COMPUTER GRAPHICS (CCG3013)

LESSON 4

GEOMETRY IN 3D GRAPHICS: PART I

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

TOPIC LEARNING OUTCOMES

1. Illustrate three-dimensional (3D) environment on 2D screen.

2. Explain and implement perspective view with matrix stacks.

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	

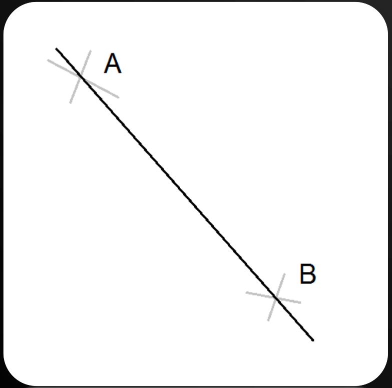
CONTENT

No.	Topics	Duration (Minutes)
1	Mini lecture 1: 3D coordinates system & 3D views	15
2	Exercise 1	10
3	Mini lecture 2: Matrix transformations in 3D space	15
4	Exercise 2	10
5	Break	10
6	Mini lecture 3: Matrix stacks & its operations	15
7	Exercise 3	10
8	Mini lecture 4: Configure a 3D space	15
9	Exercise 4	10

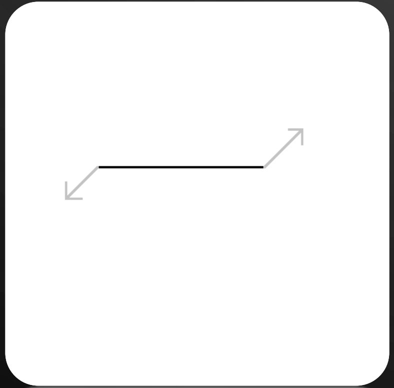
REVIEW I: EUCLID'S POSTULATE

Definition and illustration of five Euclid's postulate.

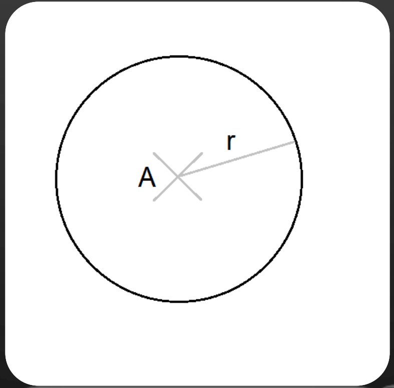
1. Given two points, it is possible to draw a right line.
2. The right line can be extended in both directions.
3. Given a center and a radius, we can draw a circle.
4. All right angles are equal.
5. Parallel postulate.



