[redacted]

Effects: SSR, Shadow Maps with PCF and Tessellation

Shadow mapping with PCF using a poison disc, we have 3 point lights in our scene, each has a cube shadow texture, the lights have a variable radius (for attenuation calculation), and a variable “shadow softness”, as well es color.
For softer lights a bigger poisson disc is used.

SSR is done by ray marching in worldspace (still just using screen space depths though of course), every step is a bit farther than the last, this leads to surfaces which can reflect faraway geometry without costing to much performance. This works great, specially in combination with roughness as it is practically impossible to notice the dropping reflection quality in rough reflections. After a hit is detected a binary search is used to refine the result.
I used https://github.com/ImanolFotia/Epsilon-Engine quite a lot to check on procedures, the final code deverges pretty far from anything that can be found there though

The first scene render pass writes to multiple textures (color, world position, metalness & roughness, world normals), These textures are then used to render the scene with SSR on a fullscreen quad. Surfaces with a higher metalness value are more reflective, roughness is simulated by sending the reflection rays in a slightly random direction (the rougher the surface the stronger the deviation from the mirror reflection ray) the randomness is simulated using a hash function which uses the world position as seed.
This leads to visible artefacts which are quite artistically pleasing though so we decided to roll with them.

Models are mainly done procedurally, simple shapes were enough for the most part. We can import models and whole scenes though and draw them with multi draw indirect, we did not really use it though, it did not seem to have much of a positive performance impact on our scene, since we use few elements and some require different shaders. Also animation import from FBX was planned but not realized, which kept us mostly with procedural animations.

The few models we did use, and discard again, were made in blender.
To animate some of the geometry, the ground plane and a sphere in the middle above the monolith, we used a self-written python script precomputed textures containing the lower half of a fft calculated over the background music. These 10 textures are collected into an array texture which can be used in the shaders.

Controls:
3 - debug cam
4 - actual cam

Libraries:

- assimp
- windows API to play sound
- glfw
- glew
- glm