

186.140 Real-time Rendering

Project Proposal

Christoph Körner, 0726266, 932, office@chaosmail.at
Marc Haubenstock 1525175, 932, marc.dhaubenstock@gmail.com

October 15, 2015

1 Treatment

The camera slowly enters the scene through a layer of clouds, seeing a wide terrain with water. After hovering around mountains, the camera makes its way towards the environment's surface. Slowly passing through the vegetation we make our way towards the water.

We inspect the water's surface trying to discern what lies beneath it. Diving into it, we display the local plant life. Diving deeper we see a silhouette of a distance object. Finally we are able to identify our target: A shipwreck lying on the bottom of the ocean.

2 Effects

2.1 Lighting and Shading

All elements in the scene will use the Phong Reflection Model and Phong Shading [8].

2.2 Infinite Procedural Terrain

To create a realistic procedural terrain, we will combine different noise functions [1], such as Perlin noise, fractal Brownian motion, multi-fractals and hybrid models [3].

We will render the noise function as a grayscale image into a buffer and use it to apply it to the ground surface as a displacement map [2]. Then, we will apply texture mapping [7] to different heights of the terrain [1].

The terrain should be rendered with higher details for regions closer to the camera, and less details for regions that are further away. We will use adaptive Tessellation [12] to implement this effect directly on the GPU.

If there is still time left, we would like to implement erosion [4] in order to imitate the effect of realistic physical processes (such as wind, weather or water) and to create a more plausible terrain.

2.3 Shadows

All elements in the scene should use soft shadows using depth maps [6] and percentage closer filtering [5].

2.4 Water

The water will be created from a simple surface geometry. It should show reflection and refraction effects depending on the camera angle [10]. In addition, we want to use bump mapping for plausible lighting [1] effects.

If there is still time left, we would like to include a wave model based on the frequency spectrum of real oceans [9].

2.5 Sky and Clouds

The Sky will be model as a Skybox [11] with additional billboard-style clouds [2].

2.6 Trees and Vegetation

We will cover the terrain with vegetation whose geometry is created with 3-dimensional L-Systems [1]. In addition, we will use texture mapping and displacement mapping to create realistic vegetation [7].

3 Technologies

For the implementation of our project, we will use *C++*, *OpenGL 4*, and additional helper tools such as *Eigen* for linear algebra, *OpenGP* for geometry processing, *GLFW* for windows and inputs and *GLEW* for extensions. In addition, we will use a small self-written object and geometry wrapper.

References

- [1] Ebert, David S. *Texturing & modeling: a procedural approach*. Morgan Kaufmann, 2003.
- [2] Akenine-Möller, Tomas, Eric Haines, and Naty Hoffman. *Real-time rendering*. CRC Press, 2008.
- [3] Musgrave, F. Kenton. *Procedural fractal terrains*. Texturing and Modelling. A Procedural approach, 1994.
- [4] Olsen, Jacob. *Realtime Procedural Terrain Generation* 2004.
- [5] Reeves, William T., David H. Salesin, and Robert L. Cook. *Rendering antialiased shadows with depth maps*. ACM Siggraph Computer Graphics. Vol. 21. No. 4. ACM, 1987.
- [6] Segal, Mark, et al. *Fast shadows and lighting effects using texture mapping*. ACM Siggraph Computer Graphics. Vol. 26. No. 2. ACM, 1992.
- [7] Shirley, Peter, Michael Ashikhmin, and Steve Marschner. *Fundamentals of computer graphics*. CRC Press, 2009.
- [8] Phong, Bui Tuong. *Illumination for Computer-generated Images*. AAI7402100, The University of Utah, 1973
- [9] Tessendorf, Jerry. *Simulating ocean water*. Simulating Nature: Realistic and Interactive Techniques. Siggraph 1.2, 2001
- [10] <http://blog.bonzaisoftware.com/tnp/gl-water-tutorial/>
- [11] <http://learnopengl.com/#!Advanced-OpenGL/Cubemaps>
- [12] <http://codeflow.org/entries/2010/nov/07/opengl-4-tessellation/>