

In place 3-D Transformations in a Multiple Object Environment with Perspective Projection

Due: Thursday, January 10, 2019

The purpose of this assignment is to give you experience implementing in-place 3-D transformations in a multiple object environment. This assignment is an extension of the work you did in Assignment 1 and will use many of the same concepts (such as point clouds, polygon lists, and perspective projection) as Assignment 1. While you may want to reorganize your code to be more object oriented (now that your environment will contain multiple objects), I expect that there will be a significant amount of code re-use between assignment 1 and assignment 2.

Please remember, this is an individual (as opposed to team) assignment.

You should do the following:

- Correct any problems with your Assignment 1 code, as the Assignment 1 code forms the basis of the code for this assignment.
- Populate your virtual world with 3 or 4 simple objects – I suggest using pyramids and cubes.
- Implement an ability to select the individual objects in your world. This can be via keyboard commands or onscreen buttons or similar mechanism. The selected object should be highlighted in some way – for example by drawing the edges of the object using thicker lines or some special color or similar mechanism.
- Implement code for in-place 3-D transformations – translation (x,y,z), scaling (smaller and larger – symmetric in all dimensions), rotation (z, y, z in both positive and negative directions). These transformation should REPLACE the non-in-place transformations of Assignment 1.

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 Thursday, January 10th to demo his or her program to me and explain how it works.

A copy of the program should be emailed to me (mike@LaTech.edu) **no later than 9am on Thursday, January 10th** with the subject line “CSC 470 : Assignment 2 : <your name>” where <your name> is replaced with your actual name. So, for example, if Leia Organa were taking this course the subject line on her email would be “CSC 470 : Assignment 2 : Leia Organa”.

A sign-up sheet will be placed outside my door no later than Wednesday, January 9th.