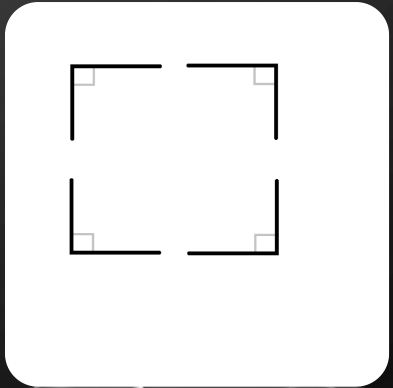
Postulate a



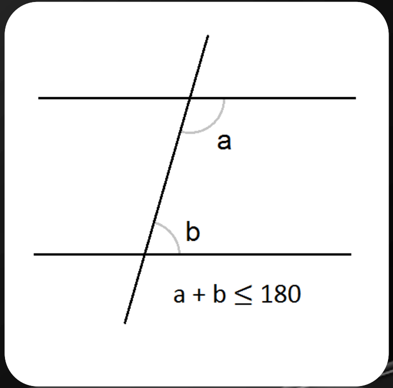
Postulate b



Postulate c



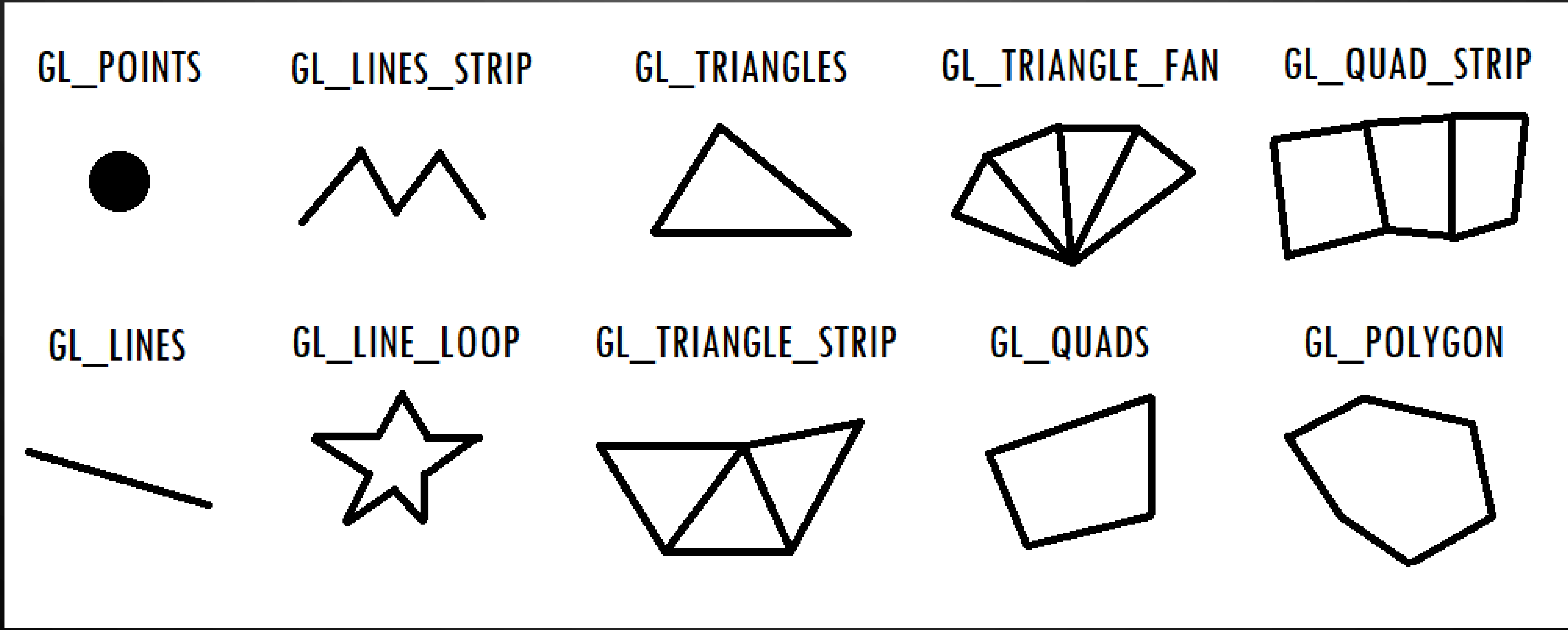
Postulate d



Postulate e

REVIEW II: DRAWING MODES IN OPENGL

Illustration of ten drawing modes in OpenGL.



REVIEW III: 2D TRANSLATE

Original point, P = (15, 25).

Translate with a vector of (t_x, t_y) = (20, -47).

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 20 \\ 0 & 1 & -47 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 15 \\ 25 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 15+20 \\ 25-47 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 35 \\ -22 \\ 1 \end{bmatrix}$$

REVIEW III: 2D CLOCKWISE ROTATION

Original point, P = (15, 25).

Rotate clockwise (CW) at 30 degrees.

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

$$= \begin{bmatrix} \cos 30 & \sin 30 & 0 \\ -\sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 15 \\ 25 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 15\cos 30 + 25\sin 30 \\ -15\sin 30 + 25\cos 30 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 25.49 \\ 14.15 \\ 0 \end{bmatrix}$$

REVIEW III: 2D COUNTER-CLOCKWISE ROTATION

Original point, P = (15, 25).

Rotate counter-clockwise (CCW) at 90 degrees.

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

$$= \begin{bmatrix} \cos 90 & -\sin 90 & 0 \\ \sin 90 & \cos 90 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 15 \\ 25 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 15\cos 90 - 25\sin 90 \\ 15\sin 90 + 25\cos 90 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} -25 \\ 15 \\ 0 \end{bmatrix}$$

REVIEW III: 2D SCALING

Original point, P = (15, 25).

Scale with a factor of $(s_x, s_y) = (0.5, 2.5)$.

