

# ME/HCI/CS/CprE 557 - Computer Graphics

## Assignment 2

### Transformations with OpenGL

Rafael Radkowski  
Iowa State University  
rafael@iastate.edu  
Fall 2018

The goal of this homework is to practice model transformations using matrices and quaternions in homogeneous coordinate space. You should be able to a) prepare code that represents a model transformation matrix, a view matrix, and a projection, b) link the variables on the host system with variables used in your GPU program, and c) transform models using matrix manipulations functions and quaternions.

Use the template *Assignment2.Transformations.cpp* to complete this homework.

#### Problem 1

Use GML matrices and functions to create a model matrix, a projection matrix, and a view matrix. Initialize the matrices with values as you see fit. You can either use GML functions for this purpose or complete the matrix content manually.

#### Problem 2

Connect your host program variables (model, projection, and view matrix) with its related variables in a GPU program. A template GPU program is given in *ShaderCode.h*. Note that this code is incomplete; variable types and names are missing.

```
1
2     #version 410 core
3
4     uniform [Type] [variable_name];
5     uniform [Type] [variable_name];
6     uniform [Type] [variable_name];
7
8     in vec3 in_Position;
9     in vec3 in_Color;
10
11     out vec3 pass_Color;
12
13     void main(void)
14     {
15         gl_Position = projectionMatrix * viewMatrix * modelMatrix *
vec4(in_Position, 1.0);
```

```
16     pass_Color = in_Color;
17 }
```

- Complete the shader code in ShaderCode.h accordingly so that it works.
- In your host program, fetch the location of all uniform variables.
- In your host program, write the content of your view, model, and projection matrix to your gpu program.

### Problem 3

The code template *Assignment2\_Transformations.cpp* contains geometry for one box. Prepare three transformations for this object.

- Prepare a matrix that rotates the object by  $45^\circ$  around the y-axis and moves it by 0.5 (units) along the z-axis.
- Describe the same transformation using a quaternion. Note that the final outcome must still be a matrix which you can copy to your graphics card.
- Build a transformation matrix manually by determining the unit vectors and the origin.

Copy the content of all matrices to your gpu program, but only one at any time. Use comments (//) to remove the other solutions temporary.

### Deliverable

The following deliverables must be uploaded to Canvas

- The entire code.
- Screenshots that show each transformation.

**Due data: Friday, Sep. 28th, 2018, 8:00 pm**

### Grading

The following rubric will be used for grading. Max. 10 points can be earned.

- Code to create the view, model, and projection matrix (2pt).
- Code that links the GPU program with your host program (2pt).
- Code that copies the matrices from the host program to the gpu (2pt).
- The three transformations are correct (3pt).
- The entire code compiles and works as intended (1t).