TU BRAUNSCHWEIG
Dr. Martin Eisemann
Institut für Computergraphik
Matthias Überheide

November 3, 2014

Real-time Computer Graphics SS2014
Assignment 1

Present your solution to this exercise on Monday, November 10th, 2014.

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 ueberheide@cg.cs.tu-bs.de instead.

In this first assignment we will look at the theoretical basics when using vector algebra and matrix multiplications. These are widely used within OpenGL for geometric transformations, texture operations, and many other things.
The first three tasks cover some important operations which you will have to solve *by hand*, while in the last task we will make use of the *glm* library, which provides a lot of useful tools when working with vectors and matrices. This library was specifically designed to be used in combination with OpenGL and GLSL, thus it provides a very similar interface and is easy to integrate into your OpenGL code. For further information, please have a look at `http://glm.g-truc.net/`.

1.1 Dot product (5 Points)

Compute the angle α (in degrees) between the vectors $v_0 = \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}$ and $v_1 = \begin{pmatrix} 2 \\ 4 \\ 1/2 \end{pmatrix}$.

1.2 Cross product (5 Points)

Given a vector $v_2 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$, compute two vectors v_3 and v_4 that are perpendicular to each other *and* to the vector v_2. Write down all possible solutions for the vectors v_3 and v_4, given the additional constraint, that the vector v_3 lies in the xy-Plane, i.e. the z-component of this vector is 0.

1.3 Matrix multiplication (5 Points)

Solve the following equations and try to interpret the operation described by the matrix:

- $\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$
- $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

- $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$
- $\begin{pmatrix} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \\ 1 \end{pmatrix}$

1.4 Using glm (5 Points)

In this task we will use the *glm* library for vector algebra computations. These are the very basis of OpenGL rendering and very important in the upcoming tasks.
Go to the lecture's website (`http://graphics.tu-bs.de/teaching/lectures/ss13/cg2/`), download the exercise stub for this assignment and extract it. As you can see, there are already a lot of files and folder set up. We make use of CMake as a build system, which provides a quite comfortable interface to extend and adapt your project and generate a proper makefile for it. The code for this exercise is already configured, so try to get it running. Once extracted, go into the `bin/` folder of this exercise. From there execute the command "`cmake ..`", which will tell the CMake build system to generate a makefile according to your project's configuration defined by `CMakeLists.txt` in the root folder of the project. This file defines the type, name and structure of your project as well as all basic dependencies and library paths.
After this operation completed successfully, you can make your project (`make`) and generate the executable `ex01`. Go ahead and try to generate your makefile and build your project.

Now that you've got the project up and running, open the file `src/Ex01.cpp` and implement code using *glm* solving the following problem:

Given three points, A, B, and C, forming a triangle (counter clockwise), compute the normal vector defined by that triangle and the angle between the triangle's normal and a vector V.

$$A = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix}, B = \begin{pmatrix} -1 \\ 3 \\ 2 \end{pmatrix}, C = \begin{pmatrix} 0 \\ 2 \\ 1 \end{pmatrix}, \text{and } V = \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}$$

Do not compute this by hand, but implement your solution within the method `triangleSolution` of `Ex01.cpp` using the *glm* library (see `http://glm.g-truc.net/` for more information about *glm*). Output your solution in the console as text.