Prof. Dr.-Ing. Ralf Steinmetz
Multimedia communications Lab

Dr. Florian Mehm
Dipl. Inf. Robert Konrad

TECHNISCHE UNIVERSITÄT DARMSTADT

„Game Technology“
Winter Semester 2014/2015

Exercise 10

For bonus points upload your solutions until **Friday the 9th of January 2015, 11:40**

General Information

- The exercises may be solved by teams of up to three people.
- The solutions have to be uploaded to the Git repositories assigned to the individual teams.
- **The submission date (for practical and theoretical tasks) is noted on top of each exercise sheet.**
- If you have questions about the exercises write a mail to game-technology@kom.tu-darmstadt.de or use the forum at https://www.fachschaft.informatik.tu-darmstadt.de/forum/viewforum.php?f=557

1. Practical Tasks: Procedural Content Generation (5 Points)

When you run the provided code, you can see a spinning cube which is textured using a procedurally generated texture. Your task is to implement new nodes for the texture generation network and then create your own texture generation network.

You can switch networks by hitting the space bar. The following networks will be cycled through:
0: Basic example: Red grid on white ground
1: Perlin Noise, Frequency 20, Amplitude 1
2: Blur Filter on an image
3: Overlay (Texture with full red over the image)
4: Your own network (see Task 1.4)

1.1 Texture Generator

Implement a Perlin noise generator. You can find the generator's code in PerlinNoiseGenerator.h and PerlinNoiseGenerator.cpp.

1.2 Texture Filter

Implement a Gaussian blur filter as a texture filter. You can find the filter's code in GaussianBlurFilter.h and GaussianBlurFilter.cpp.

1.3 Texture Combiner

Implement a combiner that carries out the "overlay" computation. You can find the combiner's code in OverlayCombiner.h and OverlayCombiner.cpp.

1.4 Create your own texture

Implement a texture of your choice. Describe textually what your intended outcome is and which nodes you have used. You can use any combination of the provided nodes or create new ones.

2. Theoretical Tasks: Procedural Content Generation (5 Points)

1.1 Voronoi Diagram

In a real-time strategy game, players can gain control of cities. The cities are specified by their x and y-positions in a plane. For this game, you want to implement a visualization that shows the area of influence of each player by displaying the border of the region they control. The input to this visualization is the set S of all cities and the subset P of cities that the player controls.

Describe how the Voronoi Diagram can be used for this task.

1.2 Filter kernel

In the lecture, we presented the following kernel as carrying out an edge detection operation:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Explain in your words why this kernel detects edges. Edges are here meant to be those areas in the image where two areas with very different colors connect.

1.3 Gaussian blur

In one dimension, the Gaussian distribution can be calculated by using the formula shown here:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Draw the Gaussian for sigma = 1, 3 and 5, for example using a spreadsheet application or an online calculator such as fooplot.com.

a) Describe how the shape of the Gaussian changes.

b) If we use only the values of G(x) in the kernel (e.g. using size 7) and do not normalize them, our filter will change the brightness of the image. This effect is more noticeable the larger we choose sigma. Explain why this is the case, using your calculations from task a)