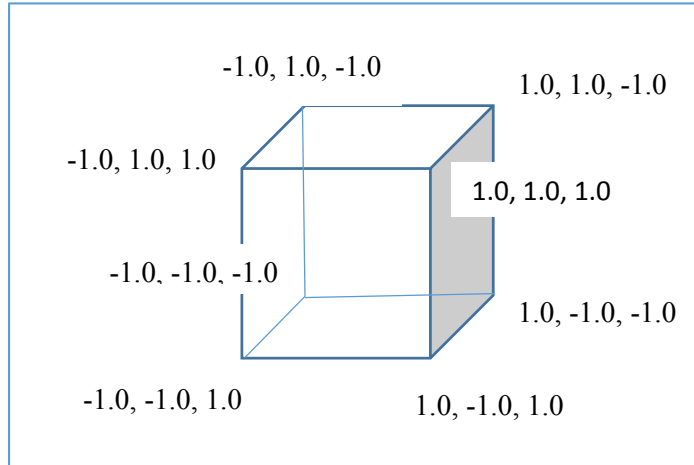
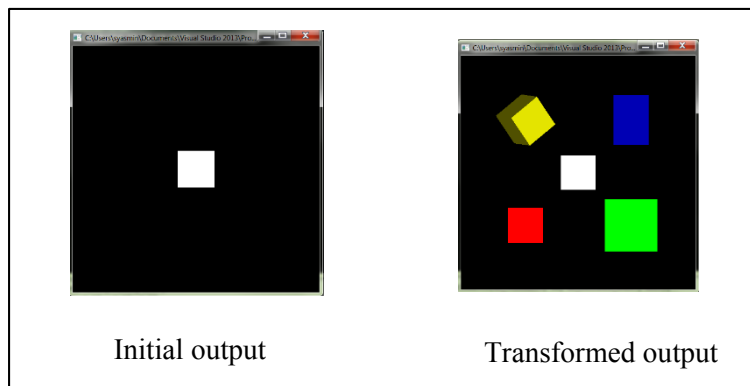


Assignment 3 (Transformed Cube)

In this assignment, you'll apply different transformation on a unit cube described as follows:



Initially, the cube has been drawn at origin $(0.0, 0.0, 0.0)$ in white color. You'll apply different transformation and for each transformation, you'll draw the transformed cubes. After applying a number of transformation described below, the final result will be as follows:



Download the program file "TransformedCube.cpp" and the associated shader files. In the .cpp file, there are global variables, colors and positions which define the corresponding color and position for each cube drawn onto the screen. The initial cube in white is drawn at position $(0.0, 0.0, 0.0)$, this uses the first four components of positions array as well as the first four components of colors array defined as follows:

```
static const GLfloat positions[] = { 0.0f, 0.0f, 0.0f, 1.0f, // position for white cube
                                     -3.0f, -3.0f, 0.0f, 1.0f, // position for red cube
                                     3.0f, -3.0f, 0.0f, 1.0f, // position for green cube
                                     3.0f, 3.0f, 0.0f, 1.0f, // position for blue cube
                                     -3.0f, 3.0f, 0.0f, 1.0f }; // position for yellow cube

static const GLfloat colors[] = { 1.0f, 1.0f, 1.0f, 1.0f, //white
                                   1.0f, 0.0f, 0.0f, 1.0f, // red
                                   0.0f, 1.0f, 0.0f, 1.0f, // green
                                   0.0f, 0.0f, 1.0f, 1.0f, // blue
                                   1.0f, 1.0f, 0.0f, 1.0f }; // yellow
```

Similarly, other cubes (red, green, blue and yellow respectively) to be drawn from consecutive positions and colors defined in those two arrays.

Now, the white cube is positioned at origin. Apply the following different transformation on the initial cube (white):

Draw Red cube: Translate the initial cube to the position defined in the position array for the red cube, i.e. (-3.0, -3.0, 0.0), apply color (red) from colors array.

Draw Green cube: Apply the scale factor S (1.5, 1.5, 1.0) to the initial cube. Then translate the scaled cube to the position defined in the position array for green cube, i.e., (3.0, -3.0, 0.0), apply color (green) from colors array.

Draw Blue cube: Apply a rotation of 45 degree around x-axis to the initial white cube. Then translate the rotated cube to the position defined in the position array for blue cube, i.e. (3.0, 3.0, 0.0), apply color (blue) from colors array

Draw yellow cube: Apply a rotation of 45 degree around an arbitrary axis (1.0, 2.0, 3.0) to the initial white cube. Then translate the rotated cube to the position defined in the position array for yellow cube, i.e. (-3.0, 3.0, 0.0), apply color (yellow) from colors array.

For drawing yellow cube, you need to take a look at week 3's lecture note.

The composite matrix for rotation of an object by angle 'θ' (in our case, 45 degree) around any arbitrary axis (1.0f, 2.0f, 3.0f) is as follows:

$$R_{\text{arbitrary}}(\theta) = R_x(-\phi) R_{yz}(\phi) R_y(\phi) R_{xz}(\theta) R_y(-\phi) R_x(\phi)$$

Then you'll need to place the rotated cube to the last position defined in positions array, i.e., (-3.0f, 3.0f, 0.0f, 1.0f)

Hence the composite matrix will be $M_{\text{composite}} = T R_{\text{arbitrary}}$

For rotating about axis (1.0f, 2.0f, 3.0f) , $\cos(\phi_{yz}) = \frac{3}{\sqrt{13}}$, $\cos(\phi_{xz}) = \frac{\sqrt{13}}{14}$

You can figure out by taking a look at the lecture note.

Submission:

Place your solution in a zipped file named with your last name followed by the first initial of your first name followed by 3 (ex: **CSCD396YasminS3.zip**) and submit the solution via canvas.

Thus, your zip should contain the following:

- A file named **TransformedCube.cpp**, and shader files.

Submission deadline is Monday, February 13, 11:59 pm. This assignment carries a weightage of 15% of this course.