Prof. Dr. Leif Kobbelt
Kersten Schuster, Philip Trettner

# Real-time Graphics Assignment 6

Date Published: December 6th 2017,  Date Due: December 13th 2017, 11:44

- The assignments have to be done in groups of 2 students.
- Hand in the solutions to the exercises via $L^2P$.
- You are only allowed to change code inside the marked strips (STUDENT CODE BEGIN/END)!
- Any questions? → $L^2P$ 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**:

- dunes.vsh
- Assignment06.cc
- fluid.glsl
- (water.fsh)

**Description** In this assignment you will create a simple desert terrain with palm trees and do some water simulation.

**Controls** You can increase water height locally with the left mouse button.
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** Each subtask corresponds to a code strip with more detailed comments and hints. You can find some screenshots in the folder `screenshots`.

## Exercise 1 Terrain Generation [2 Points]

In `dunes.vsh`, implement the method `getTerrainHeight` that computes a terrain height for the given 2D coordinates. You can use the perlin noise implementation `float cnoise(vec2)`, but make sure that the area around the pool has height 0.

## Exercise 2 Palm trees [$0.5 + 0.5 + 1 = 2$ Points]

**(a)** Load the palm tree shader (`mShaderPalms`) from file.

**(b)** Load the palm tree textures from disk.

**(c)** Set up the palm tree shader, set the texture uniforms needed by the fragment shader and the `uPalmPos` uniform needed by the vertex shader. At last, render the palm mesh with back-face culling disabled. The palm trees should be rendered somewhere close to the pool.

## Exercise 3 Water Simulation [$3 + 3 = 6$ Points]

**(a)** In `fluid.glsl`, first strip, implement the `updateHeight` that returns the new water height for the given coordinates. More information can be found in the shader code.

**(b)** In `fluid.glsl`, second strip, implement the `updateFluxX` and `updateFluxY` methods that return the new flux values for the given coordinates. More information can be found in the shader code.

## Exercise 4 (Fix the Refraction [2 Bonus-Points])

**Note: This is a bonus task and not part of the regular exercise!**
The refraction in `water.fsh` show some artifacts when the camera is a little further a way from the pool and looks flat on the pool surface. More information on that can be found in the shader.