Exercise List 4: Geometric Transformations

MO814A/MC937A - Topics in Computer Graphics

Prof. Aurea Soriano-Vargas (aurea.soriano@ic.unicamp.br)
Deadline: 23/04 - 11:59pm
Send by google-classroom.

April 2020

1 OpenGL

1.1 3D shapes

Create a scene with 9 basic 3D shapes which can be seen from the camera. The shapes have to be 3D and not flat. For example: cubes, pyramids, cones. Each face should have a different color so transformations can be seen more easily.

1.2 Interaction with the camera

Respond to user interaction from the keyboard to move the camera in the scene with the arrow keys (translate). (left and right in the x-coordinate, up and down in the y-coordinate). Clicking the 'c' button should on/off the camera movements.

1.3 Geometrical transformations

Respond to user interaction from the keyboard. The following actions should be implemented:

1.4 Interaction setup

The user should be able to select a shape by pressing the number keys between '1' and '9'. Highlight the selected shape by displaying its local coordinate system overlaid on it. Geometrical transformations should be applied to the selected shape only. By pressing the '0' key all the shapes are selected and the transformations should be computed for all of them using the global coordinate system.

1.5 Scaling

Object scaling should be activated by the following shortcuts.

- **x:** decrease the width of the shape by a factor of 0.9
- **Shift + x:** increase the width of the shape by a factor of 1.1
- **y:** decrease the height of the shape by a factor of 0.9
- **Shift + y:** increase the height of the shape by a factor of 1.1

- **z:** decrease the depth of the shape by a factor of 0.9
- **Shift + z**: increase the depth of the shape by a factor of 1.1

1.6 Rotation

The center of the rotation should be the origin of the coordinate system of the shape.

- **s:** rotate the current shape counterclockwise about the x-axis
- **q:** rotate the current shape counterclockwise about the y-axis
- **a:** rotate the current shape counterclockwise about the z-axis

1.7 Translation

Using arrow keys. Notice the camera mode must be off.

- **right:** move the shape to the right
- **left:** move the shape to the left
- **up:** move the shape up
- **down:** move the shape down
- **comma ',':** move the shape forward
- **period '.'**: move the shape backward

1.8 Implementation details

- Programming languages allowed: C++ (suggested) and Python.
- C++ projects should compile using a Makefile, you can write one or use CMake (suggested).
- Python projects should be implemented using Python 3.0 or superior.
- Implement your own version of geometrical transformation functions. You are free to use linear algebra libraries such as Eigen or Numpy. You are not allowed to use OpenGL functions such as glRotate, glTranslate, and glScale.
- Projects will be tested on the Linux operating system.

2 Questions

1. Write the translation matrix that moves the point (X_0, Y_0) to the point (X_1, Y_1).
2. Write the rotation matrices for angles 45^o, 60^o, 90^o, 180^o, and -90^o.
3. Write the scaling matrix that leads (X_0, Y_0) to (X_1, Y_1). (Suppose none of these numbers are zero).
4. Let R be the rotation by 90^o 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?

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.

3 Report

Besides the source code, you have to prepare a report containing:

1. Name, RA, and student email.
2. A description of how to use the application.

If you do not know where to start, or if you are completely lost, please contact me immediately.