Omar Mena - 185918
*CEMSE*
*KAUST*

**December 9, 2024**

**FINAL PROJECT REPORT: — Volumetric cloud rendering using Ray Marching**

# 1 Introduction

We initially aimed to render realistic clouds in real time using WebGPU and compute shaders, focusing on volumetric rendering with ray marching, noise-based density generation, and dynamic cloud animation. However, as the project progressed, it was divided into two parts. The first part involved generating a density field within a bounding box and creating points in the compute shader. The second part focused on procedural cloud generation using Fractional Brownian Motion (FBM). The main challenge originated from merging these two concepts into a cohesive solution, ultimately keeping them separate components.

# 2 Technique and Framework

## 2.1 Volumetric Cloud Rendering with Ray Marching[1][2]

Our approach to rendering realistic clouds relied on ray marching within a volumetric cloud field to sample 3D noise and create natural cloud formations with realistic depth and softness. Cloud density was achieved by defining and sampling a noise-based density field, enabling varied and layered cloud shapes.

## 2.2 Noise-Based Density Generation and Animation

We used Perlin and Worley noise functions to define the cloud density field, generating clouds' texture and volume. By layering multiple noise scales, we introduced variability in cloud shapes and thickness, enhancing realism. Dynamic cloud animations were achieved using compute shaders, allowing real-time cloud density updates and movement parameters.

## 2.3 Approach and Challenges

Our implementation followed an iterative process:

- **Bounding Box and 3D Texture Experiments:** We began by generating a bounding box with random points in a compute shader, storing them in a buffer, and passing them to a 3D texture. However, the resulting patterns were overly structured, failing to achieve the intended randomness.
- **Geometry Modifications:** To address this, we modified the geometry of the points using both compute and vertex shaders. While this improved distribution, it became computationally prohibitive when scaling to multiple objects.
- **Ray Marching in the Bounding Box:** We implemented ray marching within the bounding box to sample a density field. Although this produced a single large cloud, it lacked the complexity of smaller, distinct cloud objects, as the 3D patterns were challenging to construct.
- **Fragment Shader Noise Generation:** Finally, we generated the cloud pattern directly in the fragment shader. This output served as a noise source for the 3D texture, offering finer control over patterns. However, integrating this method into the overall volumetric framework remains an ongoing challenge.
  - **Bounding Box and 3D Texture Experiments:** We began by generating a bounding box with random points in a compute shader, storing them in a buffer, and passing them to a 3D texture. However, the resulting patterns were overly structured, failing to achieve the intended randomness.

