COMPUTER GRAPHICS 2017

Assignment 1
3D Rendering and Transformation

Patrick Gabrielsson

Contents

1 Purpose

This assignment covers 3D objects, transformations, views and projections in MonoGame. It also includes height maps, basic lighting, texturing, skyboxes and handling user inputs.

2 Description

In this assignment you will be building a terrain from a height map using vertex buffers, index buffers and a texture containing the height data. You will also be implementing transformations for a given model of a chopper, containing multiple meshes, as well as a camera to view the result (see Figure 2.1).

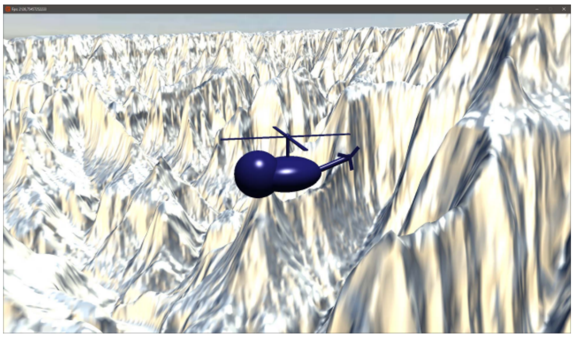

Figure 2.1: Chopper flying over terrain.

The model contains three meshes. One mesh for the main body of a chopper, one mesh for the main rotor blade and one mesh for the rear rotor blade. The two rotor meshes should rotate as expected on areal chopper. The complete chopper should also be able to rotate and move around freely without having issues with Gimbal Lock or other erroneous rotations. It should at least be possible to control the chopper with the keyboard.

The height map must be rendered using vertex buffers and index buffers that are precomputed and uploaded to the GPU. You may use either triangle lists or triangle strips (a bit harder but uses less GPU memory) to perform the draw calls. You may choose to use the supplied height map texture or make your own height map texture (using, for example, a site like terrain.party).

There are no specific shading requirements for this assignment. You may use the *BasicEffect* in MonoGame to get decent shading without much effort or just use colored vertices.

As two optional exercises, texturing can be applied to the terrain, and to a skybox (these two exercises are not mandatory).

3 Entity Component System

The implementation of the height map, model transformations and the camera, should follow the Entity Component System architecture, in order to easily be inserted into your game engine in the Game Programming course. Below are some suggestions with regards to the components and systems that you need to implement in this assignment. These are just suggestions, i.e. you are free to come up with your own components and systems.

If you decide to texture the terrain and to create a textured skybox, you will have to create additional Components and Systems (these two exercises are optional).

3.1 Components

- CameraComponent
  - Holds camera specific data such as the FOV, aspect ratio and near/far plane
- TransformComponent
  - Holds data such as position, rotation and scaling.
- HeightmapComponent
  - Holds all the data related to the height map (e.g. height data, vertex/index buffers).
- ModelComponent
  - Holds a model and the data transforms for its meshes.

3.2 Systems

- CameraSystem
  - Computes the view and projection matrix for all the CameraComponents.
- TransformSystem
  - Computes the transformation matrices (world-matrices) for all TransformComponents.
- HeightmapSystem
  - Creates and renders all the HeightmapComponents.
- ModelSystem
  - Renders models and applies the correct transforms to the models' submeshes.

4 Miscellaneous

4.1 Files

The model of the chopper and the height-texture can be found on PingPong (under Documents -> Assignments -> Assignment 1).

4.2 Groups

In order to be able to upload files to PingPong, you need to choose a group. Groups can be chosen via the Doodle link on PingPong (under Contents -> Assignment Groups). Work in groups of 2-3 people.

4.3 Tutoring

Two tutoring sessions will be given according to the schedule in Kronox. Tutoring can be booked via the Doodle links in PingPong (under Contents -> Tutoring -> Assignment 1).

4.4 Submission

The code, along with an executable, should be handed in on PingPong (under Contents -> Laboration 1) before each scheduled deadline according to Kronox.

4.5 Useful Links

XNA terrain tutorial
XNA flightsim tutorial
XNA code and tutorials