CIS 565 – Final Project Proposal Sean Thomas

DEFERRED SHADER WITH SCREEN SPACE CLASSIFICATION

For my final project I propose to implement a deferred shader renderer that incorporates several post-processing effects and three different types of light sources. Deferred shading has seen recent use in several game engines. Two main advantages of deferred shading are that "lighting cost is independent of scene complexity", it generates information that is useful to certain post-processing shaders. (Rusty Koonce, "Ch. 19: Deferred Shading in Tabula Rasa", GPU Gems 3). For each effect or light-source that does not affect the entire screen, a screen space classification will be used to determine whether a group of fragments (a "tile") is impacted by the effect or light-source. For post-processing, a shader will be run on each tile that only includes the effects that have an impact on fragments in the tile. For lighting, the lighting shader will only perform lighting calculations for the lights that affect a given tile. The post-processing categorization is inspired by Hutchinson et al., "Screen Space Classification for Efficient Deferred Shading", and the lighting classification is inspired by Johan Andersson, "Parallel Graphics in Frostbite – Current and Future".

The renderer will support point lights, directional lights, and spot lights. Point lights are light sources that illuminate in every direction from a point, directional lights are point lights that are infinite distance from the scene, and spot lights are light sources that illuminate in a cone and may have a minimum and maximum illumination distance. Generally, directional lights and point lights could affect every fragment, whereas spot lights will have a limited effect in screen-space.

The renderer will also support the following post-processing effects: motion blur, depth-of-field, SSAO, and edge blurring (to simulate AA). Motion blur will be classified on whether the geometry is flagged as static or dynamic; however, if the camera is moving, motion blur will need to be applied to the whole scene. Depth-of-field will be classified on the depth of a sample from within the tile. SSAO and edge blurring will both be classified using an edge-detection filter. To increase performance, the sampling for classification will hopefully occur on a per-tile instead of per-fragment basis.

The performance of classification will be compared to two other cases. In the first case, the full shader will be run on every fragment for every light. In the second case, there will still be no screen-space classification; however, each shader will contain an early branch condition (hopefully identical to the classification condition) that determines whether the rest of the shader is run.