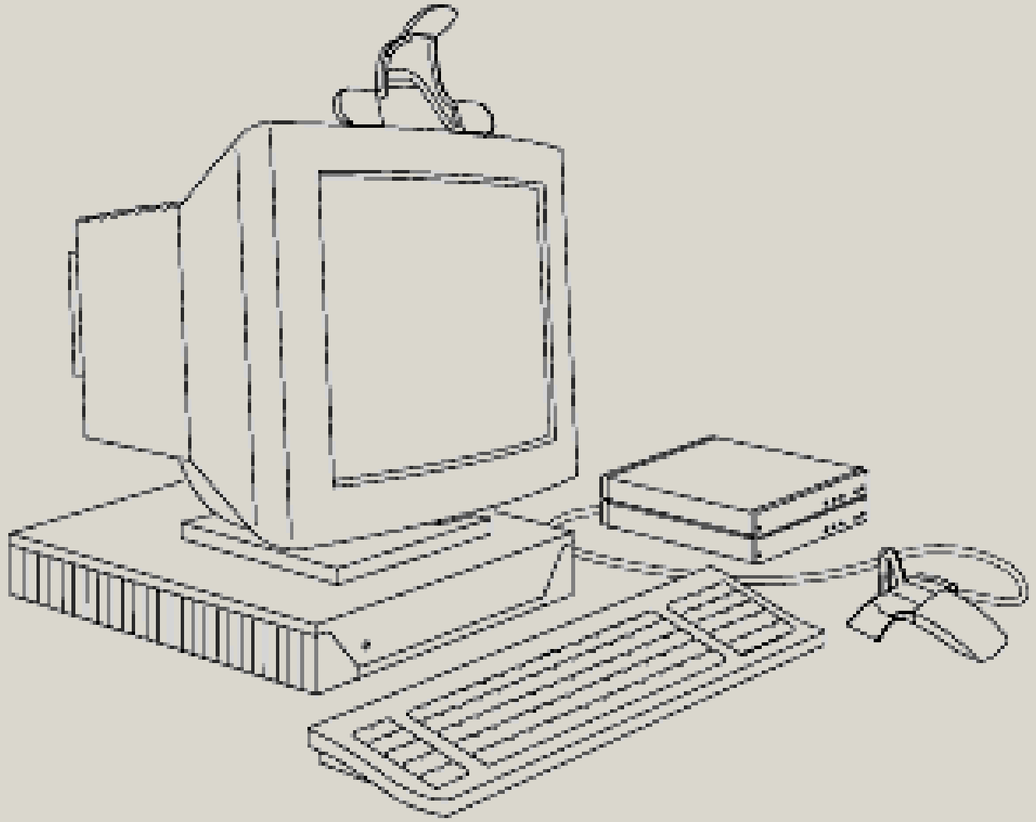
DRAWING OBJECTS

SCREEN SYSTEM

Measured in pixels:

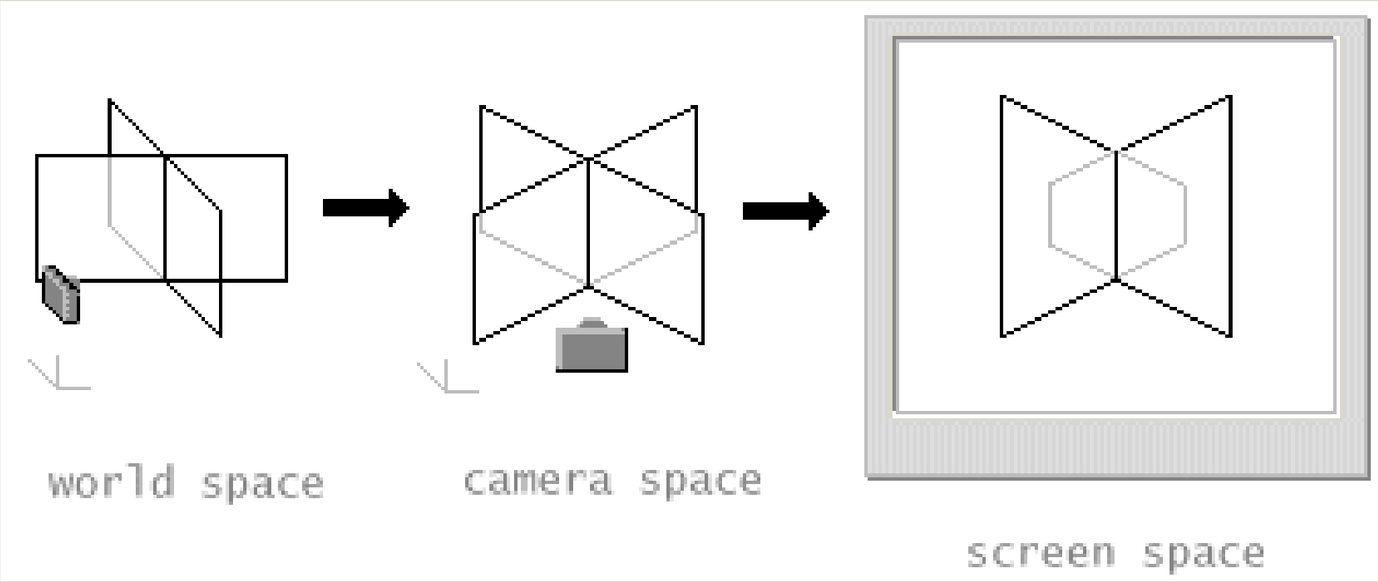
- x : 0 --> screenWidth - 1
- y: 0 --> screenHeight - 1

