[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 03: 3D OBJECT TRANSFORMATIONS

THIS IS A ONE WEEK ASSIGNMENT WITH SUBMISSION DATE AS 6th MARCH, 2020.

The aim for this assignment is to put your concepts and mathematical understanding to practice concepts for representing, drawing and transforming 3D object in a world coordinate space/system using a standard graphics library. Apart from implementation of basic operations such as translation, rotation, scaling and shearing of objects, this assignment will involve writing code for orthographic and perspective projections, basic camera motion and changes to its field of view. This assignment will also involve dealing with basic windowing functions such as keyboard interfacing and changing window size.

IMPORTANT:

There is no base/starting code package for this assignment. You have to start from scratch and develop code using OpenGL, GLU and GLUT libraries in C/C++. You code package should compile and run WITHOUT ANY MODIFICATION on general linux machines that are available for students @SoC. You may use main.cpp for either of the two previous assignments for reference.

OS required: Linux, Compiler: C/C++, Build procedure: using make file.

Scenario: You will have one object at a time in your world coordinate system, say a cube. You have to develop code to display it under two different types of projection (perspective and orthogonal). Furthermore you have to develop code to move it in the world coordinate system interactively with the use of a keyboard.

Objectives:

- Instantiate and display a Window of desired size using GLUT library.
- Write code to place a simple 3D object (say a cube) in your world coordinate system **(WCS)** and project it onto the screen for the window created earlier. Center (of mass) for the object should coincide with the origin for the world coordinate system. Initially this object should be a cube with **each of its side is colored differently** and parallel to X-o-Y, Y-o-Z or Z-o-X plane for **(WCS).**
- Provide user interactivity for all of the following tasks through keyboard inputs.
- Toggle the type of projection from perspective to orthographic and vice-a-versa.
- Translate, rotate or scale currently displayed object from where and how it is placed in the world coordinate system **(WCS)** and with respect to it **(i.e., w.r.to WCS)** whenever corresponding key (designated/ assigned/ specified by you) is pressed on the keyboard.
- Write code to clear out all transformations and display current object as displayed originally.
- Translate camera. **[OPTIONAL only for 4050]**
- **[OPTIONALLY]** Change camera's field of view.
- **[OPTIONALLY]** Write code to switch display object to a cylinder, sphere, torus or cone.

[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 03: 3D OBJECT TRANSFORMATIONS

Tasks and weightage in NUTSHELL:

BONUS scores will be clamped at 20%.

CPSC4050	CPSC6050
A. (5%) Instantiate and display a Window of desired size using GLUT library.	**A. (5%)** Instantiate and display a Window of desired size using GLUT library.
B. (15%) Write code to place a CUBE as explained above in your world coordinate system **(WCS)** and project it onto the screen.	**B. (15%)** Write code to place a CUBE as explained above in your world coordinate system **(WCS)** and project it onto the screen.
C. (10%) Provide user interactivity for all of the following tasks through keyboard inputs. Document your 'key' assignments/ associations for different tasks in a table like the one in the Appendix below and include it in your project report.	**C. (10%)** Provide user interactivity for all of the following tasks through keyboard inputs. Document your 'key' assignments/ associations for different tasks in a table like the one in the Appendix below and include it in your project report.
D. (5%) Toggle the type of projection from perspective to orthographic and vice-a-versa.	**D. (5%)** Toggle the type of projection from perspective to orthographic and vice-a-versa.
E. (60%) Translate, rotate or scale currently displayed object **as explained above**. Pay extra attention to changing the sequence of operations and how you must combine it with object transformation to camera coordinate system. Translations should move the object by a fixed amount of distance at each press. You should move the object to left, right, top, bottom, closer or further in the world coordinate system w.r.to its origin and axes. Rotations must rotate the object with a fixed amount at each key press. You should rotate the object anticlockwise around the X, Y or Z axis for the world coordinate system.	**E. (45%)** Translate, rotate or scale currently displayed object **as explained above**. Pay extra attention to changing the sequence of operations and how you must combine it with object transformation to camera coordinate system. Translations should move the object by a fixed amount of distance at each press. You should move the object to left, right, top, bottom, closer or further in the world coordinate system w.r.to its origin and axes. Rotations must rotate the object with a fixed amount at each key press. You should rotate the object anticlockwise around the X, Y or Z axis for the world coordinate system.

