Prof. Dr. Leif Kobbelt
Kersten Schuster, Philip Trettner

Visual Computing Institute | RWTH AACHEN UNIVERSITY

Real-time Graphics Assignment 8

Date Published: December 22nd 2017, Date Due: January17th 2018, 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!

Note that this assignment is worth 20 points. You have additional time for solving.

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**:

- Assignment08.cc
- common.glsl
- fullscreen.transparent-resolve.fsh

Description In this assignment you will migrate the framework to a deferred shading pipeline and implement order-independent transparancy.

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, dirt, rock, snow, snowrock, crystal, gold, copper, bronze, 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. Press space bar for jumping and hold shift for faster walking. To toggle between freecam mode and character controller, press key F. Double-clicking the center mouse button resets the camera position in freecam mode.

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`. A summary of the intended pipeline can be found in `rendering-pipeline.jpg`. The TweakBar has a lot of settings to play with. Especially the "Output" setting in the "debug" group might be helpful for debugging.

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

- Reduce shadow map size (or disable shadows completely)
- Keep the render distance small
- Reduce your window size
- Disable point light sources

Exercise 1 Deferred Shading [10 Points]

(a) Clear the G-Buffer properly and perform the opaque rendering pass. Pay attention to the correct `depthFunc` and `depthMask` states and be aware that we need to write to an sRGB buffer.

(b) Implement the depth pre-pass. Do not forget to clear the depth and set the `depthFunc` state.

(c) Implement the light pass

- Do a fullscreen lighting pass (ambient light + directional (sun) light)
- Render the point lights on top

Exercise 2 Transparent Objects [10 Points]

(a) Render the background (skybox) in `fullscreen.transparent-resolve.fsh`.

(b) Render translucent objects with "weighted, blended order-independent transparency" (transparent pass in `Assignment08.cc` and shader functions in `common.glsl` and `fullscreen.transparent-resolve.fsh`)

(c) Implement refraction via a distortion map (shader functions in `common.glsl` and `fullscreen.transparent-resolve.fsh`)