COMPUTER GRAPHICS (CCG3013)

LESSON 5

GEOMETRY IN 3D GRAPHICS: PART II

UOW MALAYSIA
KDU PENANG
UNIVERSITY COLLEGE

PART OF THE UNIVERSITY
OF WOLLONGONG AUSTRALIA
GLOBAL NETWORK

UNIVERSITY OF
LINCOLN
UNITED KINGDOM

COURSE OUTLINE

Lesson	Topic
1	Introduction to computer graphics
2	Graphics hardware and software
3	Geometry in 2D graphics
4 & 5	Geometry in 3D graphics
6 & 7	User interfaces and interactions
8	Colour
9 & 10	Motion and animation
11	Lighting and rendering
12	Surface shadings

LEARNING OUTCOMES

1. Describe and illustrate three-dimensional (3D) models in 3D space.

2. Compute matrix transformations in 3D space.

3. Explain and implement 3D drawing functions.

ASSESSMENTS

Structure	Marks (%)	Hand-out	Hand-in
Assignment 1 (Individual)	30	Week 1(Unofficial) Week 3(Official)	Week 6
Assignment 2 (Group up to four only)	30	Week 1(Unofficial) Week 3(Official)	Week 12
Final examination	40	Exam week	

CONTENT

No.	Topics	Duration (Minutes)
1	Mini lecture 1: 3D models	15
2	Exercise 1	10
3	Mini lecture 2: 3D modeling techniques	15
4	Exercise 2	10
5	Break	10
6	Mini lecture 3: 3D faces	15
7	Exercise 3	10
8	Mini lecture 4: 3D render functions	15
9	Exercise 4	10

REVIEW: 3D TO 2D PROJECTION

3D point: (4, 5, 6), scaling factors: (1, 1), and offsets: (0, 0)

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & 0 & s_y \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} c_x \\ c_z \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \\ 6 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \\ 6 \end{bmatrix}$$

REVIEW: 3D TO 2D PROJECTION

3D point: (72, 53, 64), scaling factors: (2, 2), and offsets: (10, 10).

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & 0 & s_y \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} c_x \\ c_z \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 72 \\ 53 \\ 64 \end{bmatrix} + \begin{bmatrix} 10 \\ 10 \end{bmatrix}$$

$$= \begin{bmatrix} 144 \\ 128 \end{bmatrix} + \begin{bmatrix} 10 \\ 10 \end{bmatrix}$$

$$= \begin{bmatrix} 154 \\ 138 \end{bmatrix}$$

REVIEW: 3D TO 2D PROJECTION

3D point: (28, 30, 40), scaling factors: (1/2, 1/2), and offsets: (1, 15).

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & 0 & s_y \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} c_x \\ c_z \end{bmatrix}$$

$$= \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} 28 \\ 30 \\ 40 \end{bmatrix} + \begin{bmatrix} 1 \\ 15 \end{bmatrix}$$

$$= \begin{bmatrix} 14 \\ 20 \end{bmatrix} + \begin{bmatrix} 1 \\ 15 \end{bmatrix}$$

$$= \begin{bmatrix} 15 \\ 35 \end{bmatrix}$$

REVIEW: 3D TO 2D PROJECTION

3D point: (16, 16, 16), scaling factors: (1, 1), and offsets: (5, 5).

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & 0 & s_y \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} c_x \\ c_z \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 16 \\ 16 \\ 16 \end{bmatrix} + \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$= \begin{bmatrix} 16 \\ 16 \end{bmatrix} + \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

$$= \begin{bmatrix} 21 \\ 21 \end{bmatrix}$$

REVIEW: 3D TO 2D PROJECTION

3D point: (32, 32, 64), scaling factors: (1/4, 1/4), and offsets: (20, 25).

