

Exercise List #4

Victor F. Ferrari - RA 187890

v187890@dac.unicamp.br

MO814A/MC937A - Topics in Computer Graphics

UNIVERSIDADE ESTADUAL DE CAMPINAS

May 1, 2020

1 OpenGL Geometric Transformations

The first assignment of the list was to implement an OpenGL application that allows the user to apply geometric transformations to 3D shapes.

1.1 Details

The application was made using C++ with the GLFW, GLEW and GLM external libraries. Building is done via `cmake`, followed by `make`. The application does not take any arguments, with every interaction happening during the execution.

The application consists of 9 3D shapes that can be interacted with via the keyboard, and a camera which can also be moved. The possible transformations are **rotation**, **scaling** and **translation**, in any axis, in local coordinates. If all shapes are selected, the transformations are applied in world coordinates. Every face has a different color, to ease visualization of the transformations.

When a shape is selected, it is highlighted by **scaling up** with a factor of 1.1. The original idea was to either create a contour or insert the x,y and z axes lines, but that wasn't practical with the current code structure and without much shader knowledge (for the contour), so the scaling was added for highlighting. When another shape is selected, it scales back by that same factor.

1.2 Usage

The controls are as specified in the project definition, this section repeating that information. The application starts in **camera mode**. In this mode, the arrow keys are used to move the camera along the X and Y axis, and the period and comma keys (".", ",",) are used to move backward and forward with the camera, respectively. The user can exit or reenter camera mode by pressing the "C" key at any time.

The user can select one of the 9 shapes via the number keys 1-9. The 0 key is used to select all shapes, and the transformations are then applied in world coordinates, instead of local coordinates. Every transformation from here on will follow those rules. It is possible to recognize which shape is selected because it is scaled by a factor of 1.1.

The same keys used by camera mode are also used for **translating** shapes, the only difference being in the definition of "forward" and "backward". The comma key (",") is used to move a shape forward, that is, in the direction of the camera (if it is facing the shapes), and the period key (".") is used to move a shape backward, that is, in the opposite direction from the camera (if it is facing the shapes).

Scaling in the X, Y and Z axis can be done with the X, Y and Z keys, respectively. Pressing just the keys **decreases** the respective dimension of the shape by a factor of 0.9, while pressing the keys along with **shift** **increases** the respective dimension by a factor of 1.1. This transformation also works in camera mode, with the selected shape(s).

Rotation in the X, Y and Z axis can be done with the S, Q and A keys, respectively. The rotation is **counterclockwise**, by an angle of 15 degrees. The shape rotates around the center of its bounding box, so it might not be perfect depending on the shape and combination of rotation actions. Numerical errors are also possible. This transformation also works in camera mode, with the selected shape(s).

All of the factors described above can be changed by altering constants in file `input.cpp`. Finally, the escape key can be used to close the application.

2 Questions

Question 1

Write the translation matrix that moves the point (X_0, Y_0) to the point (X_1, Y_1) . The X offset is $u = X_1 - X_0$, and the Y offset is $v = Y_1 - Y_0$. So, the resulting translation matrix is:

$$M = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & X_1 - X_0 \\ 0 & 1 & Y_1 - Y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplying this matrix M and the point (X_0, Y_0) results in point (X_1, Y_1) .

Question 2

Write the rotation matrices for angles 45° , 60° , 90° , 180° and -90° .

A rotation matrix is given by the following formula:

$$R = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So, the rotation matrix for different angles are:

- 45°

$$\sin(45) = \frac{\sqrt{2}}{2} \quad \cos(45) = \frac{\sqrt{2}}{2}$$

$$R = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

• 60°

$$\sin(60) = \frac{\sqrt{3}}{2} \quad \cos(60) = \frac{1}{2}$$

$$R = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

• 90°

$$\sin(90) = 1 \quad \cos(90) = 0$$

$$R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

• 180°

$$\sin(180) = 0 \quad \cos(180) = -1$$

$$R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

• -90°

$$\sin(-90) = -1 \quad \cos(-90) = 0$$

$$R = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Question 3

Write the scaling matrix that leads (X_0, Y_0) to (X_1, Y_1) (suppose none of these numbers are zero).