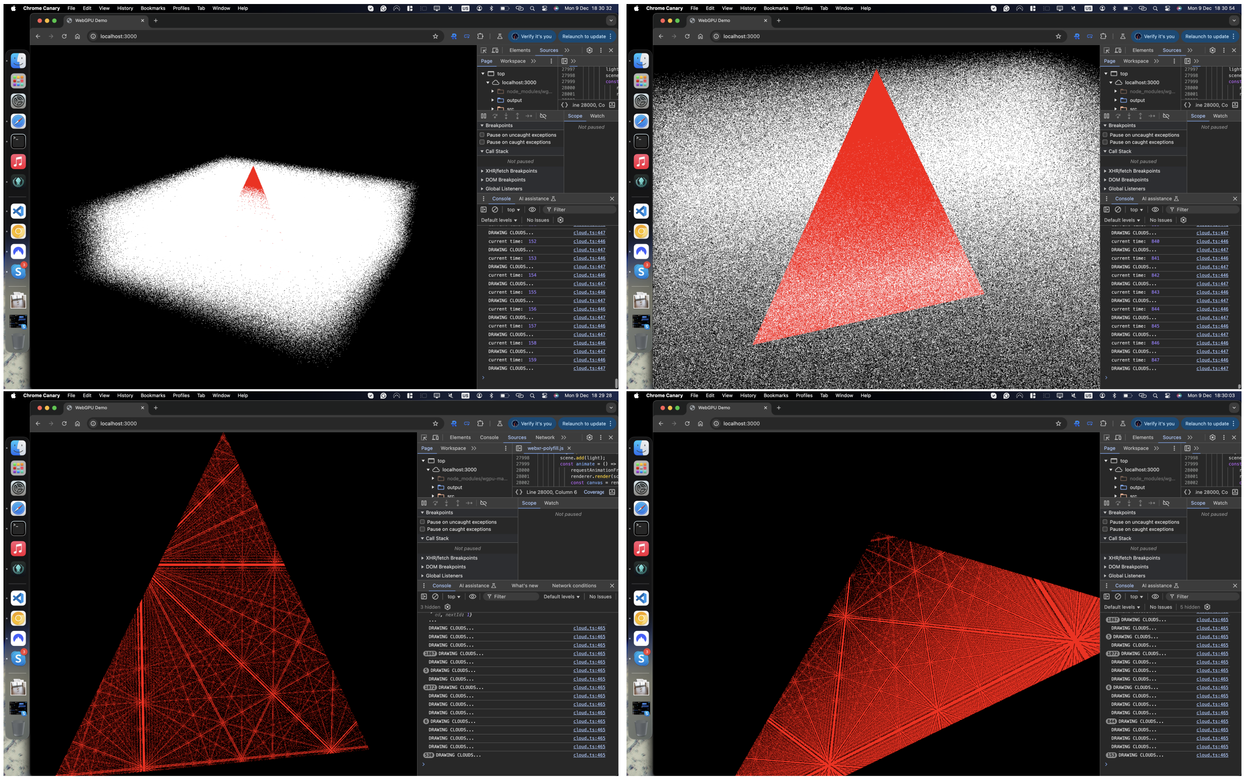

Figure 1: **Top images** show the density field randomly generated. Bottom images **Bottom imagers** show a strongly patterned point distribution and artifacts while testing.

# References


[1] Andrew Reeves and Matthew Hartner. Volume rendering techniques. In Randima Fernando, editor, *GPU Gems*, chapter 39. Addison-Wesley, 2004.

[2] Alan Hargreaves. Creating a volumetric ray marcher, 2017. Accessed: 2024-10-29.


*Submitted by Omar Mena - 185918 on December 9, 2024.*

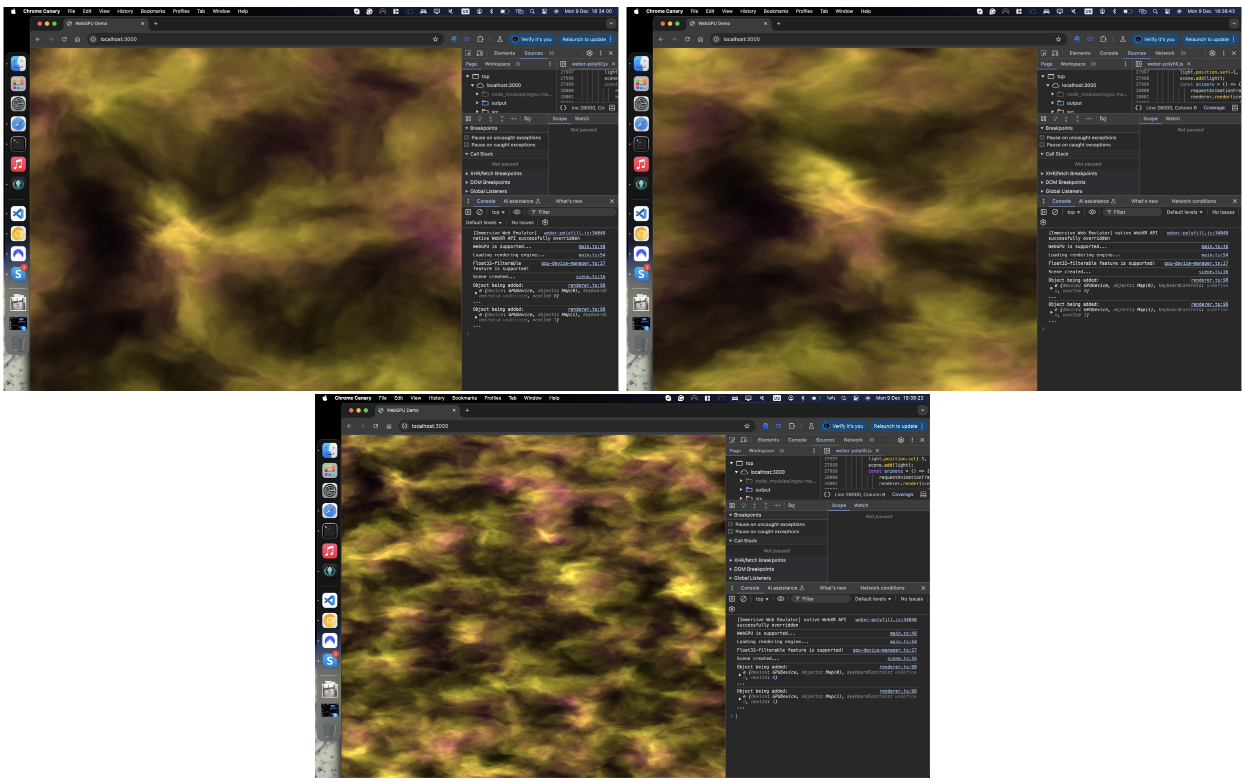

Figure 2: Different Level of Detail based on the resolution of the quad we generate.