Laboratory Practical 3

3D GRAPHICS

Important Information

The maximum mark for this laboratory practical is 10. The lab practical will be completed in three two-hour sessions according to the schedule given in the Learning Guide of the unit. Students need to submit a report for the practical. The requirements for report submission are summarised below.

Format	Where	When
1. Submit an electronic copy of your report. 2. Submit an electronic copy of your **complete C++ source code**. Please delete the 'Debug' folder, and then compress your **whole program** using WinZip to reduce the file size (Right click on the folder → Send to → Compressed (zipped) folder). Please DO NOT use 'RAR' to compress your files. Your source code should compile and run under the Visual Studio 2019 development environment. **If you do not submit your source code or your source code does not compile/run, your report may not be marked and you may be given a zero mark for this lab practical.**	Submission through the 'Assessment Zone' of the unit's vUWS site (Assessment Zone → Assessment 1 (30%) → Lab 3 → 'Browse My Computer' to attach your report and your source code → Submit).	check submission link on vUWS

Table of Contents

I. Aim

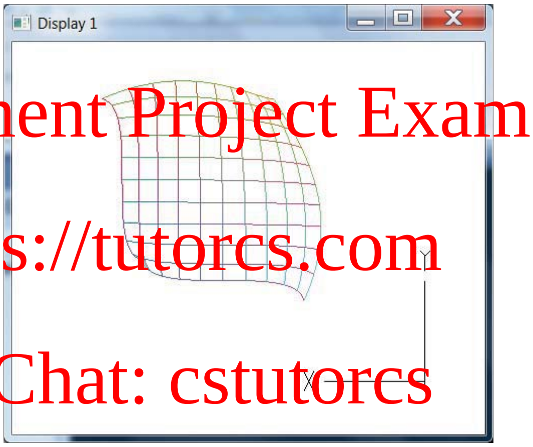
Real-world objects are usually 3D objects. On the other hand, a display device often has a 2D screen. To display a 3D object on a 2D screen, a transformation is needed to project the object onto the screen. This laboratory practical aims to draw 3D objects and project them to a plane so that they can be viewed on a display device. Specifically, you will draw a Bezier surface and a solid generated by a rotational sweep. You will also map texture images to surfaces.

II. Tasks

Run the C++ sample program 'Lab_3_Shell' available from vUWS and use the examples given in the program to complete the following tasks.

1. Draw Bezier surface patches.

   a. Draw bicubic Bezier surface patches using an orthographic parallel projection. As an example, Display_1 of Lab_3_Shell draws a bicubic Bezier surface patch as shown below.



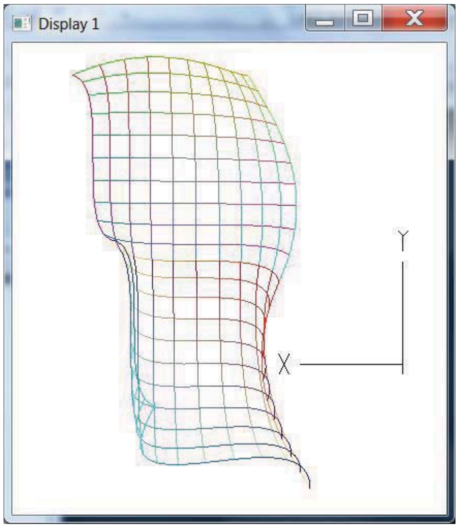
The surface patch is specified by 16 (4×4) control points, $\mathbf{p}_{ij} = (x_{ij}, y_{ij}, z_{ij})$ for $i, j = 0, 1, 2, 3,$ where

$\mathbf{p}_{00}$ = (120, 80, 200), $\mathbf{p}_{01}$ = (80, 170, 100), $\mathbf{p}_{02}$ = (120, 230, 100), $\mathbf{p}_{03}$ = (150, 280, 100),
$\mathbf{p}_{10}$ = (130, 120, -100), $\mathbf{p}_{11}$ = (170, 130, 100), $\mathbf{p}_{12}$ = (130, 270, 100), $\mathbf{p}_{13}$ = (170, 280, -100),
$\mathbf{p}_{20}$ = (270, 80, -100), $\mathbf{p}_{21}$ = (230, 170, 100), $\mathbf{p}_{22}$ = (270, 230, 100), $\mathbf{p}_{23}$ = (230, 320, -100),
$\mathbf{p}_{30}$ = (280, 120, 200), $\mathbf{p}_{31}$ = (320, 130, 100), $\mathbf{p}_{32}$ = (280, 270, 100), $\mathbf{p}_{33}$ = (320, 280, 100).