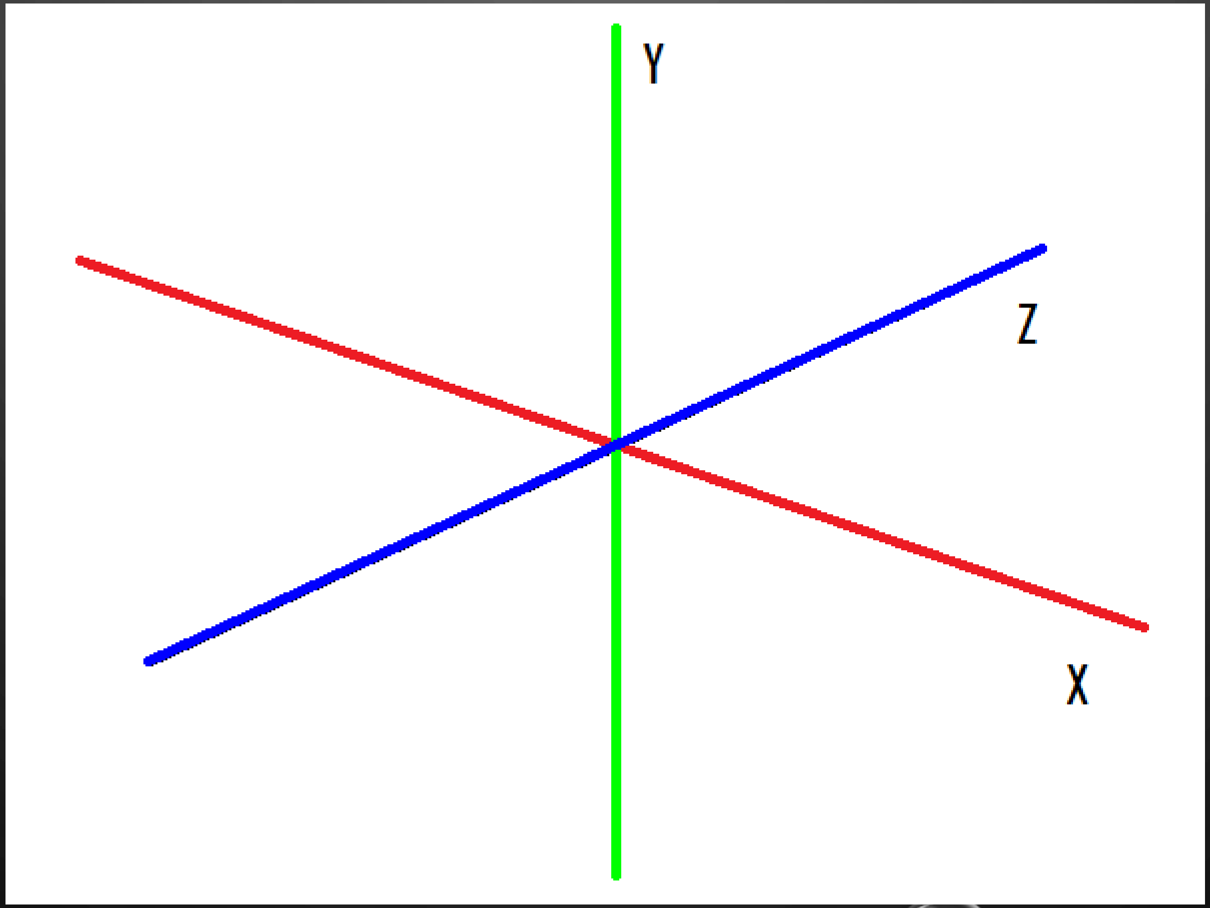
$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 2.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 15 \\ 25 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 7.5 \\ 62.5 \\ 1 \end{bmatrix}$$

