Computer Science 470
Winter 2019-2020
Assignment 1

3-D Transformations and Perspective Projection

Due: Tuesday, January 7, 2020

The purpose of this assignment is to give you experience implementing 3-D transformations and perspective projection.

This is an individual (as opposed to team) assignment. Your assignment will be written in Python and using the provided Tkinter framework.

Specifically, you will build a system that allows you to see (render) and move (scale, translate, rotate) a single object (a pyramid) in a 3-D wire-frame viewing system

You should do the following:

- Use the definition of the pyramid object provided below. From these underlying points:
    a. Build a list of points used by the object. In the sample code this object is called PyramidPointCloud and is used to scale, translate, and rotate the object.
    b. Build a list of points for each polygon. Used to draw each of the lines making up a polygon.
    c. Build a list of polygons for the object. In the sample code this object is called Pyramid and will be used to draw each of the polygons making up the object.

        apex = [0,50,100]
        base1 = [-50,-50,50]
        base2 = [50,-50,50]
        base3 = [50,-50,150]
        base4 = [-50,-50,150]

- Implement code for perspective viewing of three-dimensional images.
    - We will use a Left Handed System (LHS) for viewing. Objects in +Z. Viewer in –Z
    - The viewer is situated at (0,0,-d) in the World Coordinate System. Use d = 500

- Implement code for conversion to display coordinates. In Tkinter the origin is in the upper left hand corner of the canvas, so before you can actually draw a line on the canvas the projected points must be modified to display coordinates. (Essentially what you must do is 'mathematically move' the origin from the upper left hand corner to the center of the drawing canvas.)

- Implement code for uniform 3-D scaling. By 'uniform' I mean we will scale in all directions equally (by some scalar value).

- Implement code for 3-D translation in any direction specified by an arbitrary 3-D vector.

- Implement rotation along each primary axis: X, Y, and Z. Note that these rotation will NOT be in-place rotations. Instead the rotations will be defined with respect to the World Coordinate System origin.

- Implement a "reset" function to return the pyramid to its original location

- To drive all of the above and place it into a GUI, use the skeleton I've provided on Moodle

Additional Requirements:

1. You must work independently and develop your own code in accordance with the collaboration policy in the course handout.

2. You ***must*** thoroughly document your program. This includes comments at the beginning of your main program with: your name, your student number, the date, the assignment number, and a brief description of what the program does. Comments should also be present throughout the program to explain what each part does. It is especially important to place a "block" of comments at the beginning of each class or method.

3. Your program must be written in Python, run in Thonny, and not use any built-in graphics functions above the "line draw" level. Use of the numPy library is not allow for this program.

Submitting your assignment for grading:

As previously discussed, each student will be required to meet with me for approximately 15 minutes on Tuesday, January 7th to demo their program to me and explain how it works.

A copy of the program should be emailed to me (mike@LaTech.edu) **no later than 8am on Tuesday, January 7th** with the subject line "CSC 470 : Assignment 1 : <your name>" where <your name> is replaced with your actual name. So, for example, if Luke Skywalker were taking this course the subject line on his email would be "CSC 470 : Assignment 1 : Luke Skywalker".

A signup sheet will be placed outside my door no later than Monday, January 6th.