IM420: Advanced Computer Graphics (IM VZ WS17) Project DOCUMENTATION

Artelsmair Markus, Kriz Thomas

Volumetric:

The idea was to implement ray marching and a custom noise in the fragment shader to render a magical cloud-like volume. 3D noise is generated and sampled via the helper methods *random, noise3d* and finally *fbm* which stands for fractal brownian motion. There a „virtual cube“ is created to combine noise samples from every cornerpoint using the 3d random function. Finally the fbm can be sampled by using *fbm (in vec3 _st)* where several noises are layered to create a complex fractal noise, depending on the octaves shader-setting and other things. The same technique is used in the water shader.

For the volume to render correctly a back- and frontface-worldposition texture needs to be rendered first and then sent to the volume shader. This is done using another program called „frontBack“ in a separate rendering pass. This happens in a custom render function „volumeRenderer(…)“.

The raymarching is then performed using the calculated distance between front and back texel. Various settings such as the stepsize are defined by shader settings. The animation takes place by sending the global time to the shader and offsetting/modulating a few values – have a look on your own.

Water:

For the water the same noise technique was used but in a 2D version.

Additionally I tried to create and render a reflection with a 2 pass version, where a reflection texture is rendered using a mirrored camera and then sent to the shader. In the shader it is sampled using the texCoords, the distance from eye to fragment/worldpos and a calculated normal from the noise to display it correctly. This unfortunately does not work when using the volume rendering at the same time (see Problems) and can look a bit unrealistic.

Problems:

In the volumeshader I was a bit too ambious by trying to implement volume shadows. This would be possible by an additional raymarching loop that runs thorugh a few voxels towards the light position or along the light vector from the current voxel and sums up the density to calculate the shadowing. Unfortunately this turned out to be more complicated than expected and would have probably needed another prepass for the rendering – it is also quite expensive on the rendertime I guess.

This was my approach that i basically came up on my own – there is probably a better solution.

As mentioned the water reflection pass does not work/render when having the volume model in the scene at the same time – I could not figure out why. Its also not possible to render 2 different volumes at the same time for some reason (?).