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \begin{bmatrix} s & 0 & 0 \\ 0 & 0 & s_y \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} + \begin{bmatrix} c_x \\ c_z \end{bmatrix}$$

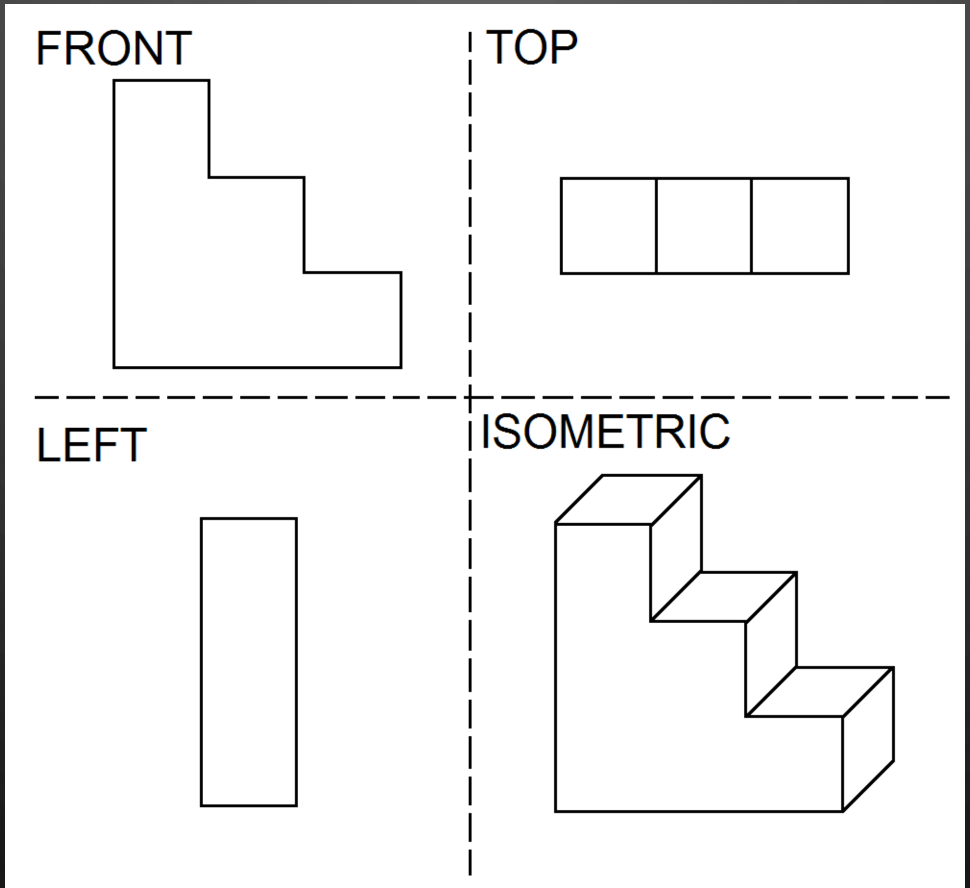
$$= \begin{bmatrix} 1/4 & 0 & 0 \\ 0 & 0 & 1/4 \end{bmatrix} \begin{bmatrix} 32 \\ 32 \\ 64 \end{bmatrix} + \begin{bmatrix} 20 \\ 25 \end{bmatrix}$$

$$= \begin{bmatrix} 8 \\ 16 \end{bmatrix} + \begin{bmatrix} 20 \\ 25 \end{bmatrix}$$

$$= \begin{bmatrix} 28 \\ 41 \end{bmatrix}$$

REVIEW: ORTHOGRAPHIC VIEWS

A standard orthographic views for the following 3D models.



