01.04.18

OpenGL

A graphics API

model
light
camera
} input to a graphics API (OpenGL) ⟶ image

4 assignments

Assignment 1 ⟶ Use of openGL
Assignment 2, 3 ⟶ Implementation of OpenGL
Assignment 4 ⟶ Ray tracer

display, then idle automatically
~~Otherwise~~
But display has to be called inside idle function
(postredisplay)

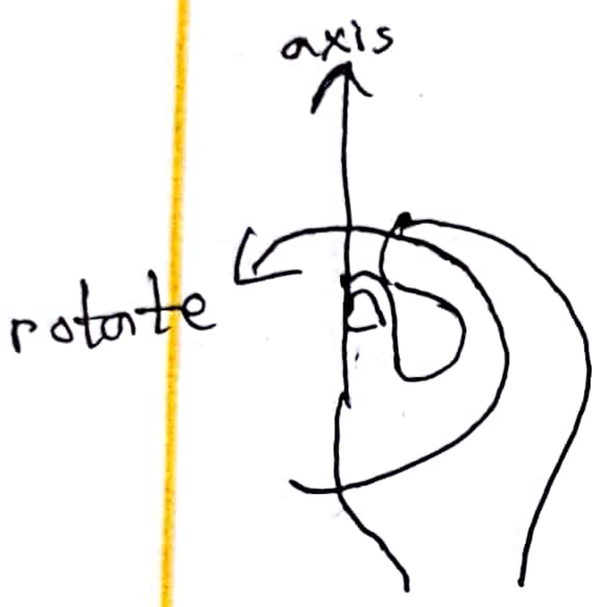
glutKeyboardFunc
glutSpecialFunc
} Handles keyboard

glutMouseFunc

event & handler

mplexity

```
glTranslatef(          );
drawSquare(10);
```



glRotatef

* * Axis of rotation always goes through the origin.

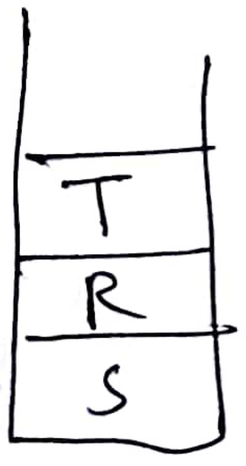
** $2\hat{i} + 3\hat{j} - 4\hat{k}$ can be even the axis of rotation

glRotate(20, 2, 3, -4)

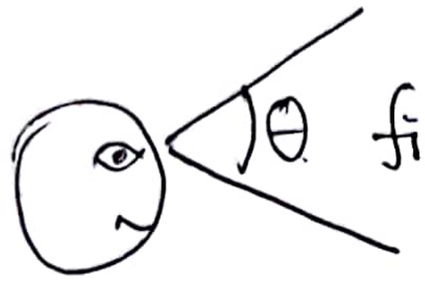
** Order of transformation is <u>very important</u>. Different orders produce different images.

Transformation stack → doesn't get empty

Scale
Rotate
Translate



gluLookAt
gluPerspective → property of camera



Aspect ratio → ratio of field of views in X and Y

Light → before ray tracing, we will assume global ambient lighting

~~glBegin (GL_BEGIN LINES)~~ → for line

```
glBegin (GL_LINES); {
    // Now takes two points at a time and draws a line



}glEnd();
```

