Aman Sachan & Meghana Seshadri
CIS565 Final Project Pitch
11/09/2017

Overview

We want to implement a cutting edge real-time cloud rendering system that was recently developed by Guerrilla Games for 'Horizon Zero Dawn'. Our implementation will be rendered using a combination of a compute and graphics pipeline using the Vulkan API and will be a public repository. This presentation has provided an intuitive solution for rendering artistically beautiful and realistic looking clouds, while still maintaining real-time rendering speeds. Furthermore, many other people in the industry are beginning to incorporate this into their rendering engines, both at game studios and animation studios.

Clouds are ray marched in a density field that is read in from textures. First, we ray march low frequency noise to create a base cloud shape; then having a decent approximation of where the clouds are we step back along our ray marcher and use high frequency noise textures to carve in detail into the base shape of the cloud and use curl noise for wisps along the edges of the clouds. Lighting will be based on multiple scattering, beer's law and phase functions. Culling, cone marching, rendering every 4th pixel per frame are some optimizations that will be added. Antialiasing via fast approximate and temporal methods applied as a separate pass is a stretch goal we want to achieve.

Goals

- Implement a real-time volumetric cloud renderer in Vulkan
    - Render clouds by ray marching noise fields defined by textures
    - Weather map to incorporate cloud type and coverage
    - Lighting
- Stretch goals:
    - Load in a terrain model through the graphics pipeline and incorporate lighting to achieve Crepuscular ("god") rays
    - Implement anti-aliasing (fast approximate and temporal AA)

Milestones

Milestone 1 (11/20): Develop a Vulkan rendering base from scratch

- Compute pipeline and graphics pipeline:
    - Load and read textures
    - Rasterize complex model that's animated per frame due to compute shader calculations

Milestone 2 (11/27): Develop a raymarching engine in the compute shader

- Render clouds with low frequency textures based on Perlin and Worley noises
- Implement optimizations and high frequency detail to clouds

Milestone 3 (12/04): Incorporate art-directable functionality

- Modify textures during runtime by painting different channels for any texture being used

Milestone 4 (12/11): Incorporate lighting & scene set up

- Achieve multiple-scattering using a combination of Beer's Law and Henyey-Greenstein functions
- Utilize a combination of Beer's law and other equations to create the "powder" effect as described for the cloud edges in the presentation

References

1. Horizon Zero Dawn Clouds (2017): https://www.guerrilla-games.com/read/nubis-authoring-real-time-volumetric-cloudscapes-with-the-decima-engine
2. Horizon Zero Dawn Clouds (2015): https://www.guerrilla-games.com/read/the-real-time-volumetric-cloudscapes-of-horizon-zero-dawn
3. Decima Engine AA: https://www.guerrilla-games.com/read/decima-engine-advances-in-lighting-and-aa
4. Real-time volumetric rendering course notes: http://patapom.com/topics/Revision2013/Revision%202013%20-%20Real-time%20Volumetric%20Rendering%20Course%20Notes.pdf
5. Autodesk Stingray Volumetric Clouds: http://bitsquid.blogspot.com/2016/07/volumetric-clouds.html
6. Volumetric Clouds Theory: http://www.futuredatalab.com/volumetricloud/
7. Gamedev Net discussion on Horizon paper https://www.gamedev.net/forums/topic/680832-horizonzero-dawn-cloud-system/
8. Creating volumetric ray marcher http://shaderbits.com/blog/creating-volumetric-ray-marcher

Images