Transformation and Modelling (Assignment - 03)

Hammad Akhtar (2013060)

Abstract

Perform Model, View, Projection transformations on the cube rendered. Model a torus using its parametric equations. Implement a trackball with zoom in and zoom out functionality.

1 Transformations

Transformations to be performed

1. Rotate about z-axis by 15 degrees.
    1. model = glm::rotate(model, glm::radians(15.0f), glm::vec3(0.0f, 0.0f, 1.0f));
2. Scale along y-axis by 2.0.
    1. model = glm::scale(model, glm::vec3(1.0f, 2.0f, 1.0f));
3. Translate by (20, 10).
    1. model = glm::translate(model, glm::vec3(20.0f, 10.0f, 0.0f));

2 Composite Transformations

1. Apply transformation 1, 2, 3
    1. model = glm::translate(model, glm::vec3(20.0f, 10.0f, 0.0f));
    2. model = glm::scale(model, glm::vec3(1.0f, 2.0f, 1.0f));
    3. model = glm::rotate(model, glm::radians(15.0f), glm::vec3(0.0f, 0.0f, 1.0f));
2. Apply transformation 3, 2, 1

model = glm::rotate(model, glm::radians(15.0f), glm::vec3(0.0f, 0.0f, 1.0f));
model = glm::scale(model, glm::vec3(1.0f, 2.0f, 1.0f));
model = glm::translate(model, glm::vec3(20.0f, 10.0f, 0.0f));

3 Implement glm::lookAt()

Implement myLookAt() function which takes camera position, centre position and up vector as argument.

4 Projection

1. Orthographic projection
    1. Top
        1. eye = (0.0, 100.0, 0.0)
        2. centre = (0.0, 0.0, 0.0)
        3. up = (0.0, 0.0, 1.0)
    2. front elevation
        1. eye = (0.0, 0.0, 100.0)
        2. centre = (0.0, 0.0, 0.0)
        3. up = (0.0, 1.0, 0.0)
    3. left side elevation
        1. eye = (-100.0, 0.0, 0.0)
        2. centre = (0.0, 0.0, 0.0)
        3. up = (0.0, 1.0, 0.0)
    4. right side elevation
        1. eye = (20.0, 0.0, 0.0)
        2. centre = (0.0, 0.0, 0.0)
        3. up = (0.0, 0.0, 1.0)
    5. isometric elevation
        1. eye = (100, 100, 100)
        2. centre = (0, 0, 0)
        3. up = (0.0, 1.0, 0.0)
2. Perspective projection : glm::perspective(glm::radians(45.0f), aspect, 0.10f, 1000.0f);
    1. one point :
        1. eye = (20.0f, 0.0f, 100.0f)
        2. centre = (20.0f, 0.0f, 0.0f)
        3. up = (0.0f, 1.0f, 0.0f)
    2. Two point:
        1. eye = (80.0f, 0.0f, 80.0f)
        2. centre = (0.0f, 0.0f, 0.0f)
        3. up = (0.0f, 1.0f, 0.0f)
    3. Three point:
        1. eye = (30.0f, 40.0f, 30.0f)
        2. centre = (-10.0f, -10.0f, -10.0f)
        3. up = (0.0f, 1.0f, 0.0f)

5 Torus

Torus is given by following equations

x = (R + r cosv) cos u

y = r sin v

z = (R + r cosv)sin u

u and v are incremented by 10 starting fro 0 till 360

we join (u,v), (u+1, v), (u+1, v+1), (u,v+1) in order. This models the surface torus.

6 Trackball (Bonus)

When user left clicks , movement along x and y axis of the view port is noted until left button is released. On the basis of offset yaw and pitch angles are updated and camera position is determined by converting spherical coordinates to cartesian coordinates.

3 Compilation Instructions

- Clone the Assignment03_code directory to the desired location on your system.
- Open the terminal and cd into the Assignment03_code directory
- execute **make** command
- execute **./assignment03** command
- window will pop up with a cube. Trackball mode is default.

References

SHADER UTILITY CLASS (USED FOR READING SHADERS FROM A FILE AND CREATING SHADER PROGRAMS)
http://learnopengl.com/code_viewer.php?type=header&code=shader
http://learnopengl.com/#!Getting-started/Camera

SCREENSHOTS

SCREENSHOTS CAN BE FOUND IN SCREENSHOTS FOLDER PROPERLY SEGREGATED INTO SUB FOLDERS