REVIEW: 3D TRANSLATION

Given a 3D original point at (30, 20, 15). Translate with a vector of (-5, -10, 12).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -10 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 1 \end{bmatrix} = \begin{bmatrix} 30-5 \\ 20-10 \\ 15+12 \\ 1 \end{bmatrix} = \begin{bmatrix} 25 \\ 10 \\ 27 \\ 1 \end{bmatrix}$$

REVIEW: 3D CLOCKWISE ROTATION

Given a 3D original point at (30, 20, 15). Rotate clockwise (CW) at 45 degrees along y-axis.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos 45 & 0 & \sin 45 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin 45 & 0 & \cos 45 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0 & 0.7071 & 0 \\ 0 & 1 & 0 & 0 \\ -0.7071 & 0 & 0.7071 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 21.213 + 10.6065 \\ 20 \\ -21.213 + 10.6065 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 31.8195 \\ 20 \\ -10.6065 \\ 0 \end{bmatrix}$$

REVIEW: 3D COUNTER-CLOCKWISE ROTATION

Given a 3D original point at (30, 20, 15). Rotate counter-clockwise (CCW) at 90 degrees along z-axis.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} \cos 90 & -\sin 90 & 0 & 0 \\ \sin 90 & \cos 90 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -20 \\ 30 \\ 15 \\ 0 \end{bmatrix}$$

EXERCISE 2 ANSWER IV

2. Refer Exercise 2, Lecture 04, slide number 32. 3D transformations, given a 3D original point at (30, 20, 15).

d. Scale with a factor of (2, 3, 1).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 0 \end{bmatrix} = \begin{bmatrix} 60 \\ 60 \\ 15 \\ 0 \end{bmatrix}$$