MINI LECTURE 1
3D COORDINATES SYSTEM & 3D VIEW
• • •

3D COORDINATES SYSTEM



3D PROJECTIONS

1. It is mapping of 3D objects onto a 2D screen.

2. There are two types of viewing projections, which are orthographic views and perspective view.

ORTHOGRAPHIC VIEW

1. Ortho means perpendicular in Greek.

2. To project a 3D point at $A(a_x, a_y, a_z)$ to image point at $B(b_x, b_y)$,

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

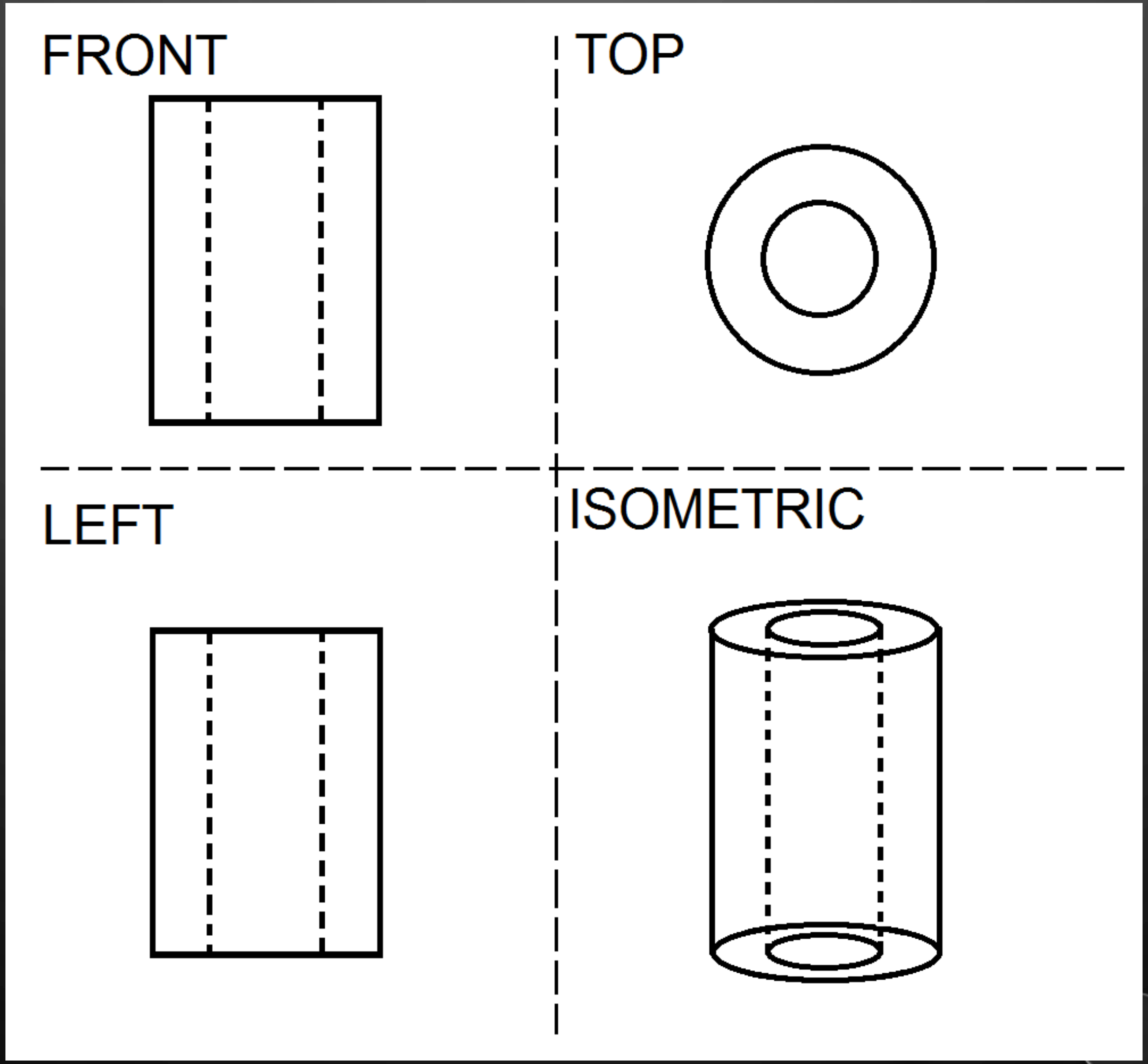
where $S(s_x, s_y)$ is the scaling factor and $C(c_x, c_z)$ is the offset for the screen.

