Practical Course Visual Computing

WS 2013/14

Exercise 5: "Atmospheric Scattering"

Michael Becher

Topic

Realtime rendering of planetary atmospheres.

Task Assignment

The goal of this assignment is to implement terrain rendering, focusing primarily on rendering planetary atmospheres to create a believable sky without a skymap. Although the implementation should be capable of rendering a planetary atmosphere from space, terrain rendering will be limited to a simple height-map based approach, since the focus should remain on rendering the atmosphere.

Introduction to space-lion

In order to concentrate on the core features and challenges of atmospheric scattering, a basic OpenGL framework/engine is used as a starting point. To that end, a new branch of the space-lion framework was created for this project and is publicly viewable at `https://github.com/invor/space-lion/tree/fapra`.

Notable features of space-lion currently are: Context and window creation. Wrapping/implementation of OpenGL core concepts e.g. GLSL shaders, vertex buffer objects, texture objects and framebuffer objects. Basic resource management including loading and management of shaders, textures, meshes and custom materials. Basic scene handling. Basic scene rendering using physically based shading. And the option of adding post processing effects to the rendering pipeline.

As these basic features are mostly unrelated to the task given by this assignment, atmospheric scattering will be a noticeable and useful addition.

Stability and reliability of space-lion has been previously field-tested by me in the sea-crossing[1] project.

Goals

- Pull the latest commit of the fapra branch and see to it, that everything is working as it should.
- Basic terrain rendering.
- Texturing the terrain.
- Pre-computation of the scattering integrals and storage of the result in 2D/3D textures.
- Actual sky-rendering.
- Day/Night cycles are possible by changing the sun position.
- Bonus! The sky is looking good, but the terrain is still a bit blank. Make use of space-lion's fbx capabilites and load a few meshes to place on the terrain.
- ...

Input/Output/Interaction

Description ...

Modules

Module A: Fapra

Check `src/fapra` and `resources/shaders/fapra` for the source code. Terrain surface will probably use the space-lion default surface shader.

[1] `https://github.com/chaot4/sea-crossing`

Module B: Space-Lion Core

Already existing module. Check `src/engine/core` for the source code. Modifications to the sources in this module are expected depending on problems and requirement that will arise in the course of this assignment.

Module C: Space-Lion Fbx

Already existing module. Check `src/engine/fbx` for the source code. Changes or frequent use of this module are not expected in the scope of this assignment.

Technical Overview

Description

Approach

Description.

Criteria of Grading

The overall 20 points given for this assignment are distributed to the following criteria:

Compulsory Criteria

1 Point An original concept is handed in. The originality will be illustrated by two high resolution screenshots of the final product.

1 Point The code is well structured and documented.

1 Point The `proposal.tex` meets all requirements.

1 Point The `README`-file contains a Section *MANUAL* that describes in detail the usage of the program.

1 Point The `README`-file contains a section *IMPLEMENTATION* that describes the fundamental parts of the implementation.

Self-defined Criteria

x Points

x Points

x Points $\dots$