CSc 305: Assignment 2

1 Objectives

- Experiment with implementations of transformations and projections

2 Implementation

You will be working with the Qt framework code attached to this assignment. You will be required to complete multiple methods within this codebase.

2.1 Part 1: matrix.cpp

In BasicOpenGLView two matrix members are included: mProjectionMatrix and mViewMatrix. You need to add methods to build proper view and projection matrices based on user input. These two matrices are already (as identity matrices) used in the paintGL function and passed on to the Geometry::draw() function. In the user interface the user should be able to switch between *perspective* and *orthographic* projection. Depending on the chosen method, you have to store the right values in mProjectionMatrix. Make sure these values are updated, when the Window is resized, and when the mode is changed.

- Implement the functions to create the projection matrix
    - perspective projection: takes the field of view, image ratio, near plane, far plane
    - orthographic projection: takes left, right, top, bottom, near plane, far plane
- Implement the lookAt function to create the view matrix: takes: eye position, object position and the up vector

2.2 Part 2: geometry.cpp

In the geometry, sliders are used to control the scale, rotation and translation of objects and a toggle to indicate if these transformations have to be pre-multiplied or post-multiplied in code. BasicOpenGLView already has a member mUsePostMultiply, for which you should need to store the method to be used. It is already passed to the geometry class when needed. Geometry.cpp has already the functions for all the transformations that you must complete. In the function createModelMatrix, create the actual model matrix for the geometry and store it in mModelMatrix.

- Implement the following methods in geometry.cpp NOTE: The method skeletons are already present. Store the transformations in separate values (e.g. a Vector3 for the aggregated Translation). Only the updateModelMatrix() function should then create the proper matrices and multiply them to build mModelMatrix
- Complete the implementation of the following functions
    - translate
    - scaleX
    - scaleY

- scaleZ
- rotate
- rotate
- rotate
- createModelMatrix

2.3 Part 3: Understanding and extending the codebase

2.3.1 Explain the Viewing Pipeline

In lecture slides 07_Viewing, slide #31 there is an overview of the *Viewing Pipeline*. Describe the implementation of first three transformations in this slide within the Assignment 2 codebase. Describe how are these three steps are implemented in terms of the model-view transformation matrix and the projection matrix, with your description leveraging the specific variables, functions and files within the codebase.

Your medium of explanation is up to you, you will be graded on your ability to convey the concepts and the demonstration of your understanding of both the *Viewing Pipeline* concept and the codebase itself. Some ideas for mediums of explanation: UML and sequence diagrams, flowchart, drawings, diagrams, video, song, interpretive dance (ok, maybe not but you get the idea…)

2.3.2 Extend the projection implementation

Now that you understand the implementation of the viewing pipeline within the codebase, you are to expand the current projection functionality. In the user interface there is a checkbox to allow you to switch between *perspective* and *orthographic* projection. If you have not noticed already, the interface only works if the *orthographic projection* is set. You must extend the codebase to support the switch between *perspective* and *orthographic* projection.

3 Submission details

Submit via Assignment 1 link on conneX:

- A zip file containing you're the full Qt project with required function implementations complete.
- A document containing your codebase explanation

4 Marking Rubric

Part	Item	Weight
Part 1	Projection functions	26
	lookAt Function	10
Part 2	translate, scale, rotate functions	9
	createModelMatrix function	17
Part 3	Explanation	25
	Perspective projection	13
TOTAL MARKS		100