ORTHOGRAPHIC VIEWS

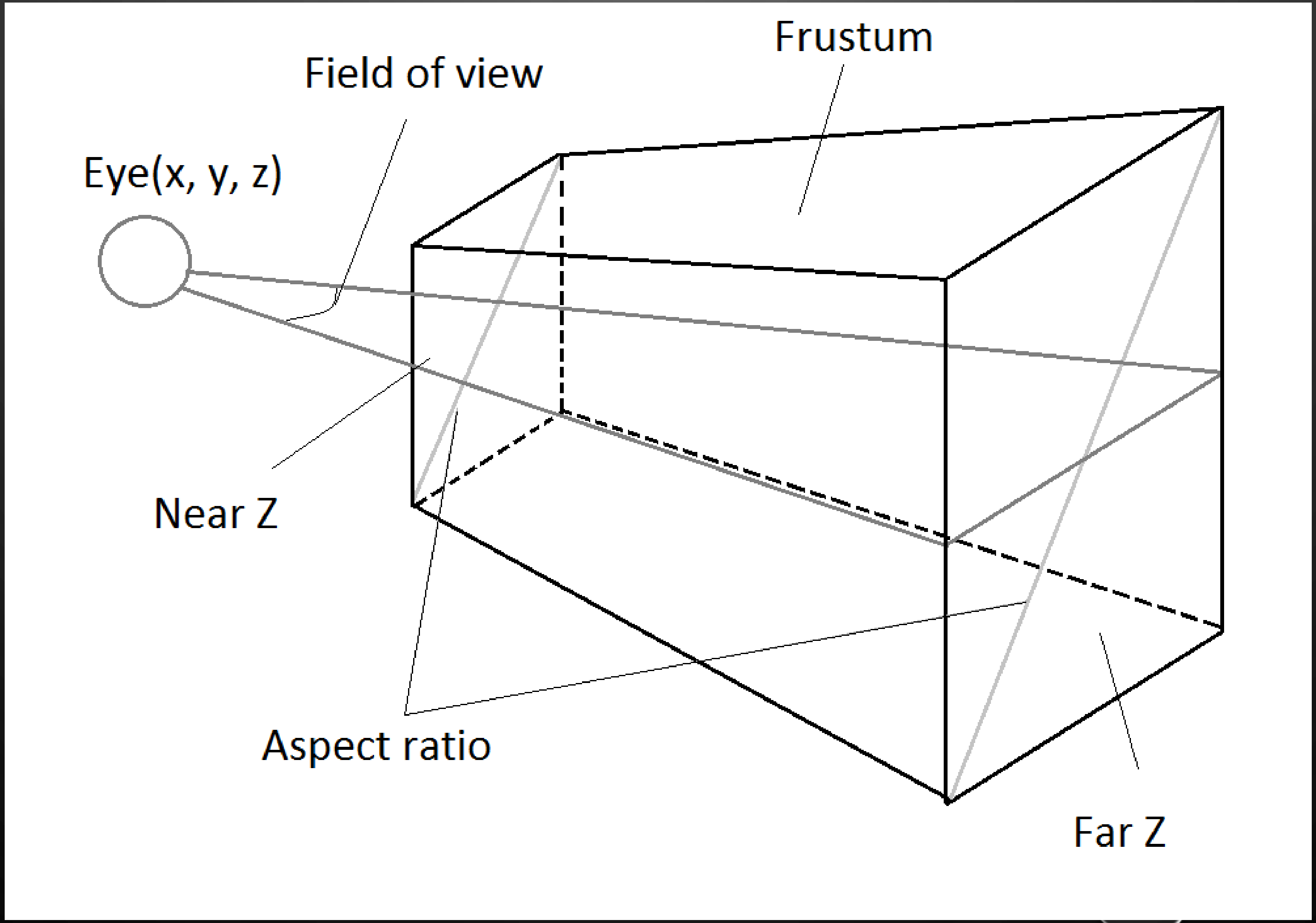
1. It consists of six sides to view a 3D model, which are top, bottom, left, right, front, and back.

2. A standard orthographic views are front, top, left, and isometric views.

ORTHOGRAPHIC VIEWS: EXAMPLE



PERSPECTIVE VIEW



EXERCISE 1

This activity will takes about ten minutes.

Compute the position of an image point, (x', y'), pr
3D point, (x, y, z), scaling factor, (s_x, s_y), and the of

a. (4, 5, 6), (1, 1), and (0, 0)

b. (72, 53, 64), (2, 2), and (10, 10)

c. (28, 30, 40), (½, ½), and (1, 15)

d. (16, 16, 16), (1, 1), and (5, 5)

e. (32, 32, 64), (¼, ¼), and (20, 25)

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} sx & 0 & 0 \\ 0 & 0 & sy \end{pmatrix}\begin{pmatrix} ax \\ ay \\ az \end{pmatrix} + \begin{pmatrix} cx \\ cz \end{pmatrix}$$

a) $$\begin{pmatrix} sx & 0 & 0 \\ 0 & 0 & sy \end{pmatrix}\begin{pmatrix} ax \\ ay \\ az \end{pmatrix} + \begin{pmatrix} cx \\ cz \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$= \begin{pmatrix} 4 \\ 6 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$= \begin{pmatrix} 4 \\ 6 \end{pmatrix}$$

b) $$\begin{pmatrix} sx & 0 & 0 \\ 0 & 0 & sy \end{pmatrix}\begin{pmatrix} ax \\ ay \\ az \end{pmatrix} + \begin{pmatrix} cx \\ cz \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}\begin{pmatrix} 72 \\ 53 \\ 64 \end{pmatrix} + \begin{pmatrix} 10 \\ 10 \end{pmatrix}$$
$$= \begin{pmatrix} 144 \\ 128 \end{pmatrix} + \begin{pmatrix} 10 \\ 10 \end{pmatrix}$$
$$= \begin{pmatrix} 154 \\ 138 \end{pmatrix}$$

c) $$\begin{pmatrix} sx & 0 & 0 \\ 0 & 0 & sy \end{pmatrix}\begin{pmatrix} ax \\ ay \\ az \end{pmatrix} + \begin{pmatrix} cx \\ cz \end{pmatrix} = \begin{pmatrix} 0.5 & 0 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}\begin{pmatrix} 28 \\ 30 \\ 40 \end{pmatrix} + \begin{pmatrix} 1 \\ 15 \end{pmatrix}$$
$$= \begin{pmatrix} 14 \\ 20 \end{pmatrix} + \begin{pmatrix} 1 \\ 15 \end{pmatrix}$$
$$= \begin{pmatrix} 15 \\ 35 \end{pmatrix}$$

