Overview

We want to implement a cutting edge real-time renderer for games. To do this we have decided to implement a Forward+ renderer. A Forward+ renderer builds on the concepts of deferred lighting to bring that technique to the next level. Deferred lighting was designed to take advantage of the GPUs found on last generation consoles. Modern day GPUs have a much higher ALU/memory ratio and much smaller branch penalty, and the Forward+ technique takes advantage of these differences.

The implementation relies on compute shaders, so we will be implementing the project using OpenGL. We could attempt to use WebGL and do the compute on the CPU, but we believe it is important to implement it like a real game studio implementation would.

Once we complete the Forward+ renderer we also have some stretch goals that we really want to achieve. There is a Crytek presentation describing their engine for *Rise* and why they almost decided to use Forward+ rendering before going with the standard deferred renderer. They say that while the positives are encouraging, they were concerned that most research on the topic focused on point lights and not on other common light types like area and spot lights. They go on to say that the technique would be a good one to pursue in the future. We'd like to take the first crack at these different light sources, and if we have more time implement additional features like MSAA and SSAO.

Goals

- Implement the tile-based Forward+ renderer
    - Light culling compute shader.
    - Soft shadows through shadow map.
    - Textures / arbitrary meshes.
    - Point lights.
- Stretch Goals:
    - Area lights.
    - Spot lights.
    - MSAA
    - SSAO

References

1. https://takahiroharada.files.wordpress.com/2015/04/forward_plus.pdf
2. http://www.slideshare.net/takahiroharada/forward-34779335
3. http://www.crytek.com/download/2014_03_25_CRYENGINE_GDC_Schultz.pdf