Jacob Daniels-Flechtner

CMPM 163

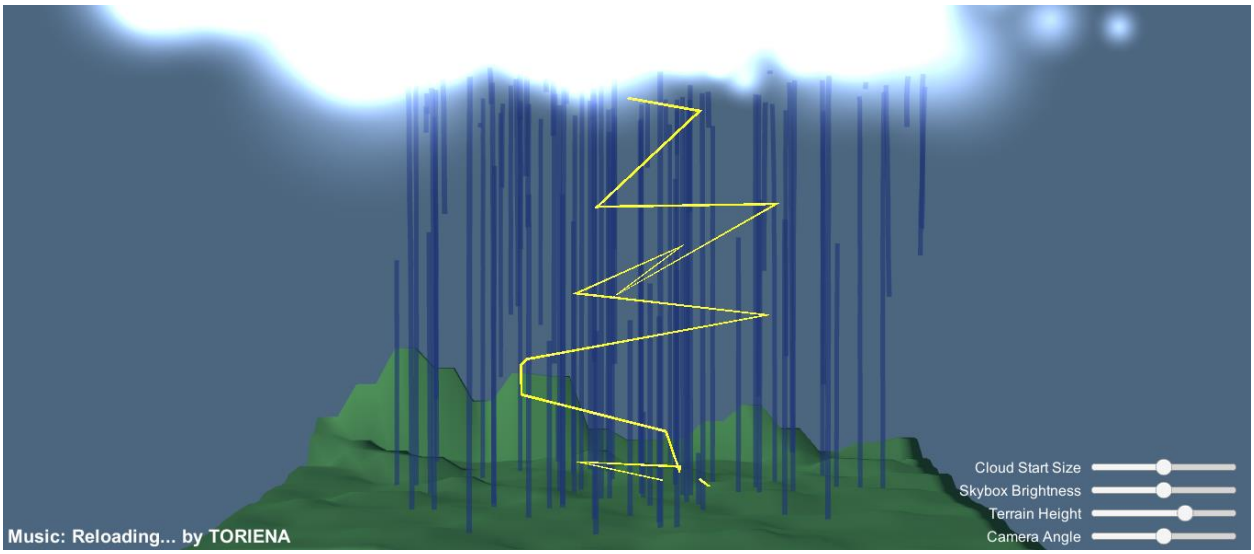
May 28, 2019

Homework 3 ReadMe

Github link: https://github.com/JacobDanielsF/HW3

Part A

This scene includes three particle systems that react to an audio source. Lightning appears when upper pitches occur, rain is emitted and expands when middle pitches occur, and the clouds have noise-based size and movement that fluctuates rapidly when low pitches occur. The ground terrain also instantly reacts to the music. Multiple elements of the scene can be edited with the UI sliders on the bottom-right.



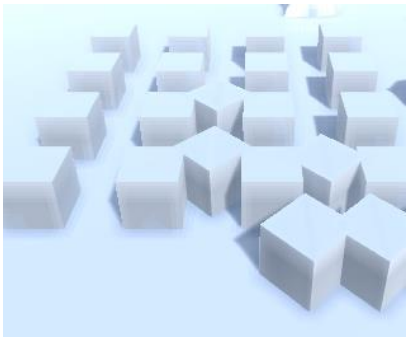
Part B

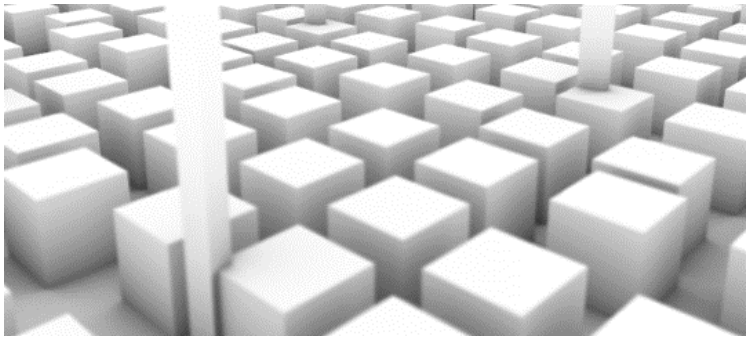
Final project concept: Ambient occlusion

I was considering doing real-time indirect lighting for the final project, though I want the speed of computations to be quick and lag-free. I decided to turn to ambient occlusion, though I may choose to add other features to make shading smoother and have light blend smoothly into existing shadows.

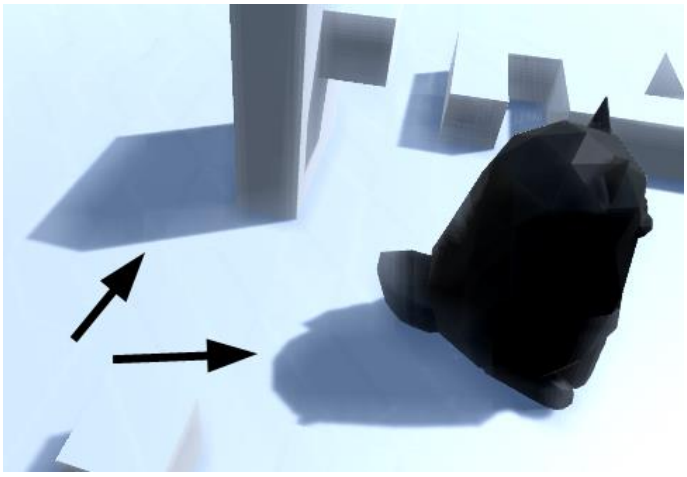
I want to pass the camera viewport to a shader and retrieve the depth of each texel in order to determine which areas of the screen will be the most difficult for ambient light to hit, then making those texels darker. I can also make a rough approximation of each surface normal

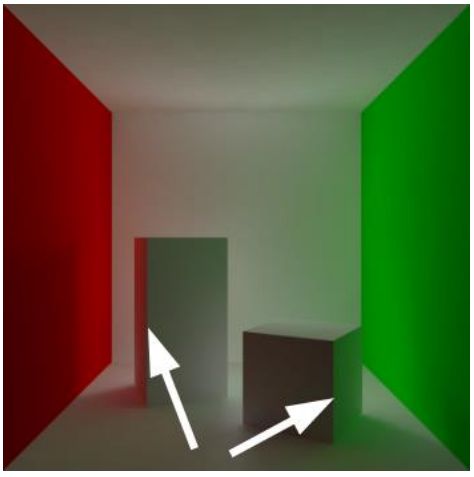
by checking for a continuous difference in depth values across a series of texels (a flat surface will have distances from the camera that are evenly spaced.) I could also try checking for the exact normal that occurs at a specific point. A change in surface normals between texels can help determine if a texel is located next to or between other surfaces that could easily block ambient light. Texels can also be sampled from larger distances to detect changes in distance across the screen. A difference between texel distances can create a shadow, but only when the distances are within a reasonable range. This prevents background objects from receiving excessive shadows from foreground objects. I also want to make the ambient occlusion dither out for normals facing upwards.





I also want to include a feature where sufficiently dark areas of the screen are blurred into the surrounding area, making it seem as though light is bouncing between shadows. A wider portion of the screen will likely need to be sampled in order to cover larger shadows. It may also be necessary to differentiate between locations where actual shadows land and areas that have multicolor textures. I would prefer if most of the computations were done on the GPU and raytracing was avoided if possible.





Team members for final project: Taylor Infuso, Oskar Alfaro, Jason Chen, Junhao Su