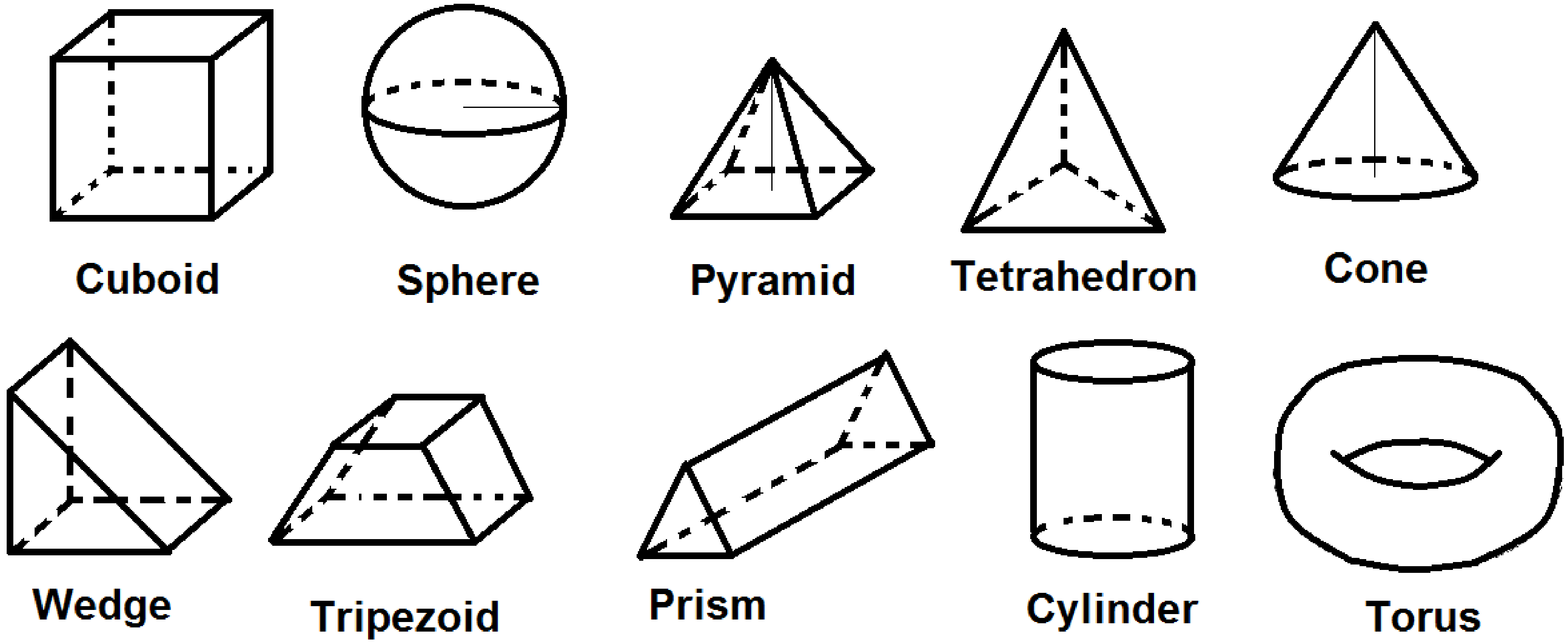
MINI LECTURE 1
3D MODELS

...

BRAINSTORM FOR 3D PRIMITIVE MODELS

What are the 3D primitive models that you can identified?

3D PRIMITIVES MODELS



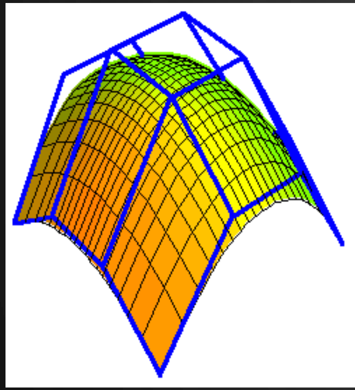
3D MODELING

Approach	Explanation
Surface reconstruction	It fits 3D point cloud into UV polygonal equations.
Mesh generation	It generates a set of vertices with triangles or quadrilaterals.
Cube generation	It forms voxels using binary space partitions (BSP).

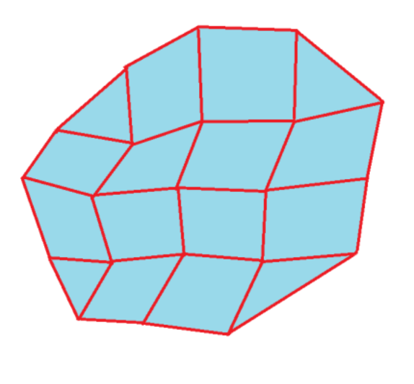
3D MODELING TECHNIQUES

Approach	Explanation
Surface reconstruction	Bezier surfaces; b-Spline surfaces; Coons (Citroën); NURBS;
Mesh generation	Triangulation; Grid;
Cube generation	Voxel; Octree;

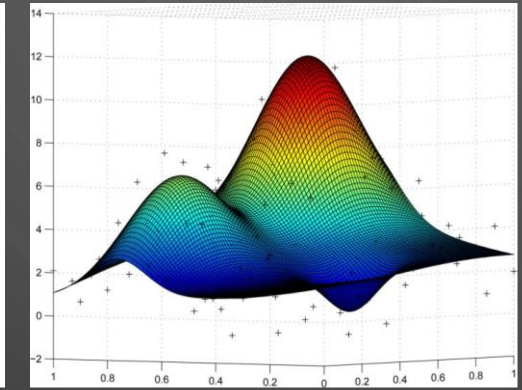
3D MODELING TECHNIQUES: EXAMPLES



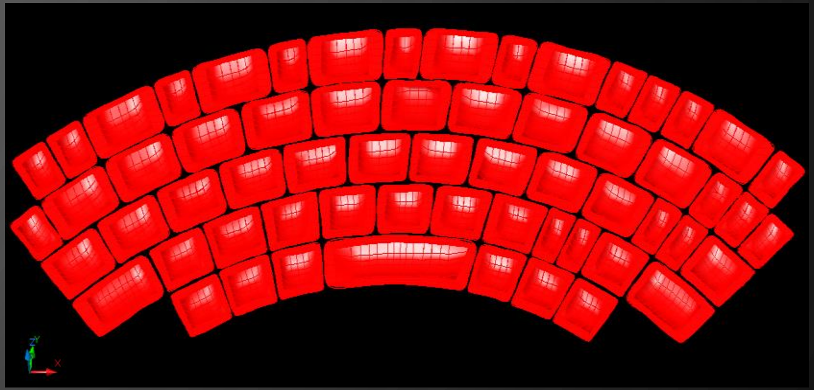
(a)



