Homework #2: Transformations Library

Due Date: 11:59pm Tuesday 8 October 2013

Look on the internet for a vector/matrix library (e.g. you can use eigen or glm or find another one or write your own) and get familiar with it and make sure it includes the following components/classes and add them if necessary:

- Vector3 (3D vector)
- Vector4 (4D homogeneous vector)
- Matrix3 (3x3 matrix)
- Matrix4 (4x4 matrix)
- Conversion from vector3 to vector4 and vice versa
- Matrix/Vector Multiplication
- Matrix/Matrix Multiplication
- Scalar multiplication
- Addition
- Dot product
- Transpose
- Matrix Inversion

Transformations

Extend the library to include the following transformations (you **must** implement them even if they are already in the library!)

- Transformations on 3D and 4D homogeneous vectors:
  - Translation: which is defined by three parameters $[t_x, t_y, t_z]$
  - Rotation: which is defined by an axis of rotation and an angle $[a_x, a_y, a_z] + [\theta]$
  - Scaling: which is defined by three parameters $[s_x, s_y, s_z]$

Write a C++ program ***transform4x4*** that reads in a series of transformations (rotation, translation, scaling) and outputs the 4x4 matrix that is equivalent to these transformations. The input should be through stdin and the output should be on stdout.

For example:

```
transform4x4 < transformations.txt | less
```

will input a file called “transformations.txt” and should output a 4x4 transformation matrix that is equivalent to all the input transformations.

Input

The transformations are one per line and defined as follows:

```
translation tx ty tz
scaling sx sy sz
rotation x y z theta
```

where the rotation is a rotation around a vector (*x, y, z*) with an angle *theta* in *radians*. The transformations should be applied such that the top transformation is the leftmost matrix and the bottom transformation is the rightmost (the one that is applied first). For example:

```
translation 1 0 0
scaling 1 2 1
rotation 1 1 2 0.5
```

should correspond to a matrix *T S R* where the rotation is applied first, followed by the scaling and then the translation.

To compute the rotation matrix from the axis of rotation and the angle of rotation, you could use the following formula (or any other formula you like):

$$\begin{pmatrix} x^2 + (1 - x^2)\cos\theta & xy(1 - \cos\theta) - z(\sin\theta) & xz(1-\cos\theta) + y(\sin\theta) \\ xy(1 - \cos\theta) + z(\sin\theta) & y^2 + (1 - y^2)\cos\theta & yz(1-\cos\theta) - x(\sin\theta) \\ xz(1 - \cos\theta) - y(\sin\theta) & yz(1-\cos\theta) + x(\sin\theta) & z^2 + (1 - z^2)\cos\theta \end{pmatrix}$$

where (*x, y, z*) is the **normalized** axis of rotation i.e. make sure you **normalize** the vector to have unit norm before using the formula above.

Instructions

- All code should be implemented in C++ under Linux.
- Please submit your homework in one .zip file named as follows: *HW##.FirstName.LastName.zip*, so for example if your name is Mohamed Aly and this is homework #1, then the file name should be *HW01.Mohamed.Aly.zip*.

- Please include all your code and sample output in the zip file, with a README file to explain what you did. Failure to follow these instructions will cause deductions from your grade.
- You are allowed to discuss the problems among yourselves. However, **copying** any part of the code will result a grade of **ZERO**. No exceptions.

Acknowledgment

This homework is adapted from CS 171 at Caltech.