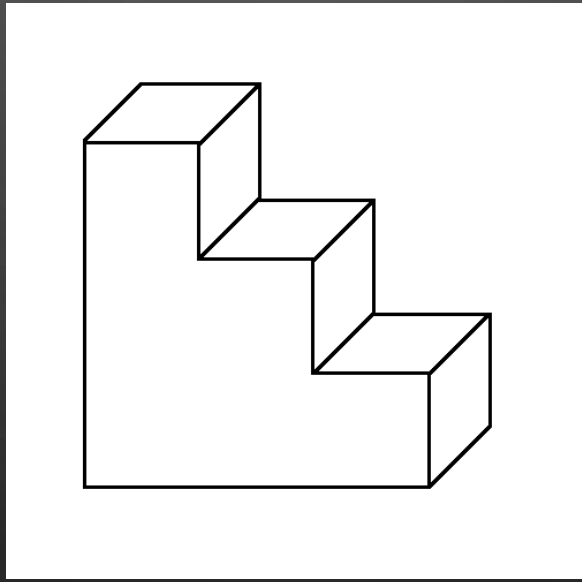
d) $$\begin{pmatrix} sx & 0 & 0 \\ 0 & 0 & sy \end{pmatrix}\begin{pmatrix} ax \\ ay \\ az \end{pmatrix} + \begin{pmatrix} cx \\ cz \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 16 \\ 16 \\ 16 \end{pmatrix} + \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$
$$= \begin{pmatrix} 16 \\ 16 \end{pmatrix} + \begin{pmatrix} 5 \\ 5 \end{pmatrix}$$
$$= \begin{pmatrix} 21 \\ 21 \end{pmatrix}$$

e) $$\begin{pmatrix} sx & 0 & 0 \\ 0 & 0 & sy \end{pmatrix}\begin{pmatrix} ax \\ ay \\ az \end{pmatrix} + \begin{pmatrix} cx \\ cz \end{pmatrix} = \begin{pmatrix} 0.25 & 0 & 0 \\ 0 & 0 & 0.25 \end{pmatrix}\begin{pmatrix} 32 \\ 32 \\ 64 \end{pmatrix} + \begin{pmatrix} 20 \\ 25 \end{pmatrix}$$
$$= \begin{pmatrix} 8 \\ 16 \end{pmatrix} + \begin{pmatrix} 20 \\ 25 \end{pmatrix}$$
$$= \begin{pmatrix} 28 \\ 41 \end{pmatrix}$$

EXERCISE 1.2

This activity will takes about five minutes.

Illustrate a standard orthographic views for the following 3D models.



MINI LECTURE 2
MATRIX TRANSFORMATIONS IN 3D SPACE

…

3D TRANSLATION

1. To find an image point, (x', y', z') that translate an original point, (x, y, z) with a translation vector, (t_x, t_y, t_z).

2. Image point,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

3D ROTATION IN CLOCKWISE

1. To find an image point, (x’, y’, z’) that rotates an original point, (x, y, z) at certain degrees, *θ* in clockwise (CW) direction with respect to x-axis.

2. Image point,

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

3D ROTATION IN CLOCKWISE

1. To find an image point, (x', y', z') that rotates an original point, (x, y, z) at certain degrees, θ in clockwise (CW) direction with respect to y-axis.

2. Image point,

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

3D ROTATION IN CLOCKWISE

1. To find an image point, (x’, y’, z’) that rotates an original point, (x, y, z) at certain degrees, θ in clockwise (CW) direction with respect to z-axis.

2. Image point,

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

3D ROTATION IN COUNTER-CLOCKWISE

1. To find an image point, (x', y', z') that rotates an original point, (x, y, z) at certain degrees, θ in counter-clockwise (CCW) direction with respect to x-axis.

2. Image point,

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

3D ROTATION IN COUNTER-CLOCKWISE

1. To find an image point, (x', y', z') that rotates an original point, (x, y, z) at certain degrees, θ in counter-clockwise (CCW) direction with respect to y-axis.

