[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 04: 3D SURFACES AND SCENES

THIS IS A ONE WEEK ASSIGNMENT WITH SUBMISSION DATE AS 1st APRIL, 2020.

This assignment builds further on few of the concepts and algorithms assimilated through first three assignments. It will allow you to practice building and rendering of basic geometric surfaces and organize few of them into simple scene-graphs. The key ideas to learn here include the dynamics of managing geometric structures and transforming them in parts and as a whole. To this end, you will be using Legacy OpenGL owing to its simplicity and acquaintance from third assignment. More specifically, you will be get to develop a customizable surface of revolution that can be subject to transformations similar to those in third assignment. You CAN re-use your code and algorithms from your previous assignments when they directly contribute to sub-tasks for this assignment. For instance, your code for transforming the cube can be extended to transform the surface of revolution in 3D. This assignment will build further on the basic windowing functions, projective transformations and keyboard interfacing which will serve as the pre-requisite and not graded again.

IMPORTANT:

There is no base/starting code package for this assignment. You have to start from code for the third assignment or 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.). For details on using the Linux virtual desktop systems at School of Computing—Clemson, kindly go through the following webpage: https://www.cs.clemson.edu/help/remoteaccess.html. Furthermore, you will be able to this virtual desktop @SoC (virtual.computing.clemson.edu) in an off-campus manner without using Clemson VPN. After you enter your username and password, it'll prompt you for DUO authentication by push notification (option 1) or SMS one time code (option 2).

You may use code (including main.cpp) from any of the 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. This one object will either be your surface of revolution or the basic robot. You have to develop code to display it under perspective projection. Furthermore you have to develop code to move it in the world coordinate system interactively with the use of a keyboard.

PRE-REQUISITE TASK: Objectives based on RE-USE from your assignment 3:

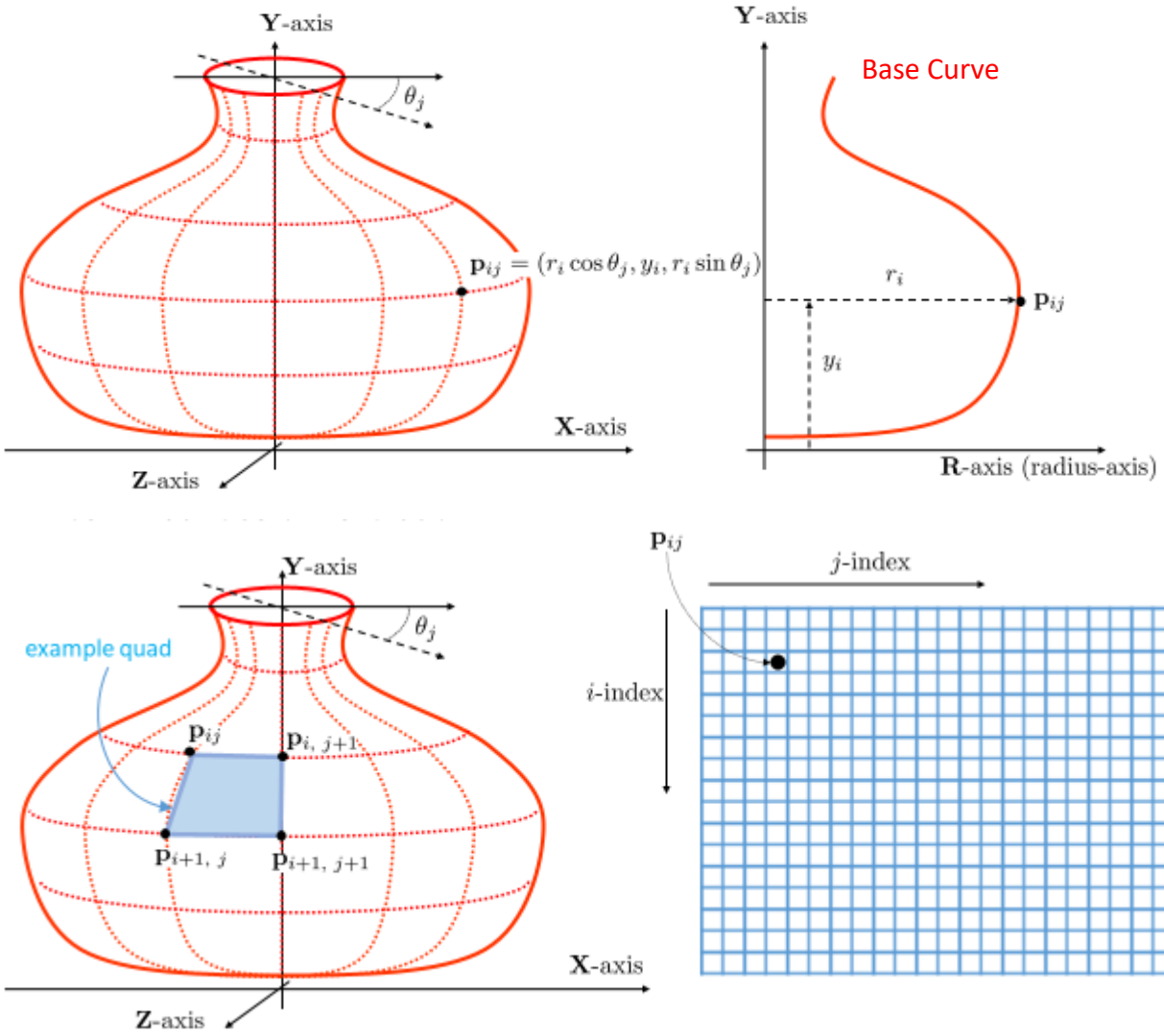
- Instantiate and display a Window of desired size using GLUT library.
- Place your 3D object/surface in your world coordinate system (WCS) and project it onto the screen for the window created earlier. Initially this object should be the surface of revolution.
- Fix the camera and projection to perspective projection as in your assignment 3.

