



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

Dipl.-Inform. Robert Konrad  
Polona Caserman, M.Sc.



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## „Game Technology“ Winter Semester 2016/2017

### Exercise 3

For bonus points upload your solutions until **Tuesday, November 14, 13:39**

### 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](mailto: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: Triangle Mesh Rendering (5 Points)

Implement a basic software triangle renderer. Specifically, implement keyboard camera controls by carrying out rotation, translation and projection as described in lecture 3. You may use the `Kore::vec3/4` classes to store data, but not the `Kore` matrix or quaternion classes. As an ungraded bonus, you can try out mouse camera control.

You will find comments in the code that direct you to the correct positions to add your code.

The suggested order is to start with translation first, then the perspective transform, and finally rotations. Note the effect each transformation has on the result. In case the model is not visible anymore after a change you made, check your calculations with some example data, e.g. the point  $(0, 0, 1)$ , and find out at which position on the screen it ends up in. Keep in mind what your coordinate system is and where your 3D mesh is located (you won't recognize the mesh if your camera is located inside of it). Be especially careful with  $z$  values as they can easily mess up your mesh – if that happens find the smallest and largest  $z$  value of your mesh and have a close look at what  $z_{min} / z$  (for the perspective division) does.

<https://github.com/TUDGameTechnology/Exercise3.git> contains code for triangle rasterizing and mesh loading and a mesh to get you started. You can either copy the code changes manually or just pull them into your own repository using `git pull` <https://github.com/TUDGameTechnology/Exercise3.git>

Please remember to push into a branch called “exercise3”.

## 2. Theoretical Tasks (5 Points)

### 2.1 Geometric interpretations (2 Points)

a) Write down the definition of the cross product. What for can we use cross product in games? Give a small example.

b) Write down the definition of the dot product. What for can we use dot product in games? Give a small example.

## 2.2 Angle calculations (1 Point)

Rotate a Triangle with points  $p_1 = (1,1)$ ,  $p_2 = (2,3)$  and  $p_3 = (3,0)$  by  $\alpha = \frac{\pi}{3}$ . Show your calculations.

## 2.3 Raytracing and Rasterization (2 Point)

Explain in your own words the differences between raytracing and rasterization.