Scaling should change the size of the object by a fixed factor at each press. You should scale up the object along the X, Y or Z axis for the world coordinate system. This may move the object along the corresponding axis as well. [HINT: makes use of glPushMatrix() and glPopMatrix() as explained in class]	Scaling should change the size of the object by a fixed factor at each press. You should scale up the object along the X, Y or Z axis for the world coordinate system. This may move the object along the corresponding axis as well. [HINT: makes use of glPushMatrix() and glPopMatrix() as explained in class]
F. (5%) Write code to clear out all transformations and display current object as displayed originally.	**F. (5%)** Write code to clear out all transformations and display current object as displayed originally.
BONUS: (20%) Translate camera in the world coordinate system. Translations should move the camera by a fixed amount of distance at each press. You should move the camera to left, right, top, bottom, closer or further in the world coordinate system w.r.to its origin and axes. It should not change the field of view for the camera or near and far clipping planes for the camera.	**H. (15%)** Translate camera in the world coordinate system. Translations should move the camera by a fixed amount of distance at each press. You should move the camera to left, right, top, bottom, closer or further in the world coordinate system w.r.to its origin and axes. It should not change the field of view for the camera or near and far clipping planes for the camera.
BONUS: (20%) Write code to replace the display object to a cylinder or cone. Orient your object any way that you like. Make sure that the center of mass coincides with the origin for WCS.	**BONUS: (10%)** Write code to replace the display object to a cylinder or cone. Orient your object any way that you like. Make sure that the center of mass coincides with the origin for WCS.
BONUS: (10%) Change camera's field of view. At each pressing of the corresponding key, the angle for the vertical field of view must increase of decrease by a fixed amount.	**BONUS: (10%)** Change camera's field of view. At each pressing of the corresponding key, the angle for the vertical field of view must increase of decrease by a fixed amount.
BONUS: (20%) Write code to replace the display object to a sphere or torus. Orient your object any way that you like. Make sure	**BONUS: (10%)** Write code to replace the display object to a sphere or torus. Orient your object any way that you like. Make sure

that the center of mass coincides with the origin for WCS.	that the center of mass coincides with the origin for WCS.

WorkFlow:

- There is no base code package for this assignment. Write your code from scratch as explained at the beginning of this document.
- Your package should simply compile with the use of **make** command and produce an executable named **cg03.** Make sure you document clearly about the structure of your package and which folder contains the Makefile.
- You are organize your code in multiple files but you will have to modify the Makefile accordingly.
- You may use math.h for math operations. Although there should not be any need to use additional math packages such as vmath, GLM, e.t.c., if you choose to do so, it will be your responsibility that the source code for such a library is included in your code package and get compiled and linked to your code with the single use of **make** command as before.
- Prepare a PDF report explaining the structure of your code package and how to build and run it along with keyboard interfacing details. Especially, **include a version of Appendix A** from below that clearly associates a key input with corresponding desired operation.
- Include few screen-grabs for the output of your implementation for each of the task. You need not document each possible operation's output. Some representative cases would do.

- **Prepare and upload a single zip file as your submission package that should include:**
    - **The above report**
    - **All your source code files and the Makefile with a readme.txt if the need be.**

Reference material:

- Chapters 6 and 7 from the course book Fundamentals of Computer Graphics by Marschner and Shirley, 4th edition. You can access this book from O'Reilly's online resources.
- Class slides/videos.
- First few chapters for the book OpenGL Distilled by Paul Martz (available from O'Reilly's online resource link @ Clemson Library).
- Following links for legacy OpenGL, GLU and GLUT:
    - https://www.glprogramming.com/red/appendixb.html#name2
    - https://www.glprogramming.com/blue/ch03.html
    - https://www.opengl.org/resources/libraries/glut/spec3/spec3.html

[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 03: 3D OBJECT TRANSFORMATIONS

SUBMISSION DEADLINE: 11:59 p.m., 6th March (Friday) 2020.

- A delay of upto 24 hours will result in a proportional reduction of scores by 10%.
- A delay of upto 48 hours will result in a proportional reduction of scores by 20%.
- Any further delay will result in a proportional reduction of scores by 40%.

If there is a delay of for any medical exigency or any other unavoidable exigency, kindly contact me to avoid score reductions.

Upload your package over CANVAS

[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 03: 3D OBJECT TRANSFORMATIONS

APPENDIX A: Tabulation for ‘key’ associations with tasks

#	Operation	Detail	Shortcut-key/key-combination
1	Reset Display	Remove all transformations	
2	Translate	Left	
3	Translate	Right	
4	Translate	Up	
5	Translate	Down	
6	Translate	Closer to screen	
7	Translate	Away from screen	
8	Rotate	Around X	
9	Rotate	Around Y	
10	Rotate	Around Z	
11	Scale	Along X	
12	Scale	Along Y	
13	Scale	Along Z	
14	Toggle the view between orthographic and perspective projections		
15	Switch object (again) to	Cube	
16	Translate Camera	Left in WBS	
17	Translate Camera	Right	
18	Translate Camera	Up	
19	Translate Camera	Down	
20	Translate Camera	Along WCS Z axis, closer to the origin	
21	Translate Camera	Along WCS Z axis, away to the origin	
22	Switch object to	Torus	
23	Switch object to	Sphere	
24	Switch object to	Cone	
25	Switch object to	Cylinder	
26	Field of view	Increase	
27	Field of view	Decrease	