The X scaling factor is $u = \frac{X_1}{X_0}$, and the Y scaling factor is $v = \frac{Y_1}{Y_0}$. So, the resulting scaling matrix is:

$$M = \begin{bmatrix} u & 0 & 0 \\ 0 & v & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{X_1}{X_0} & 0 & 0 \\ 0 & \frac{Y_1}{Y_0} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Multiplying this matrix M and the point (X_0, Y_0) results in point (X_1, Y_1) .

Question 4

Let R be the rotation by 90° around the origin, and T be the translation by the vector $(2,3)$. Calculate the matrices for transformations $A = TR$ and $B = RT$. Are the results the same? Why?.

$$\sin(90) = 1 \quad \cos(90) = 0$$

$$R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

Doing $A = TR$ and $B = RT$, we have:

$$A = \begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & -1 & -3 \\ 1 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

As can be seen, the results are different. This happens because translation and rotation are not commutative. Rotation always happens in relation to the origin point, and when a translation happens, the result of such rotation is then different than rotating before translating, since the distance to the origin changes.

Question 5

Prove that the multiplication of transformation matrices for each of the following sequences is commutative:

- a) Two successive rotations.
- b) Two successive translations.
- c) Two successive scalings.

The answers to this question will, as with the ones before, consider 2D transformations. 3D transformations obey the exact same rules, but with an added dimension, so it was chosen to not demonstrate with 3D transformations.

- Two successive rotations.

A rotation matrix by angle α is given by the following formula:

$$R_\alpha = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So, multiplying :

$$R_\alpha R_\beta = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos \beta & -\sin \beta & 0 \\ \sin \beta & \cos \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\alpha R_\beta = \begin{bmatrix} \cos \alpha \cos \beta - \sin \alpha \sin \beta & -\sin \beta \cos \alpha - \sin \alpha \cos \beta & 0 \\ \sin \alpha \cos \beta + \sin \beta \cos \alpha & \cos \alpha \cos \beta + \sin \alpha \sin \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_\alpha R_\beta = \begin{bmatrix} \cos \alpha + \beta & -\sin \alpha + \beta & 0 \\ \sin \alpha + \beta & \cos \alpha + \beta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As can be seen by the previous equation, two successive rotations are the same as rotating by the sum of their angles. Since sum is a commutative operation ($a + b = b + a$), rotation is also commutative.

$$R_\beta R_\alpha = \begin{bmatrix} \cos \beta + \alpha & -\sin \beta + \alpha & 0 \\ \sin \beta + \alpha & \cos \beta + \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Two successive translations.

A translation matrix is given by the following formula, with u being the X offset and v being the Y offset:

$$T = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix}$$

So, multiplying:

$$T_1 T_2 = \begin{bmatrix} 1 & 0 & u_1 \\ 0 & 1 & v_1 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & u_2 \\ 0 & 1 & v_2 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & u_1 + u_2 \\ 0 & 1 & v_1 + v_2 \\ 0 & 0 & 1 \end{bmatrix}$$

As can be seen by the previous equation, two successive translations are the same as translating by the sum of their offsets. Since sum is a commutative operation ($a + b = b + a$), translation is also commutative.

$$T_2 T_1 = \begin{bmatrix} 1 & 0 & u_2 + u_1 \\ 0 & 1 & v_2 + v_1 \\ 0 & 0 & 1 \end{bmatrix}$$

- Two successive scalings.

A scaling matrix is given by the following formula, with s_x being the X scaling factor and s_y being the Y scaling factor:

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

So, multiplying:

$$S_1 S_2 = \begin{bmatrix} s_{x1} & 0 & 0 \\ 0 & s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{x2} & 0 & 0 \\ 0 & s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x1}s_{x2} & 0 & 0 \\ 0 & s_{y1}s_{y2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As can be seen by the previous equation, two successive scalings are the same as scaling by the product of their scaling factors. Since scalar product is a commutative operation ($ab = ba$), scaling is also commutative.

$$S_2 S_1 = \begin{bmatrix} s_{x2}s_{x1} & 0 & 0 \\ 0 & s_{y2}s_{y1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$