2. Image point,

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

3D ROTATION IN COUNTER-CLOCKWISE

1. To find an image point, (x', y', z') that rotates an original point, (x, y, z) at certain degrees, θ in counter-clockwise (CCW) direction with respect to z-axis.

2. Image point,

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

SCALE

1. To find an image point, (x', y', z') that scale an original point, (x, y, z) on a scaling factor, (s_x, s_y, s_z).

2. Image point,

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

EXERCISE 2

This activity will takes about ten minutes.

Given a 3D original point at (30, 20, 15) in the 3D spac

corresponding image point with the following matrix tra

(a) Translate with a vector of (-5, -10, 12).

(b) Rotate clockwise (CW) at 45 degrees along y-axis

(c) Rotate counter-clockwise (CCW) at 90 degrees al

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

a) $$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & tx \\ 0 & 1 & 0 & ty \\ 0 & 0 & 1 & tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -5 \\ 0 & 1 & 0 & -10 \\ 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 30-5 \\ 20-10 \\ 15-12 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 25 \\ 10 \\ 3 \\ 1 \end{bmatrix}$$

b) $$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(t) & 0 & \sin(t) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(t) & 0 & \cos(t) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(45) & 0 & \sin(45) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(45) & 0 & \cos(45) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 30\cos(45) + 15\sin(45) \\ 20 \\ -30\sin(45) + 15\cos(45) \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 21.21 + 10.61 \\ 20 \\ -21.21 + 10.605 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 31.82 \\ 20 \\ -10.61 \\ 0 \end{bmatrix}$$

c) $$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(t) & -\sin(t) & 0 & 0 \\ \sin(t) & \cos(t) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \begin{bmatrix} \cos(90) & -\sin(90) & 0 & 0 \\ \sin(90) & \cos(90) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 30\cos(90) - 20\sin(90) \\ 30\sin(90) + 20\cos(90) \\ 15 \\ 0 \end{bmatrix}$$

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

d) $$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} sx & 0 & 0 & 0 \\ 0 & sy & 0 & 0 \\ 0 & 0 & sz & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 30 \\ 20 \\ 15 \\ 1 \end{bmatrix} = \begin{bmatrix} 60 \\ 60 \\ 15 \\ 1 \end{bmatrix}$$

BREAK

...

MINI LECTURE 3
MATRIX STACKS & ITS OPERATIONS

…

MATRIX IDENTITY

It is a zero matrix whereby the diagonal entries are ones.

$$I_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

MATRIX IDENTITY

Function name	glLoadIdentity
Purpose	It reset the current matrix mode to identity matrix.
Arguments or parameters	None
Return value	None

SET MATRIX MODE

Function name	glMatrixMode
Purpose	It set the matrix mode for operations of matrices.
Arguments or parameters	GL_PROJECTION GL_MODELVIEW GL_TEXTURE
Return value	None

MATRIX MODES

Modes	Description
GL_PROJECTION	Projection matrix stack uses to clip a scene or a volume.
GL_MODELVIEW	Mode view matrix stack uses to move around objects in the scene.
GL_TEXTURE	Texture matrix stack uses to manipulate texture coordinates.
GL_COLOR	Colour matrix stack uses to manipulate images.

SET A PERSPECTIVE VIEW

Function name	gluPerspective
Purpose	Specify the viewing frustum.
Arguments or parameters	GLdouble fov GLdouble ar GLdouble nearZ GLdouble farZ
Return value	None

PERSPECTIVE VIEW PARAMETERS

Mode	Description
fov	It set the field of view (FOV) in degrees, in the x direction.
ar	It set the aspect ratio for the clipping planes (near and far).
nearZ	Distance between eye or viewer to the near clipping plane. nearZ > 0.
farZ	Distance between eye or viewer to the far clipping plane. farZ > 0.

CLIP A PERSPECTIVE VIEW

To clip a perspective view in 3D space.

```
void view(){
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(viewer.fov, viewer.ar,
                 viewer.nearZ, viewer.farZ);
}
```

SET VIEW POINT

Function name	gluLookAt
Purpose	It defines a viewing transformations.
Arguments or parameters	Eye point, (eyeX, eyeY, eyeZ) Reference point, (centerX, centerY, centerZ) Look up vector (upX, upY, upZ)
Return value	None.

