**Capstone Proposal**

Name: Andrew Jensen

Capstone Title: Raytracing Renderer with a scene editor

**What?** *brief summary of what you want to do for capstone*

|  |
| --- |
| I want to make an editor that allows the user to put together a 3d scene and then render it using raytracing. This project would include two parts: first the scene editor, which lets users quickly put together a scene and be able to see roughly what it would look like. This editor would basically only render models and let users set their orientation, rotation, and material, as well as define a camera. If there is time for it, users could also define textures to apply to the materials. Second, a raytracing engine which would read the scene created by the user and then render it at their chosen resolution, sample count, etc. This would likely only render basic shapes (spheres) and models in OBJ format. |

**Why?** *why do you want to do this capstone project*

|  |
| --- |
| I want to learn more about 3D rendering technologies, both raytracing and rasterization, and I also want experience building tools that might be used in the game industry. I also want to learn about making technologies using different languages communicate. |

**How?** *what are the technologies that you are going to use for this capstone project*

|  |
| --- |
| The editor will be written in C# using UWP or WPF, and use a simple rendering engine using C++ and OpenGL to let users interact with the scene.  The raytracer will be written in C++ using CUDA to accelerate the rendering on the GPU.  JSON will be used to create the scene file in the editor and pass it to the raytracer. |

Raytracing Renderer with a Scene Editor

For my capstone I am making an editor that allows the user to put together a 3d scene and then render it using raytracing. This project would include two parts: first the scene editor, which lets users quickly put together a scene and be able to see roughly what it would look like. This editor would basically only render models and let users set their orientation, rotation, and material, as well as define a camera. If there is time for it, users could also define textures to apply to the materials. Second, a raytracing engine which would read the scene created by the user and then render it at their chosen resolution, sample count, etc. This would likely only render basic shapes (spheres) and 3d models.

Technologies used: The editor will be written in C# using UWP or WPF, and use a simple rendering engine using C++ and OpenGL to let users interact with the scene. These two programs will most likely communicate through C#’s interop services, with the C++ code being compiled to a .dll first.

The raytracer will be written in C++ using NVIDIA’s CUDA technology to accelerate the rendering on the GPU.

JSON will be used to create the scene file in the editor and pass it to the raytracer.

Various libraries will be in use, including SDL2, OpenGL using GLAD, GLM, rapidjson and probably others as needs arise.

From this project, I want to learn more about 3D rendering technologies, both raytracing and rasterization, and I also want experience building tools like those that are used in the game industry. I also want to learn about making technologies using different languages communicate (interoperability).