In this case, an orthographic parallel projection is used to display the surface by setting *viewing = 3 in the function Display_1::Draw(…). The view point (or eye position) is (0, 0, –600). The reference point (or look-at point) is (0, 0, 0). The view-up vector is (0, 1, 0).

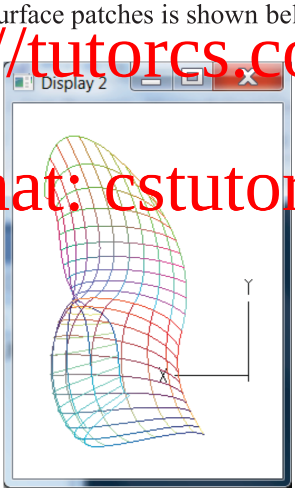
Using the same orthographic parallel projection, your task is to draw another bicubic Bezier surface patch that shares one edge of the bicubic Bezier surface patch mentioned above. Use different colours to show different parts of your Bezier surface to enhance the viewing effect. Place your drawing code in the drawing function of Display_1 (Display_1::Draw(...)) in the sample program 'Lab_3_Shell'. Both Bezier surface patches should show when the key '1' of the keyboard is pressed.

An example of drawing the Bezier surface patches is shown below.



b. Draw the above bicubic Bezier surface patches using a perspective projection. The projection reference point (or centre of the projection), which is also the view point (or eye position), is located at (0, 0, –600) in the world coordinate system. The reference point (or look-at point) is (0, 0, 0). The view-up vector is (0, 1, 0). Use different colours to show different parts of the Bezier surface to enhance viewing effect. Place your drawing code in the drawing function of Display_2 (Display_2::Draw(...)) in the sample program 'Lab_3_Shell'. Both Bezier surface patches should show when the key '2' on the keyboard is pressed.

An example of drawing the Bezier surface patches is shown below.



Hints:

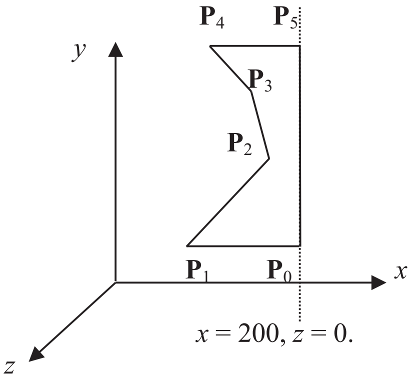
- You should select 3D perspective projection (*viewing = 4) in the sample program in this case.
- The view plane, which is also the near clipping plane, is 400 from the eye position. This has already been set up in the sample program 'Lab_3_Shell'.

Listed below are the requirements for Task 1 (consisting of 1a and 1b), which should be addressed in your report.

i. Explain what an orthographic parallel projection is with the aid of relevant illustrations.
ii. Explain what a perspective projection is with the aid of relevant illustrations.
iii. Describe a bicubic Bezier surface in mathematical equations and explain how the equations are used to display the surface.
iv. Explain the C++ source code you have written for this task so that an experienced reader can repeat the task to obtain the same result without the need of your C++ code. You may use a sequence of steps, a flow chart or pseudo-code for your explanation. Do not include your C++ source code in your report.
v. Include the drawing result of the task in your report.

2. Draw a solid by rotational sweep.

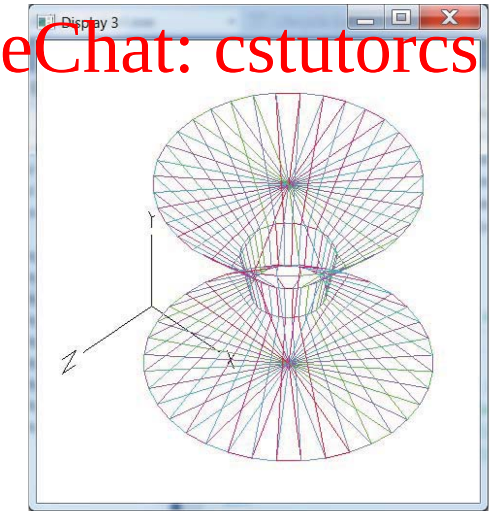
   a. Draw the surface of a solid generated by rotating the following polygon $\mathbf{P}_0\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3\mathbf{P}_4\mathbf{P}_5$ through 360° about an axis defined by $x = 200$, $z = 0$.