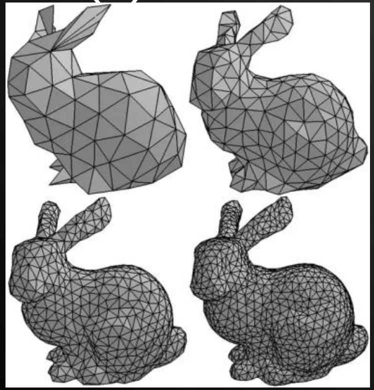
(b)



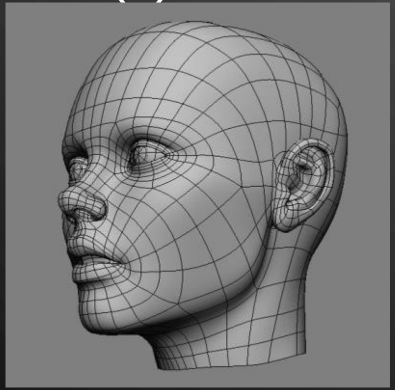
(c)



(d)

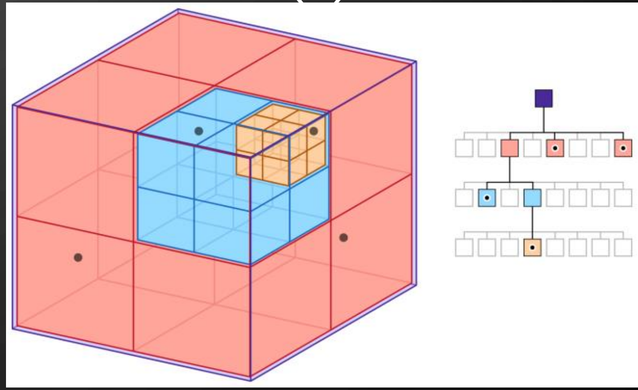


(e)



(f)

(g)



(h)

EXERCISE 1

This activity will takes about 10 minutes.

Identify a 3D model, then specify 3D primitives and corresponding quantities that can be used to represent the model in a table.

Petronas Twin Tower

3D primitives	Quantity
Cylinder	2
Cuboid	1
Trapezoid	2

Victorian house

3D primitives	Quantity
Wedge	3
Cuboid	1+6
Cone	1 (low poly cone)

MINI LECTURE 2
3D MODELING TECHNIQUES

...

SELECTIONS OF 3D OBJECTS

1. Vertex, it is a point or the smallest unit in a 3D model.

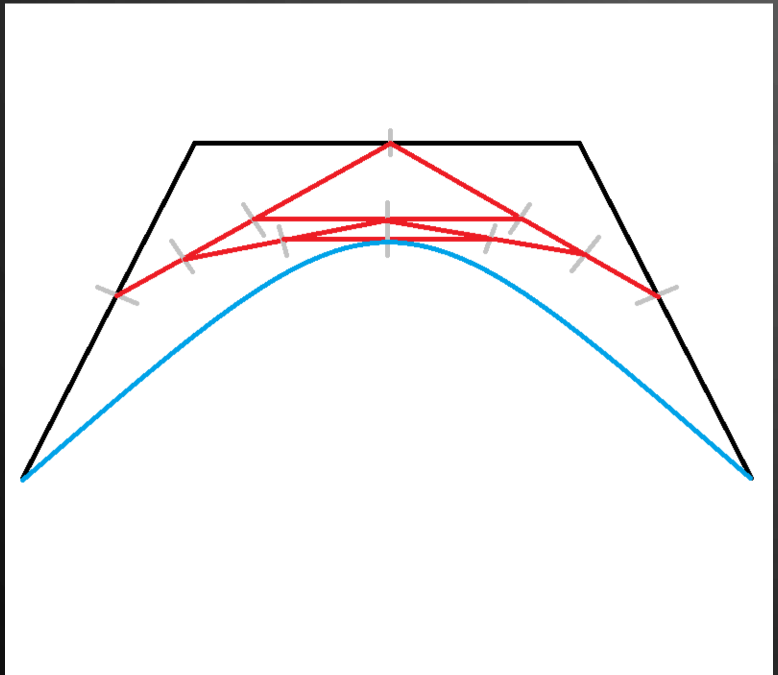
2. Edge, it is a line that connects two vertices.