[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 04: 3D SURFACES AND SCENES

- Provide keyboard interface to change the object being rendered
- Provide keyboard interface to reset all transformations

Objectives for TASK A: Constructing a surface of revolution.

You will be constructing a surface of resolution around the Y-Axis. Following diagram illustrates a surface of revolution and the key aspects for its construction. Kindly go through the class video/slides for 24th March, 2020 for further details.

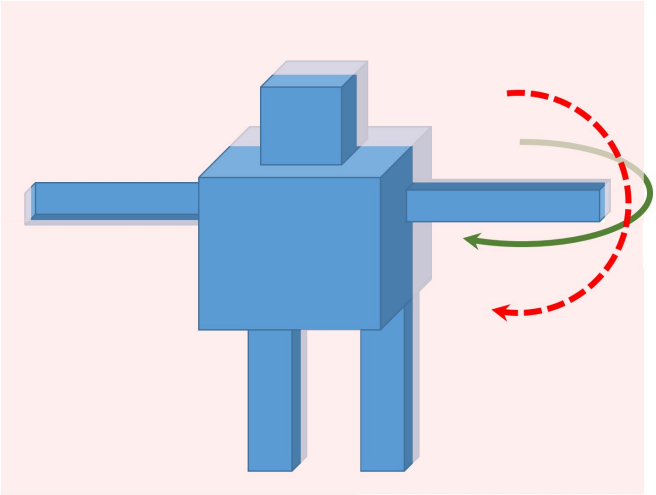


- Inputs: number of steps along y and θ dimensions, i.e. number of 'i' and 'j' indices respectively.
- Choice of Base Curves (4050): (a) parabolic arc, (b) circular arc, (c) Cubic Bezier curve
- Choice of Base Curves (6050): (a) Cubic Bezier curve, (b) Quintic Bezier curve, (c) B-Spline

[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 04: 3D SURFACES AND SCENES

- Write code to construct **ANY ONE** base 2D curve from above. For this base curve, one of the dimensions represents the radial-axis and the second dimension represents the y-axis for the world coordinate system.
- Your program should be able to switch between rendering of the surface of revolution and the basic robot from Task B.
- **Each time your program switches to drawing the surface of revolution,** intake main parameters corresponding to your curve of choice and the number of 'i' and 'j' indices, i.e. number of steps along the 'y' and the 'radial' axes respectively. [HINT: do not invoke user interactivity from within the display() function]
- With the input parameters from the above step, invoke your curve drawing algorithm to produce as many pairs of (y_i, r_i) as the given number of 'i' indices.
- Next, write code to generate the array of points p_{ij} as depicted in the illustrations above and explained in the class. With these points, generate a list of quads, where each quad connects 4 adjacent points as depicted above. This list of quad represents your surface of revolution.
- Write code to render above surface/object in your world-coordinate system.
- **[OPTIONAL/BONUS 4050/6050]** Apply translate and rotate operations from your Assignment 3 to transform this surface of revolution in much the same way as before.
- **[OPTIONAL/BONUS 6050]** Use mouse interactivity to move the control points for your Bezier or B-Spline curve. You can refer to mouse interactivity code from the base package for the second assignment. Note that for that code, the view is 2D and the origin for the mouse coordinates lie at the bottom left. [HINT: Assume that all control points lie in the XoY plane with z = 0]

Objectives for TASK B: Basic Scene-Graph with a Robot



[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 04: 3D SURFACES AND SCENES

You will be constructing a basic scene-graph consisting of a simple robot similar to the illustration above. The semitransparent plane in it represents the central XoY plane for the robot. The dotted red arrow represent an in-plane rotation for the robot's left arm. The solid green arrow represents the out-of-plane rotation for the same arm. You will be implementing these two rotations for the arm along with translation and rotation for the entire robot. Kindly go through the class video for 24th March, 2020 for further details.

