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

Dr. Florian Mehm
Dipl. Inf. Robert Konrad

TECHNISCHE UNIVERSITÄT DARMSTADT

„Game Technology“
Winter Semester 2016/2017

Exercise 4

For bonus points upload your solutions until **Saturday, November 19, 9:50**

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: Textures and Depth Buffers (5 Points)**

Extend your software renderer to support texture mapping and depth buffering. For closer instructions see the comments in the `shadePixel` function in the source code.

https://github.com/TUDGameTechnology/Exercise4.git contains code for texture coordinate loading and interpolation. You can either copy the code changes manually or just pull them into your own repository using https://github.com/TUDGameTechnology/Exercise4.git

Please remember to push into a branch called "exercise4".

2. Hypertheoretical Task: Matrix Multiplication Performance (5 Points)

2.1 Weird things (1 Point)

In lecture 4, the three problems listed below were addressed. Choose one of them and describe both the problem and the possible solution in your own words.

Weird depth problems
Weird textures
Weird rotations

2.2 Matrix analysis (2 Points)

a) What geometric operation(s) does this matrix encode? How can you tell?

$$\begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

b) What geometric operation(s) does this matrix encode? How can you tell?

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

2.3 Alphabet war (1 Point)

Your game engine's rasterizer uses a z-buffer. You want to display decals on objects. You implement them by creating quadrilaterals which are exactly aligned with the target surface and which have a distance of 0 to the surface (i.e. like gluing a poster to a wall).

Your game shows weird artifacts where the surface and the decal are both partly visible. What happened and what could be a fix?

2.4 Performance (1 Point)

Which is more performant: Phong or Gouraud-Shading? Explain in your own words why.