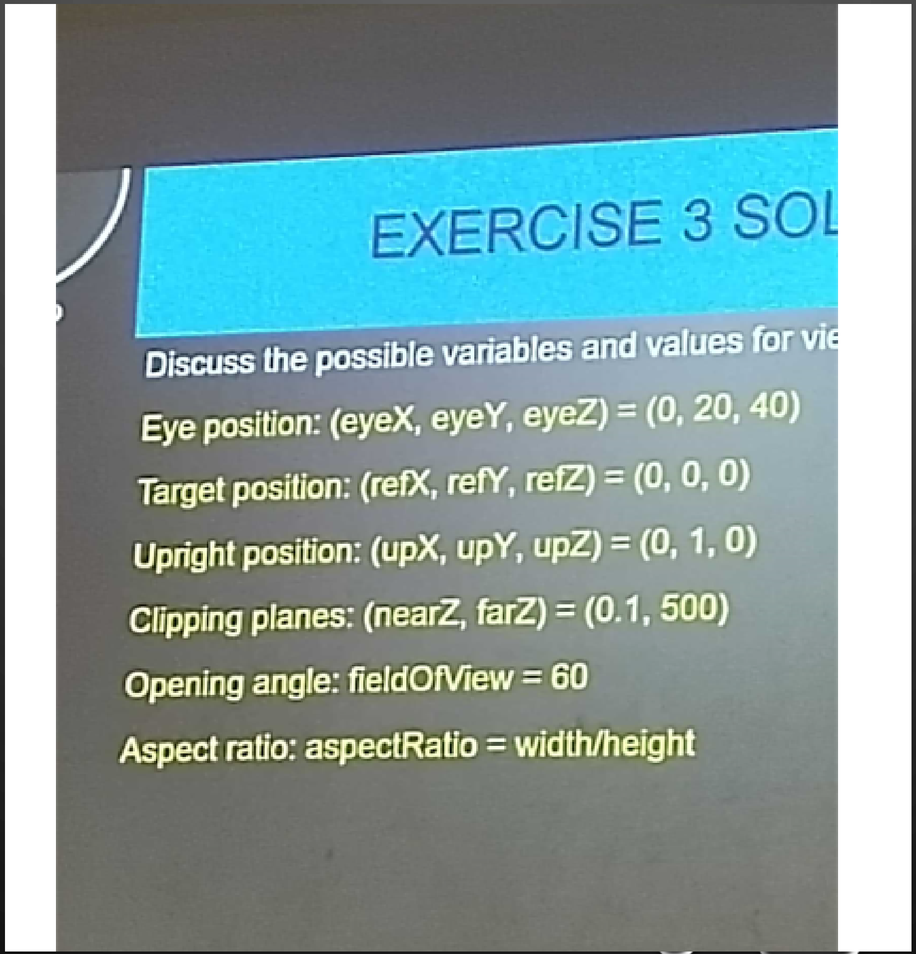
EYE ON PERSPECTIVE VIEW

To pick a look at a perspective view in 3D space.

```
void view(){
  glMatrixMode(GL_MODELVIEW);
  glLoadIdentity();
  gluLookAt(viewer.eyeX, viewer.eyeY, viewer.eyeZ
             viewer.centerX, viewer.centerY,
             viewer.centerZ, viewer.upX, viewer.upY,
             viewer.upZ);
}
```

EXERCISE 3

Discuss the possible variables and values for viewer class of a 3D frustum.



MINI LECTURE 4
CONFIGURE A 3D SPACE

…

PUSH MATRIX

Function name	glPushMatrix
Purpose	It adds current matrix onto matrix stack specified by glMatrixMode function.
Arguments or parameters	None
Return value	None

POP MATRIX

Function name	glPopMatrix
Purpose	It removes the top matrix from the matrix stack specified by glMatrixMode function.
Arguments or parameters	None
Return value	None

3D TRANSLATION

Function name	glTranslated, glTranslatef
Purpose	It times the current matrix by a translation matrix.
Arguments or parameters	Translation vector, (t_x, t_y, t_z)
Return value	None

3D ROTATION

Function name	glRotated, glRotatef
Purpose	It times the current matrix by a rotation matrix.
Arguments or parameters	Angle, angle of rotation; (x, y, z), rotation vector;
Return value	None

3D SCALE

Function name	glScaled, glScalef
Purpose	It times the current matrix by a scaling matrix.
Arguments or parameters	(x, y, z) specifies the scaling factors in x, y and z axes.
Return value	None

SWAP BUFFER

Function name	glutSwapBuffers
Purpose	It shows the backward frame buffer.
Arguments or parameters	None
Return value	None

UPDATE DISPLAY

Function name	glutPostRedisplay
Purpose	It redraws display after updates.
Arguments or parameters	None
Return value	None

EXERCISE 4

Write a render function in C++ OpenGL to translate, rotate, and scale a 3D coordinates system.

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.