Prof. Dr. Leif Kobbelt
Kersten Schuster, Philip Trettner

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Real-time Graphics Assignment 7

Date Published: December 13th 2017, Date Due: December 20th 2017, 11:44

- The assignments have to be done in groups of 2 students.
- Hand in the solutions to the exercises via L^{2}P.
- You are only allowed to change code inside the marked strips (STUDENT CODE BEGIN/END)!
- Any questions? → L^{2}P discussion forum or rtg@cs.rwth-aachen.de!

If not done yet, obtain the (publicly accessible) exercise framework and assignments from `https://www.graphics.rwth-aachen.de:9000/Teaching/rtg-ws17-assignments/`.
Use **git pull** to fetch the newest changes of the framework (including the code for this exercise).

The **only** files that you should modify and **upload**:

- Chunk.hh (basically for delaration of helper functions)
- Chunk.cc
- Vertices.hh
- terrain.vsh

Description In this assignment you will create the mesh data for a blocky voxel terrain.

Controls You can place blocks with left mouse button (green overlay). Keep Ctrl pressed to remove blocks (red overlay). Keeping Shift pressed while clicking (yellow overlay) stores the clicked block's material (pipetting). The currently existing materials are sand, grass, rock, snow, water, and air.
W, A, S and D keys can be used for navigation while the right mouse button allows you to rotate the camera. Double-clicking the center mouse button resets the camera position.

Further Help As always, you find code strips in the framework with more detailed comments and hints. You can find some screenshots in the folder `screenshots`.

Performance Hints If your hardware is somewhat weak, the following might help (mostly in the TweakBar):

- Reduze shadow map size (or disable shadows completely)
- Keep the render distance small
- Reduce your window size

Prof. Dr. Leif Kobbelt
Kersten Schuster, Philip Trettner

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Exercise 1 Basic Mesh Generation [1 + 2 + 2 = 5 Points]

This exercise deals with a basic mesh generation that does not focus on rendering performance or quality.

(a) In the method `queryMeshes()` in `Chunk.cc`, increase performance by only creating meshes that have changed since last creation. You should use the member variable `mIsDirty` and the method `isDirty()`.

(b) Create faces for all blocks that do not consist of air (`Chunk.cc`, `Chunk.hh`).

(c) Adapt your vertex format (in `Vertices.hh`) and the terrain vertex shader (`terrain.vsh`). Derive texture coordinates and tangents in the vertex shader. Do *not* upload them as vertex attributes.

Exercise 2 Advanced Mesh Generation [2 + 2 + 1 = 5 Points]

This exercise improves the methods from Exercise 1 regarding performance and ambient occlusion. Note that if you finished an advanced task, you of course get the points of the implicitly done basic tasks, too (e.g. 1(b) $\leftrightarrow$ 2(a)).
However, it should be easier to start with the basic tasks and do the improvements afterwards.

(a) Do not create unnecessary block faces, i.e. only

- from any block to air
- from an opaque material to a translucent one
- from a translucent material to a different translucent one

(b) Implement and upload some fake ambient occlusion per vertex. Note that the triangulation of a block quad should depend on the ambient occlusion values of the four vertices. (See the reference screenshots, the two whiteboard images, and `https://0fps.net/2013/07/03/ambient-occlusion-for-minecraft-like-worlds/`)

(c) Optimize the vertex format. Upload at most 16 bytes (e.g. a single `vec4`) in which you pack position, normal, and ambient occlusion.
Hint: For axis-aligned cubes, there are only 6 different normals.