- Write code to switch the object of display to the basic robot.
- Write code to draw this robot composed of a scene-graph with one torso, one head, two arms and two legs.
- Provide keyboard interactivity for the tasks below.
    - Rotate the left-arm (or the right if you wish so) clockwise and anticlockwise along an axis that passes through the point of contact between the arm and the torso and is perpendicular to the pink (semitransparent) plane above.
    - Rotate the same arm clockwise and anticlockwise along the same point of contact and in a manner that the rotation is along the axis lying in the pink plane and parallel to the vertical direction for the robot (i.e. parallel to its legs).
    - Translate and rotate the entire robot much in the same manner as for the cube in assignment #3.
- For 6050, the arms and legs must be cylindrical.
- [OPTIONAL for 6050] Divide the arm into fore-arm and the upper-arm and provide separate rotations for each of these parts at the point of contact.

Tasks and weightage in NUTSHELL:

BONUS scores will be clamped at 20%.

CPSC4050	CPSC6050
A. (40%) Construction and rendering of the surface of revolution	A. (40%) Construction and rendering of the surface of revolution
B. (20%) Robot Construction and basic rendering	B. (20%) Robot Construction and basic rendering.
C. (20%) Robotic arm rotations in its present location and orientation in the manner described above.	C. (20%) Robotic arm rotations in its present location and orientation in the manner described above.
D. (20%) Translations and rotations for the entire robot while preserving relative	D. (20%) Translations and rotations for the entire robot while preserving relative

[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 04: 3D SURFACES AND SCENES

orientation of its arms the same w.r.to its rest of the body.	orientation of its arms the same w.r.to its rest of the body.
BONUS: (10%) Apply translations and rotations to the surface of revolution in much the same way as described in assignment #3.	BONUS: (20%) Prove mouse interactivity for changing the control points for the base curve for the surface of revolution
BONUS: (20%) Use cylinders for arms and legs for the robot.	BONUS: (10%) Use sphere for the robot's head.
BONUS: (10%) Use sphere for the robot's head.	BONUS (20%) Divide the arm into a fore-arm and an upper arm and implement separate rotations for each of them at their point of contact.

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 **cg04.** 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:

[CPSC-4050/6050, SPRING 2020] ASSIGNMENT # 04: 3D SURFACES AND SCENES

- Chapters 6, 7 and 12 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

SUBMISSION DEADLINE: 11:59 p.m., 1st April (Wednesday) 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 any issue in accessing the Linux virtual desktop @ SoC, kindly get in touch with me at the earliest. You are advised to plan well and work in advance and not wait till the last hour to complete the assignment. Delay due to non-availability of the virtual computational resources will be considered in the same spirit and manner that would have happened on-site in the lab-room.

If there is a delay 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 # 04: 3D SURFACES AND SCENES

APPENDIX A: Tabulation for 'key' associations with tasks

#	Operation	Detail	Shortcut-key/key Combination
1	Reset Display	Remote all transformations	
2	Switch object to:	Surface of revolution	
3	Switch object to:	Robot	
4	Translate current object to	Left	
5	Translate current object to	Right	
6	Translate current object to	Above	
7	Translate current object to	Below	
8	Translate current object	Closer to the screen	
9	Translate current object to	Away from the screen	
10	Rotate entire object	Around X-Axis in WCS	
11	Rotate entire object	Around Y-Axis in WCS	
12	Rotate entire object	Around Z-Axis in WCS	
13	Rotate arm	Clockwise around the first axis explained above	
14	Rotate arm	Anti-clockwise around the first axis explained above	
15	Rotate arm	Clockwise around the second axis explained above	
16	Rotate arm	Anti-clockwise around the second axis explained above	
17	Rotate forearm	Clockwise around the first axis explained above	
18	Rotate forearm	Anti-clockwise around the first axis explained above	
19	Rotate forearm	Clockwise around the second axis explained above	
20	Rotate forearm	Anti-clockwise around the second axis explained above	