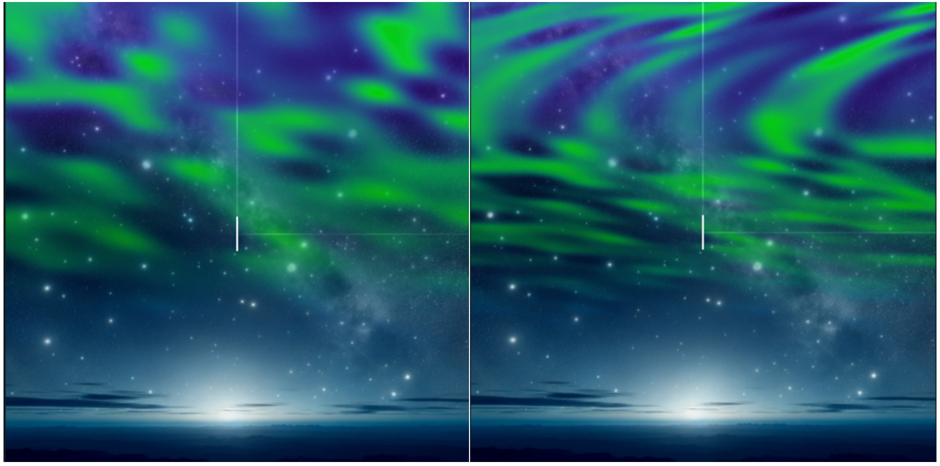
For the animation of the aurora, I utilized time-based modulation in the shader to animate the aurora. It allowed me to create a shift in its colors and a sine wave movement.

The last step in my shader was to create an integration between the aurora effect and a background texture. This blending considers the vertical and intensity positioning of the light and ensures that it's visible where it should be.

uNoiseScale:



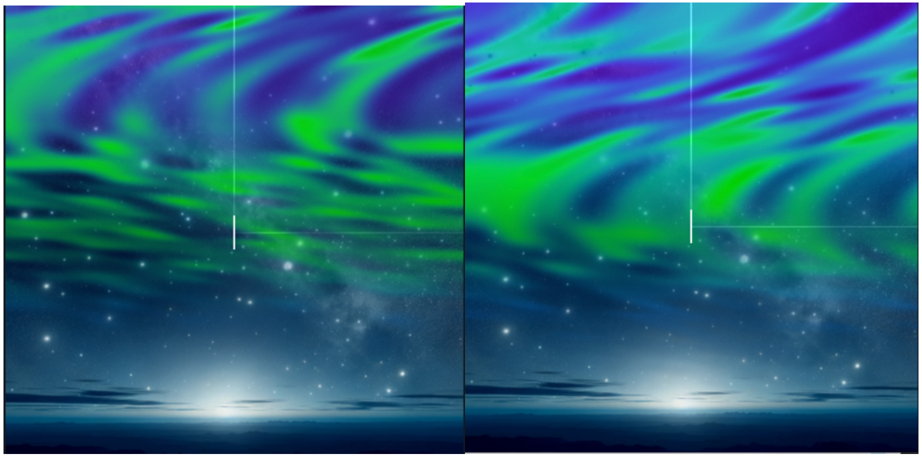
uFrequency:



uWaveSpeed just controls the speed of the waves.

Hao Truong
truongh@oregonstate.edu

uIntensity:



Here is the link to my media: https://media.oregonstate.edu/media/t/1_8wv5rma0