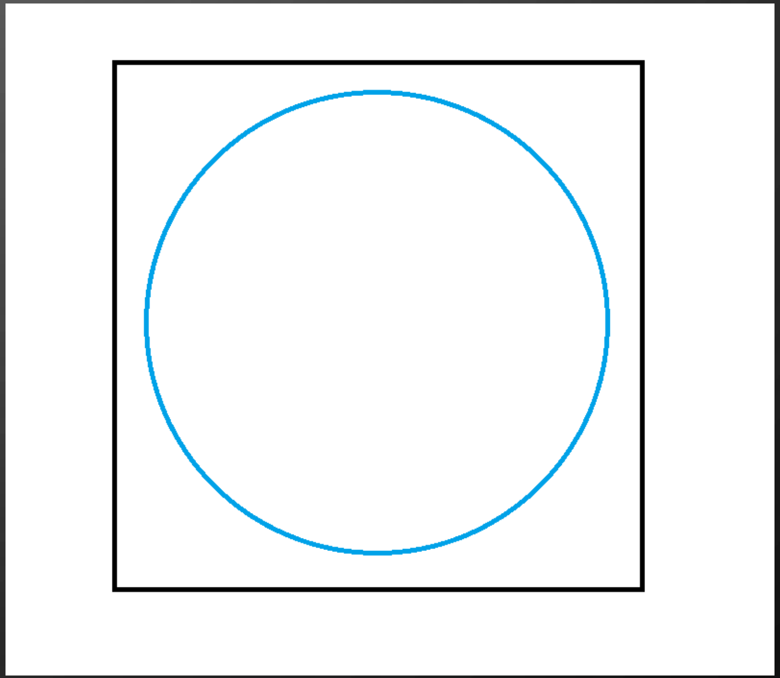
3. Face, it is a 2D shape or polygon which is formed by the edges.

4. Control polygon, it is a constraint for a polynomial interpolation, which consists of vertices and edges.

BEZIER CURVE

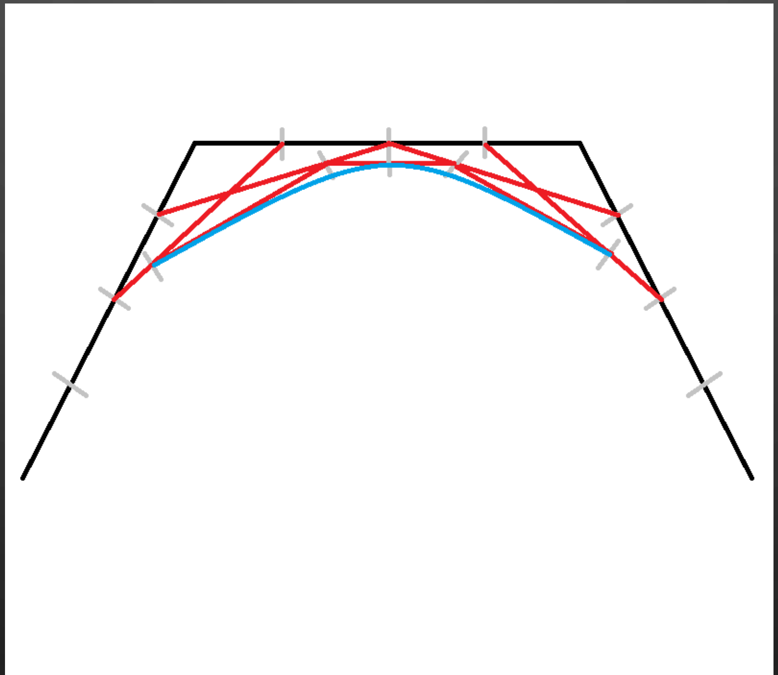
It subdivides a control polygon using the mid-points of edges.





