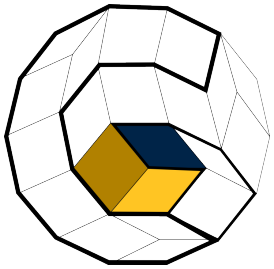
TU BRAUNSCHWEIG
Dr. Georgia Albuquerque
Institut für Computergraphik
Matthias Überheide

May 7, 2018



Real-time Computer Graphics, Summer 2018
Assignment 3

Present your solution to this exercise on Thursday, May 17, 2018.

The exercises are going to take place in the CIP pool, room G40 in Mühlenpfordstraße 23. Please make sure your solutions compile and run on the CIP pool computers. Note that you need a y-number, which can be obtained at the Gauß-IT-Zentrum, to use the computers. If for some reason you are not able to attend the exercise, you may send your solution to ecg@cg.cs.tu-bs.de instead.

This exercise and the corresponding version of the framework can be found on the lectures website (`http://www.cg.cs.tu-bs.de/teaching/lectures/ss18/ecg/`).

Theoretical Tasks

3.1 Transformations (20 Points)

For the following chains of transformations compute the combined transformation and give the orthonormal basis of the model space and its origin in world coordinates.

a) $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

b) $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

c) $\begin{pmatrix} 0.85 & 0.15 & -0.5 & 0 \\ 0.15 & 0.85 & 0.5 & 0 \\ 0.5 & -0.5 & 0.71 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

3.2 Short Presentation (Extra credit 10 Points)

Prepare a 5min talk about an interesting topic related to the current lecture. Send your topic proposal via email to ecg@cg.cs.tu-bs.de at least 24 hours before the presentation.

Practical Tasks

3.3 Organize geometry in a scene graph structure (30 Points)

Using vertex array objects through classes like the `OGLMeshObject` allows to render multiple instances of the same data. Using affine transformations and a scene graph structure these instances can be rendered at arbitrary positions in space.
`OGLSceneGraphNode` is a simple node class which allows the construction of scene graphs. Implement the missing parts in `OGLSceneGraphNode.cpp`. Additionally the transformation and projection has to be uploaded to the GPU. Complete the `OGLMeshObject::getShaderUniforms` to store the uniform location of the matrices `modelView` and `projection` and upload the correct transformation in `OGLMeshObject::render`. Make sure to create the desired transformation for the node in `ecg_ex03_a.cpp`. The final result should be a slightly rotated monkey head.

3.4 Scenegraph animation (10 Points)

In this task we will use a special scene graph node `OGLSGRotationalAnimator` which allows some simple animation. Implement `OGLSGRotationalAnimator::updateRotation` to get a rotating monkey head. Play with the parameters of `OGLSGRotationalAnimator`. Is the speed guaranteed to be the same on different devices? What happens for negative axis?

3.5 Implement a Camera Controller (30 Points)

To freely navigate the 3D environment we need to capture mouse and keyboard events and update transforms accordingly. The user input handling by GLFW is wrapped by the `GLFWContext` and is available through the observer adapter `GLFWInputEventHandler`. Implement the missing parts of `FreeCameraControler` to create a free flying camera which can be controlled with the right button of the mouse and WASD+RF keys.

3.6 Small Scene (10 Points)

Bringing everything together, complete `ecg_ex03_d.cpp`. Create and register a camera controller. Construct a small scene with a monkey head in the center and two armadillos and two bunny's rotating around it. Reuse the `OGLMeshObject`s for multiple instances. Select a nice camera starting angle. Feel free to extend the scene with additional objects or scene graph constructions but be aware that the current implementation can not detect circles and will produce infinite recursion (and will likely crash with a heap corruption).

3.7 Coding (Extra credit 10 Points)

Present one of the following:

- a bug in the framework
- a helpful test case to solve an exercise
- the implementation of an additional feature
- a nice demo using the current framework

Note that a short documentation is required and code has to be handed in. Please report bugs immediately (via email) so they can be fixed as soon as possible.