$\mathbf{P}_0 = (200, 50, 0)$, $\mathbf{P}_1 = (50, 50, 0)$,
$\mathbf{P}_2 = (160, 150, 0)$, $\mathbf{P}_3 = (150, 200, 0)$,
$\mathbf{P}_4 = (60, 300, 0)$, $\mathbf{P}_5 = (200, 300, 0)$.

Use an orthographic parallel projection to display the surface of the solid. The view point (or eye position) is (400, 310, 200). The reference point (or look-at point) is (200, 50, 0). The view-up vector is (0, 1, 0). One way to show the surface is to rotate a polyline defined by $\mathbf{P}_0$, $\mathbf{P}_1$, …, $\mathbf{P}_5$ in steps (e.g., 10-degree difference per step) to generate a set of polygons. Then, the surface is approximated by a wire-frame formed by the polygons. Use different colours to show different polygons to enhance the viewing effect. Place your drawing code in the drawing function of Display_3 (Display_3::Draw(...)) in the sample program 'Lab_3_Shell'. Your drawing results should show when the key '3' on the keyboard is pressed.

An example of the above surface is shown below.

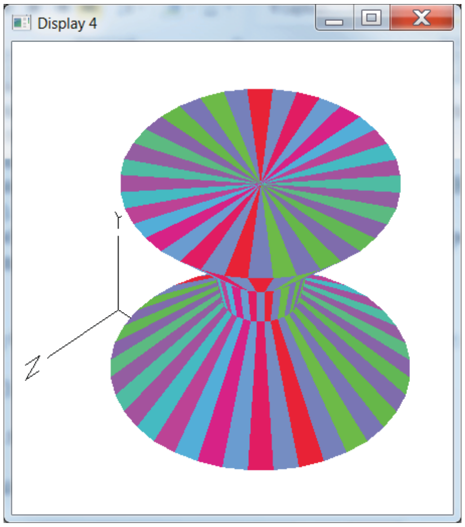


Hints:

- An example of drawing a partial surface is given in Display_3 of Lab_3_Shell after the polyline is rotated by 10 degrees.

   b. Draw the visible surface of the above solid. The visible surface of the solid can be determined using the z-buffer/depth-buffer algorithm. Place your drawing code in the drawing function of Display_4 (Display_4::Draw(...)) in the sample program 'Lab_3_Shell'. Your drawing results should show when the key '4' on the keyboard is pressed.

   An example of the above surface is shown below.



Hints:

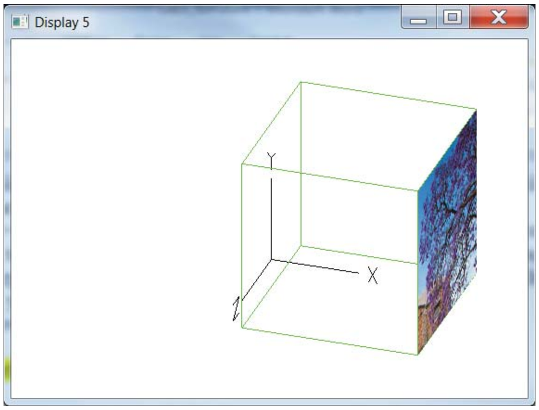
- An example of drawing a partial surface is given in Display_4 of Lab_4_Shell after the polyline is rotated by 10 degrees.

Listed below are the requirements for Task 2 (consisting ‘2a’ and ‘2b’), which should be addressed in your report.

i. Explain how the vertices of the polygons approximating the above surface are determined with the aid of relevant mathematical equations.
ii. Explain what the z-buffer/depth-buffer algorithm is and how it is used to determine a visible surface with the aid of relevant illustrations.
iii. Explain the C++ source code you have written for this task so that an experienced reader can repeat the task to obtain the same result without the need of your C++ code. You may use a sequence of steps, a flow chart or pseudo-code for your explanation. Do not include your C++ source code in your report.
iv. Include the drawing result of the task in your report.

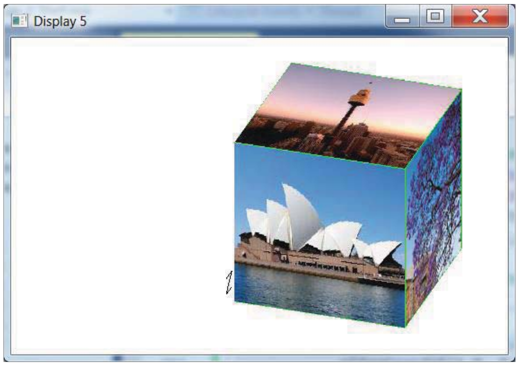
3. Map texture images to surfaces.