B-SPLINE CURVE

It subdivides a control polygon using the quarter-points of edges.



EXERCISE 2

Write a function in C++ OpenGL to generate a quadratic curve in OpenGL window.

BREAK

…

MINI LECTURE 3
3D FACES

…

FACES

1. There are two sides for a face, which are front face and back face.

2. Two sides of a face is rendered by default in OpenGL.

3. Usually, either front face or back face will be rendered to save GPU memories, unless the interior of a 3D model is required to be displayed.

FRONT OR BACK FACE

Function name	glFrontFace
Purpose	To set front face or back face for rendering to reduce computation.
Arguments or parameters	GL_CW, set front face for vertices specified in clockwise, or GL_CCW, set front face for vertices specified in counter-clockwise.
Return value	None

FACE CULLING

Function name	glCullFace
Purpose	To remove front face or back face from rendering.
Arguments or parameters	GL_FRONT, set front face for culling, or GL_BACK, set back face for culling, or GL_FRONT_AND_BACK, set both front and back for culling.
Return value	None

CULLING

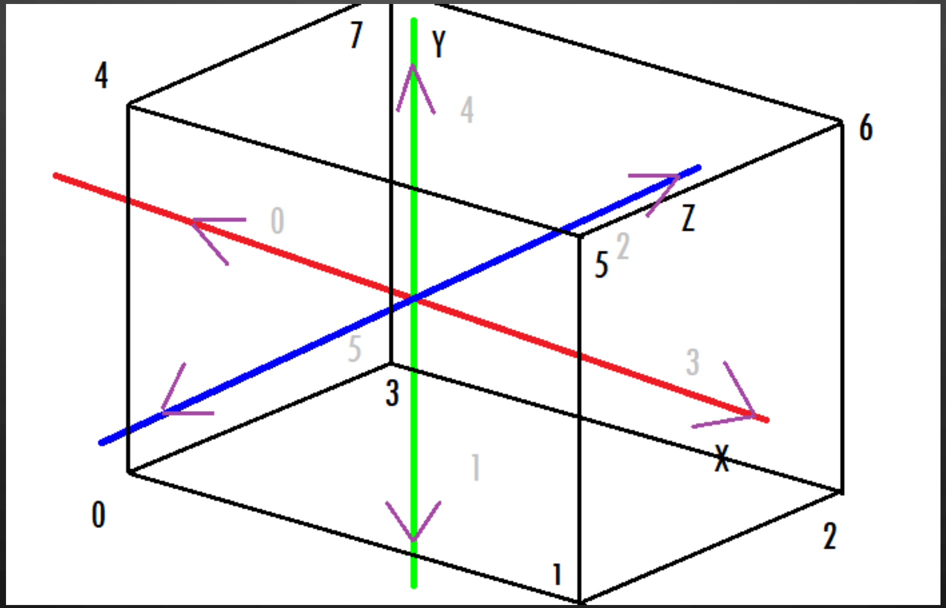
1. To enable the culling of faces, glEnable(GL_CULL_FACE)

