CS 488 - Spring 2025

Final Project Proposal

Benjamin Colussi - 2046973

Introduction and purpose:

Welcome to Mirrodin, a planet made entirely of metal. Five moons orbit the planet, each emitting a different wavelength of light corresponding to the five colours of mana, red, green, white, blue, and black. Glimmervoid is Mirrodin's equivalent of a desert. It is a landscape of rolling hills made of hexagonal metal plates stretching infinitely into the horizon.

This project is inspired by the original artwork of Lars Grant-West depicting Glimmervoid for the trading card game, Magic: The Gathering. The scene makes for an interesting simulation of light transport due to the combination of reflective surfaces, multiple distant light sources of various wavelengths, and atmospheric scattering effects. The simulation of light in this scene will be difficult to implement because of the purely reflective objects that make up the landscape. As rays are traced and reflected, the colour that is shown in each pixel will ultimately correspond to the colours of the atmosphere and light sources themselves. The light sources are also in the distance, and will take some creative modelling and simulation to get the desired effect of being giant, distant objects in the sky.

Reference art depicting Glimmervoid, as well as the coloured moons and hazy atmosphere of Mirrodin:

Technical outline:

In order to simulate the scene in a physically-realistic (as physically realistic as an alien atmosphere can be simulated) manner, unidirectional path tracing will be used to model light transport. Path tracing is good at modelling the reflection of light rays, and the scene will produce many reflections due to the smooth metallic materials. The existing ray tracer can be extended to a path tracer in a logical manner. A BVH has already been defined and I have implemented SAH-based sorting to improve performance.

Ray marching will be used to simulate the volumetric scattering of the participating media in the atmosphere. Since the entire scene will be from within the atmosphere of the planet, ray marching will be used for all rays. Physical models for light scattering will be used, such as Beer-Lambert, Rayleigh, and Mie. Constant values similar to those corresponding to Earth, its moon, and its atmosphere will be used as a starting point, from which artistic experimentation may expand in order to get the desired atmospheric effect of the alien planet. Since moons are much closer than suns, we can use metres as a unit without worrying too much about floating point inaccuracies. Further optimizations may be pursued to speed up the rendering process, as path tracing and ray marching together will be computationally expensive.

We will have to define new structures to model the atmosphere, as well as the distant light sources. Using spheres will make implementation simpler, because the scene will be static and the atmosphere and sky will be fixed, so we will not require actual object files. We will be able to load in a landscape object to the renderer, similar to functionality of the base code. This will allow us to experiment with different objects and scenes, as well as test objects more easily. I found a simple way to model a hexagonal landscape in Blender, so I will be able to build the landscape object file myself.

Revised objectives:

1. Refactor existing renderer:
   i. Create an offline rendering process that writes to a .ppm lossless image file.
   ii. Implement parallelization using OpenMP.
   iii. Load object files from the command line.

2. Unidirectional (backward) Monte Carlo path tracing:
   i. Monte Carlo integration: The Monte Carlo estimator is used to approximate the integral in the light transport (rendering) equation.
   ii. Subpixel sampling using stratified sampling ("jittered" samples): Pixels are divided into equally-spaced strata and a randomly-directed ray is traced through each to reduce variance and combat aliasing.
   iii. Russian roulette: Unbiased stopping condition used to improve efficiency without increasing variance.
   iv. Implement the inversion method of sampling from PDFs.
   v. Multiple importance sampling: Optimal weighting of importance sampling when sampling from multiple functions such as reflection functions, scattering functions, incident radiance from light sources, etc.

3. Light source models:
   i. Implement area light sources.
   ii. Implement coloured light sources.
   iii. Implement spherical objects, ray-sphere intersections, and an efficient sampling method for generating a point on the visible surface of the sphere. Might be better to implement as circles instead of spheres.

4. Reflection and transmission models:
   i. Implement improved BSDF models using microfacet theory, as described in PBR 9.6, ie. microfacet distribution, masking/shadowing function, only sampling from visible surface, etc.
   ii. Implement Fresnel equations.

5. Volumetric scattering:
   i. Implement ray marching.

6. Atmospheric scattering:
   i. Use ray marching to accurately simulate the scattering of light in an atmosphere using physical equations such as Beer-Lambert, Rayleigh, Mie, etc.

7. Modelling the scene:
   i. Use Blender to create a simple, metal, hexagonal landscape with a central, glass orb.
   ii. Use Blender to create simple test objects.
   iii. Position static light sources and atmosphere in the scene.

Note:

I received bonus credit for the following assignment tasks:

- A1: SAH-BVH
- A2: Improved rasterization

Bibliography:

Used as reference for implementing path tracing, volumetric scattering, and pretty much everything else too:

- Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2016. Physically Based Rendering: From Theory to Implementation (4th. ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Atmospheric scattering:

- Display of The Earth Taking into Account Atmospheric Scattering, Nishita et al., Siggraph 1993.
- Display Method of the Sky Color Taking into Account Multiple Scattering, Nishita et al., Siggraph 1996.
- Preetham, A. J., P. S. Shirley, and B. E. Smits. 1999. A practical analytic model for daylight. In Proceedings of SIGGRAPH '99, Computer Graphics Proceedings, Annual Conference Series, 91–100.

Used as supplementary reference. Good article on atmospheric scattering implemented with ray marching.

- Scratchapixel 4.0, founded by Jean-Colas Prunier around 2009.

Photon mapping:

- Jensen, H. (1996). Global Illumination using Photon Maps.