Display_5 of the sample program ‘Lab_3_Shell’ shows an example of mapping a texture image to one of the surfaces of a cube as shown below.



Use this example to map a texture image of your choice to the front surface of the cube and map another texture image of your choice to the top surface. Place your drawing code in the drawing function of Display_5 (Display_5::Draw(...)) in the sample program ‘Lab_3_Shell’. The texture mapping result using three images should show when the key ‘5’ on the keyboard is pressed.

An example of the drawing result is shown below.



Listed below are the requirements for Task 3, which should be addressed in your report.

i. Explain how a texture image is mapped to a surface with the aid of relevant mathematical equations.
ii. Explain the C++ source code you have written for this task so that an experienced reader can repeat the task to obtain the same result without the need of your C++ code. You may use a sequence of steps, a flow chart or pseudo-code for your explanation. Do not include your C++ source code in your report.
iii. Include the drawing result of the task in your report.

4. Perform a task of your choice, which can be used to distinguish you from others. This is a required task and it must be directly related to this unit of study. Use Display_9 of the sample program 'Lab_3_Shell' to show the result of this task. Examples of the task are given below.

Example 1: You find a problem during this lab practical. You decide to tackle the problem. You propose a method to solve the problem and implement the method using C++. You show results to verify your proposal.
Example 2: You find an interesting algorithm in the textbook, which has not been (and will not be) discussed in this unit of study. You decide to implement the algorithm in C++. You show results to verify your implementation.
Example 3: You find an interesting algorithm in visualisation in an IEEE journal. You decide to implement the algorithm in C++. You show results to verify your implementation.

The requirements for this task are listed below.

i. Please respond to the following questions in your report.
   - What is the aim of the task? In other words, what do you want to do?
   - Why do you want to do it?
   - What are the challenges of this task when compared to the tasks above?
   - How do you do it? In other words, can you explain your method or implementation?
   - What are your results? Can you explain them?
   - What can you conclude from the results?
   - Have you achieved your aim?
   - Is your conclusion significant and why?

III. Lab Report

You need to write a report for this lab practical. You may exchange ideas and techniques with others. However, you must write your report independently. The requirements for your lab report are listed below.

i. Attach an assignment cover sheet that you have completed for this lab practical.
ii. State the aim of the lab practical.
iii. Include your response to the requirements for each task. Your report should be divided into sections such that each task is discussed individually in one section.
iv. Include the conclusion of this lab practical.

V. Marking Criteria and Standards

Tasks/Criteria	Standards	Raw Marks Awarded
1. Drawing Bezier surface patches (Maximum 2 marks)	100% of the requirements for the task are addressed with correct answers.	2
	At least 70% of the requirements for the task are addressed with correct answers.	1
	Less than 70% of the requirements for the task are addressed with correct answers.	0
2. Drawing surface of a solid by rotational sweep (Maximum 3 marks)	100% of the requirements for this task are addressed with correct answers.	3
	At least 90% of the requirements for this task are addressed with correct answers.	2
	At least 50% of the requirements for this task are addressed with correct answers.	1
	Less than 50% of the requirements for this task are addressed with correct answers.	0
3. Mapping texture images to surfaces (Maximum 2 marks)	100% of the requirements for the task are addressed with correct answers.	2
	At least 70% of the requirements for the task are addressed with correct answers.	1
	Less than 70% of the requirements for the task are addressed with correct answers.	0
4. Self-selected task. (Maximum 3 marks)	The aim is innovative and significant. The task is very challenging. Extensive observations and connections are made. Extensive results are presented and explained. The conclusion is significant and will be appreciated by practitioners in the field.	2 – 3
	The aim and rationale are stated. The task is challenging when compared to the set tasks. The method and implementation are described. Results are presented and explained. The conclusion is stated. The student makes observations and connections regarding the underlying task. The aim is innovative and it is achieved.	1
	The aim and rationale are not clearly stated. The task is not challenging when compared to the set tasks. The task may be challenging, but the student has no or almost no idea on how to perform the task. No method is presented or the method is not clearly described. No implementation is given or the implementation is not clearly described. A similar method has been used for the set tasks. There are no or very limited results. There is no or very limited explanation/discussion of the results. No conclusion is stated or it is not clear what the conclusion is. The conclusion is not supported by the results. The aim is not achieved. The conclusion is insignificant.	0
Highest possible score for this lab practical		10