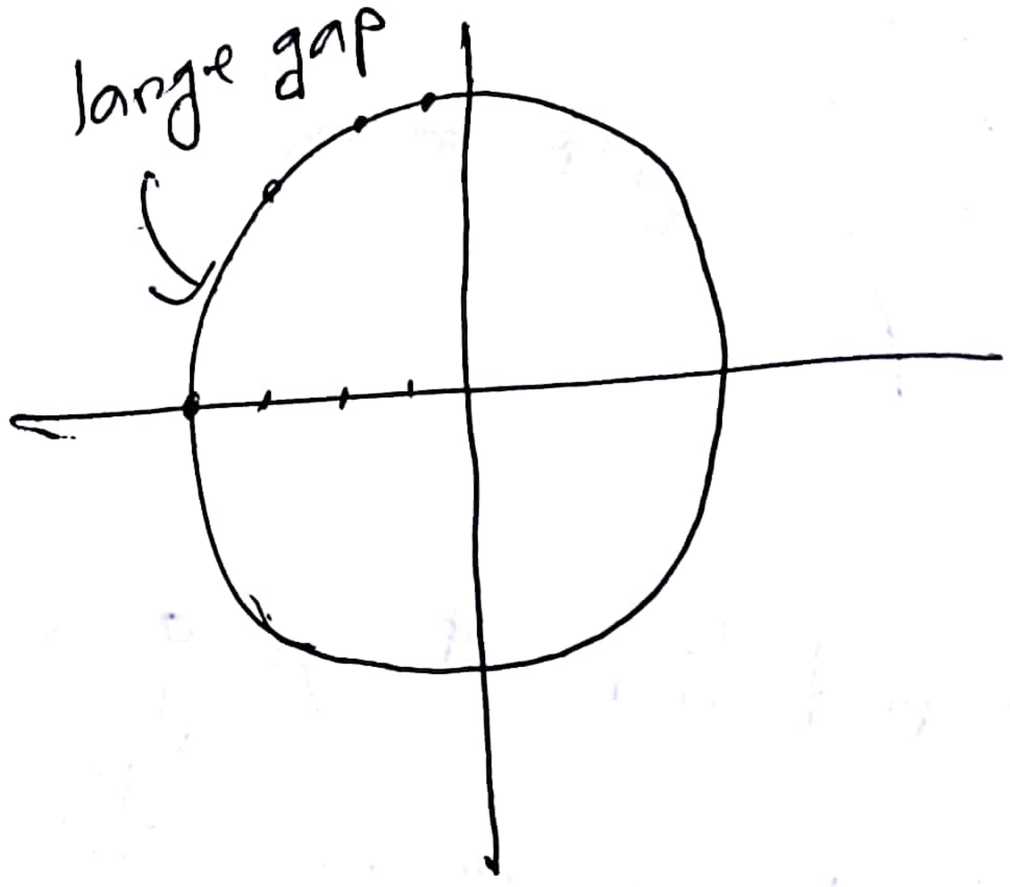
Transformations in OpenGL
- Translate (glTranslatef)
- Scale (glScalef)
- Rotate (glRotatef)

glPushMatrix() → save the state of stack
glPopMatrix() → last যে অবস্থায় save করা আছে, সেটা restore করবে

DrawCircle()

$x^2+y^2=25$

$y = \pm\sqrt{25-x^2}$

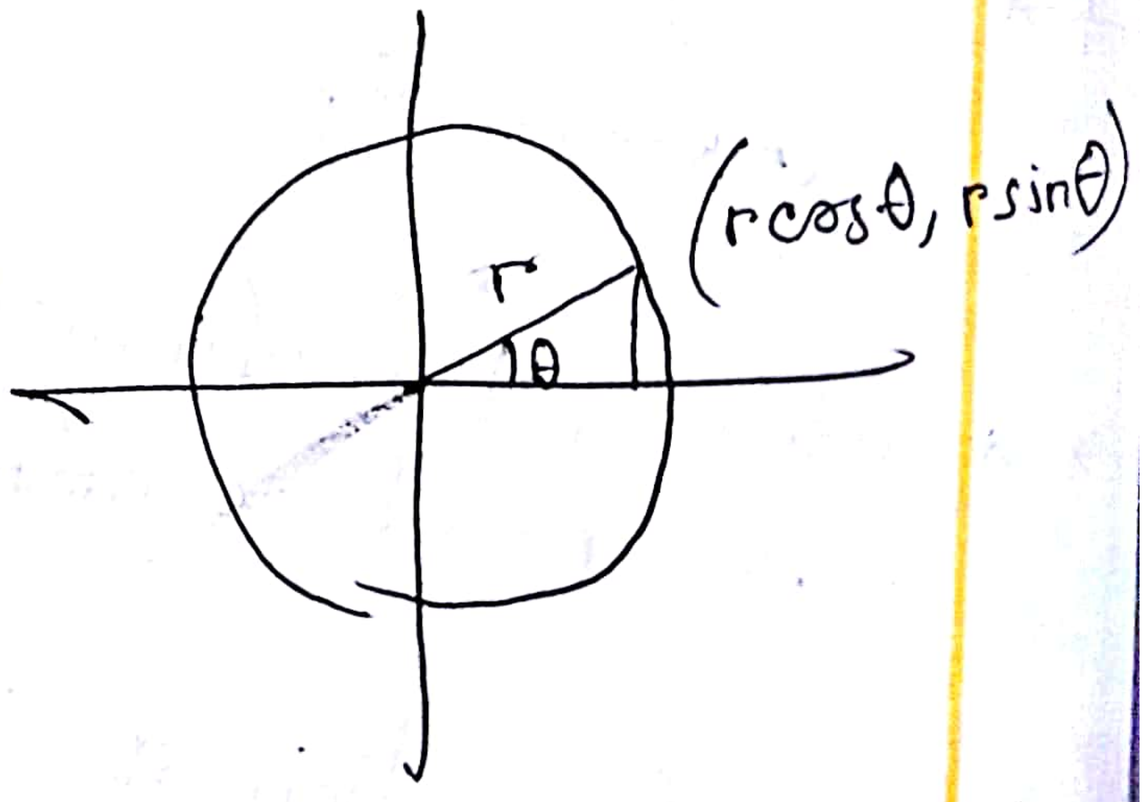


Better approach:

