

1 Progress

What I did this week:

- Extended the code to support point lights.
- Optimized all the shaders to avoid as much as conditional branching as possible, gained around 3-4 ms
- Added support for skyboxes and added the reflected fresnel term.
- Tried to change the main arraytexture to use image load/store of OpenGL 4.2, no performance improvement noticed
- Tried to change the mipmap generation to use a compute shader of OpenGL 4.3, but it gives a worse performance compared to the framebuffer based solution (8 ms in the best case with kernel (8,8,1) vs 5-6 ms).
- Tried different random number generators on gpu (including texture with random numbers loaded from CPU), testing for performance and quality
- Studied animated version of the model (rotating light around it), reasoned on how to avoid some flickering problems. Found out that it depends a lot on which things you randomize in the shader. If the condition of light changes, to avoid flickering is often better to avoid excessive randomization. Also, to reduce noise, it is useful to accumulate the results over various frames (= less sensible to light changes, but more stable).
- Added advanced timer for precise performance timing (it uses glFinish, so it must be taken with a grain of salt)
- Added simple pre-processing unit for shaders in order to recycle some pieces of code.

Stuff left to do (implementation wise):

- Environment lights using 16 different directions
- Some small adjustments in the engine code
- Add support for distant area lights