2. To disable the culling of faces, glDisable(GL_CULL_FACE)

NORMAL SURFACE

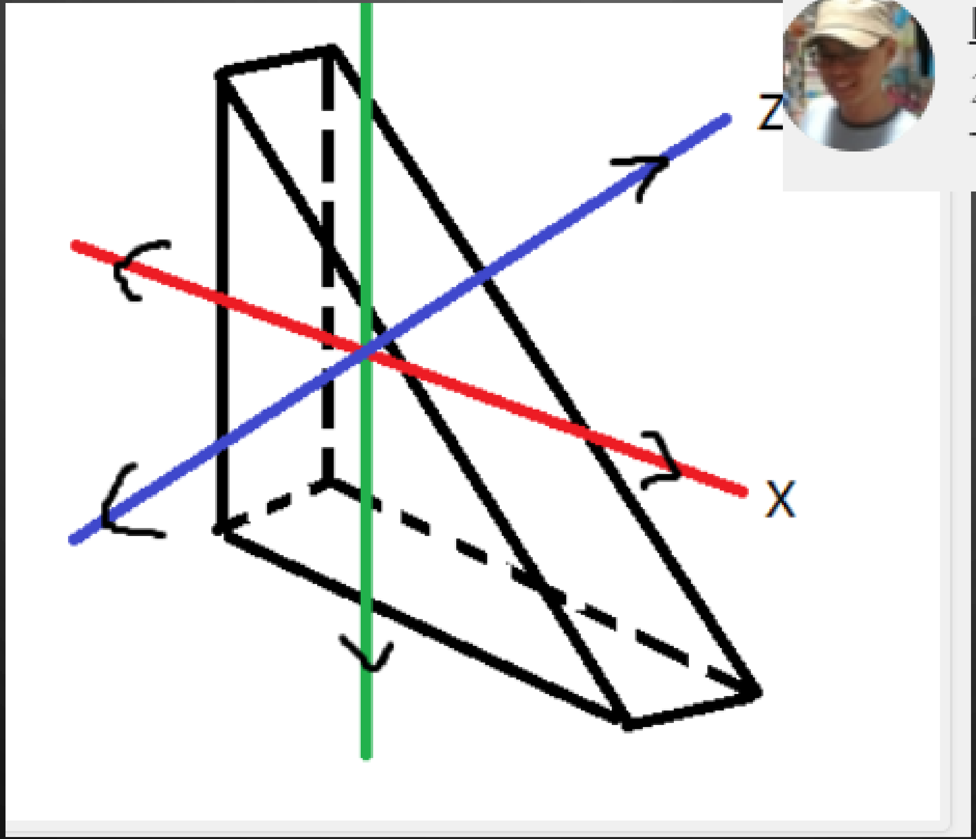
1. A direction where it responses to one or more light sources.

2. The normal direction is always perpendicular to a face.



EXERCISE 3

Pick one of the 3D primitive models, then illustrate the normal direction for each surface.



Khoo Hee Kooi

2 minutes ago

There are only five normal directions.

MINI LECTURE 4
3D DRAWING FUNCTIONS

…

TEAPOT

Function name	glutSolidTeapot, glutWireTeapot
Purpose	Render a solid or wireframe teapot.
Arguments or parameters	Size, relative size of the teapot.
Return value	None

SPHERE

Function name	glutSolidSphere, glutWireSphere
Purpose	Render a solid or wireframe sphere.
Arguments or parameters	Radius, radius of sphere; Slices, number of subdivisions in vertical lines. Stacks, number of subdivisions in horizontal lines.
Return value	None

EXERCISE 4

Write a function in C++ OpenGL to render a solid sphere and a wire sphere in OpenGL frustum.

REFERENCES

Main reference:

Hajek, D. (2019). Introduction to Computer Graphics 2019 Edition. Independently Published.

Additional reference:

Marschner, S. and Shirley, P. (2021). Fundamentals of Computer Graphics, 5th Edn. CRC Press: Taylor's & Francis.