Use parametric equation

$x(t) = r\cos t$

$y(t) = r\sin t$



Ellipse:

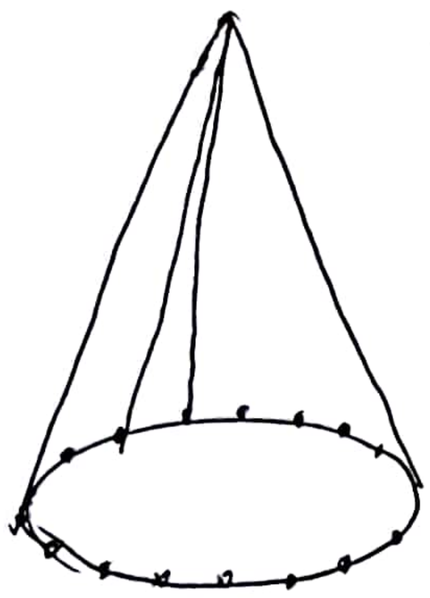
$x(t) = a\cos t$

$y(t) = b\sin t$

* Transformation is applied to points only, not a shape.

DrawCone

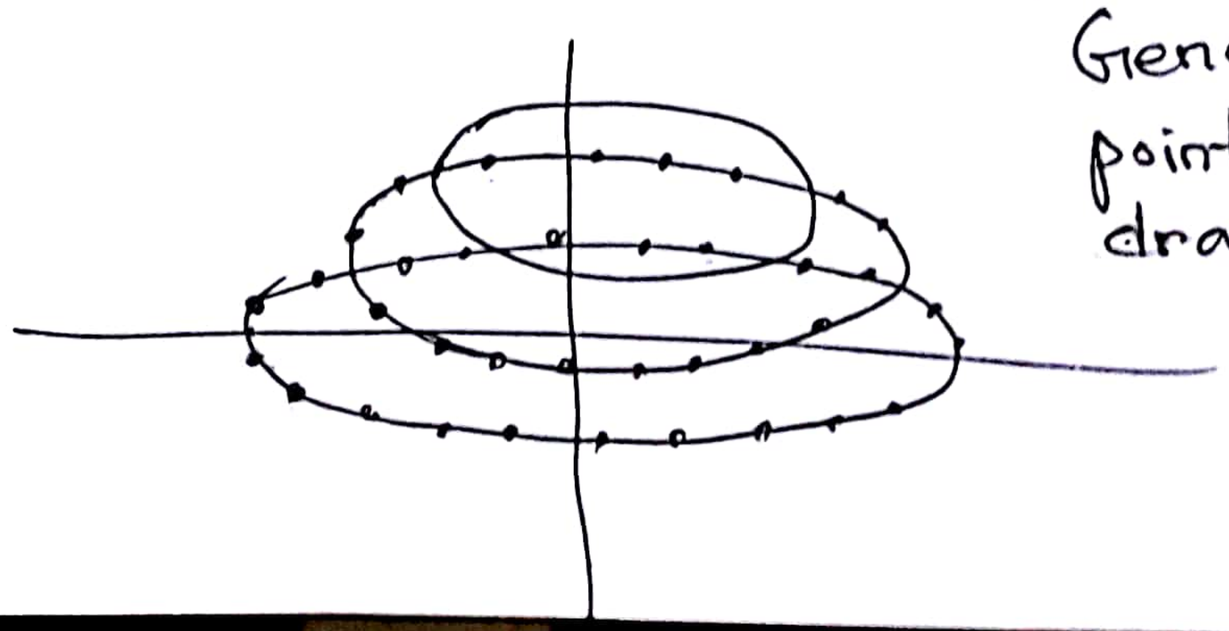
DrawCircle Circle Points
Draw Triangles



DrawCylinder → Draw 2 circle points
Then Draw 8 rectangles

* Points of a rectangle must be mentioned in circular order.

Sphere



Generate circle points, then draw rectangles

Assignment:

- Green → Cyclinder (90°, not 360°) → 12
- Brown → Sphere (one-eighth) → 8
- White → Plane → 6

→ 3 different functions

Then use translate, rotate, scale