Documentation for Submission 2
Project “GodMode”

(course 186.140 Echtzeitgraphik)

Group 02		
Strohmayer Julian	01426125	e1426125@student.tuwien.ac.at
Gantner Patrick	01576033	e1576033@student.tuwien.ac.at

Bloom implementation details:

For the Bloom effect the full scene is rendered to an off screen framebuffer object with a colour attachment and a depth attachment. This texture is then filtered using a luma conversion [1] to get the bright areas of the scene. After that the highlight texture is blurred by a horizontal 1D Gaussian kernel with size 11 and a vertical 1D Gaussian kernel with size 11. In the last step the blurred highlight texture is combined with the original scene texture by adding the color values of both textures together. (currently all textures are rendered in full screen resolution since the framerate is well over 60fps (200-500+ depending on the terrain size), but may be scaled down at a later point to improve performance)

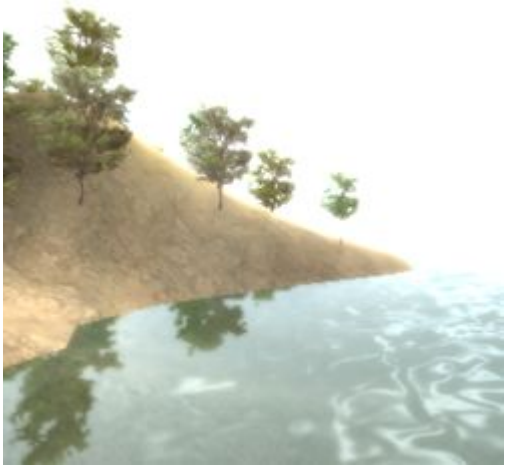

Figure 1 - implemented Bloom effect (with unusual high bloom factor)

Volumetric Light Scattering implementation details:

For the God Rays effect a sun billboard texture is created which is always facing the camera. This is done by replacing the part of the ModelView-Matrix which is responsible for scaling and rotation (3x3 upper left) with the an identity matrix. To create the light source masking texture the sun is rendered with black background color and every object in the scene is rendered completely black on top. This texture is then blurred using a radial blur where the blur origin is the screen space position of the sun. For the radial blur the approach described in [2] was used. The radial blurred light source masking texture is then combined with the output texture of the bloom effect.

Figure 2 - implemented volumetric light scattering effect

Lens Flares implementation details:

The basic idea used for this effect was found in [3][4]. The first step in creating the Lens Flare effect is to downsample the current rendered scene and apply a threshold to it to get the input texture for the lens flare shader.
In the lens flare shader an alignment vector for the “ghosts” is created from the center of the screen to the inverted texture coordinates. With a loop the ghosts are placed along this vector by calculating an offset and a weight value depending on how far from the center the ghost is. The ghosts themselves are distorted texture lookups in the downsampled input texture from the previous stage. After this another ghost is added in form of a halo by using a fixed length offset from the center.
Finally the result of the lens flare shader is added to the original rendered scene.

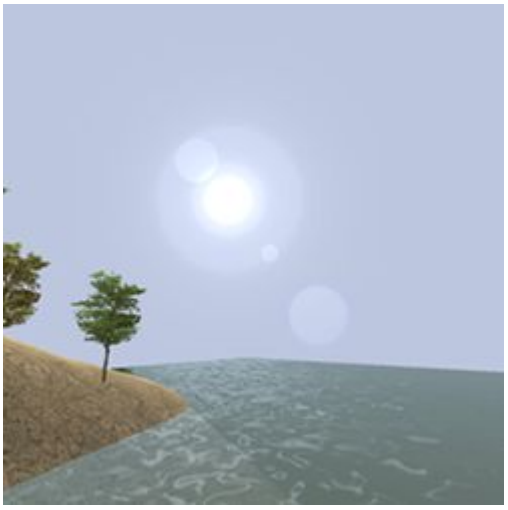

Figure 3 - implemented Lens Flare effect

Other Features:

The framework of this project is a terrain simulator implemented by Julian Strohmayer, which is a reduced and optimized version of the UE Computergraphik project “evilution” back in 2015. The game functionality was completely removed and new features were implemented such as parametric terrain (using perlin noise) of arbitrary size or instance rendered vegetation which is automatically placed onto the terrain by a space filling algorithm. The framework also has realistic looking water with moving waves (using normal mapping), reflection, refraction, Fresnel Shading and a murkiness effect.

Controls:

camera movement	WASD-keys + mouse
enable/disable fps counter	F1 key
enable/disable wireframe mode (not working at the moment because the post processing effects are rendered as full screen quad over the scene)	F2 key
enable/disable fog effect	F3 key
start demo	**F4 key**
end demo (switch to debug camera)	**F5 key**
adjust sea level	Up/Down arrow keys
close program	ESC key

The program was tested on the following GPU's:
Lab PC "AMAROK" card on the bottom
NVidia GTX 960
NVidia 940MX
AMD Radeon HD 7850

References:
[1] https://en.wikipedia.org/wiki/Luma_(video), Accessed: 26.11.2017 10:20
[2] https://wiki.delphigl.com/index.php/shader_radial_blur, Accessed: 26.11.2017 11:36
[3]https://informatik-forum.at/showthread.php?112939-Lens-Flares-Effekte-Allgemein&p=841342&viewfull=1#post841342, Accessed: 26.11.2017 13:09
[4] http://john-chapman-graphics.blogspot.co.at/2013/02/pseudo-lens-flare.html, Accessed: 26.11.2017 13:04