Upper left corner is (0,0)



WORLD WINDOW

the rectangular window in world coordinates, specifying which part of the world to be drawn.



VIEWPORT

a rectangular space defined in the screen window mapping between the world window and the viewport is established

- **objects in world window are displayed in viewport**
- **the viewport can be displayed in anywhere of the screen window**
- **the viewport can be displayed in anywhere of the screen window**

2D DRAWING

gluOrtho2D()

- set world window

glViewport()

- set view ports
- Restrict OpenGL to drawing in only part of the window (divides the window)
- Default viewport is entire window

glutInitWindowSize()

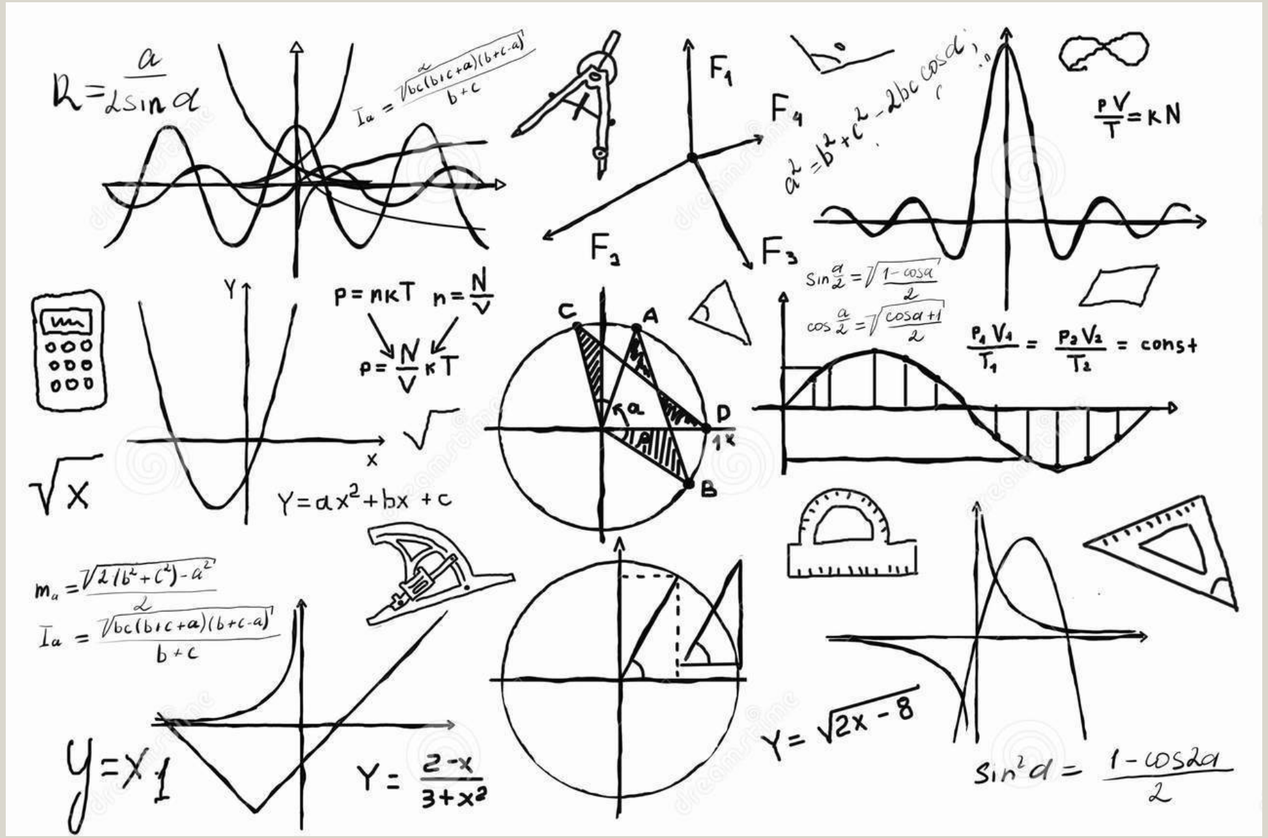
set screen window size

glutInitWindowPosition()

set screen window position

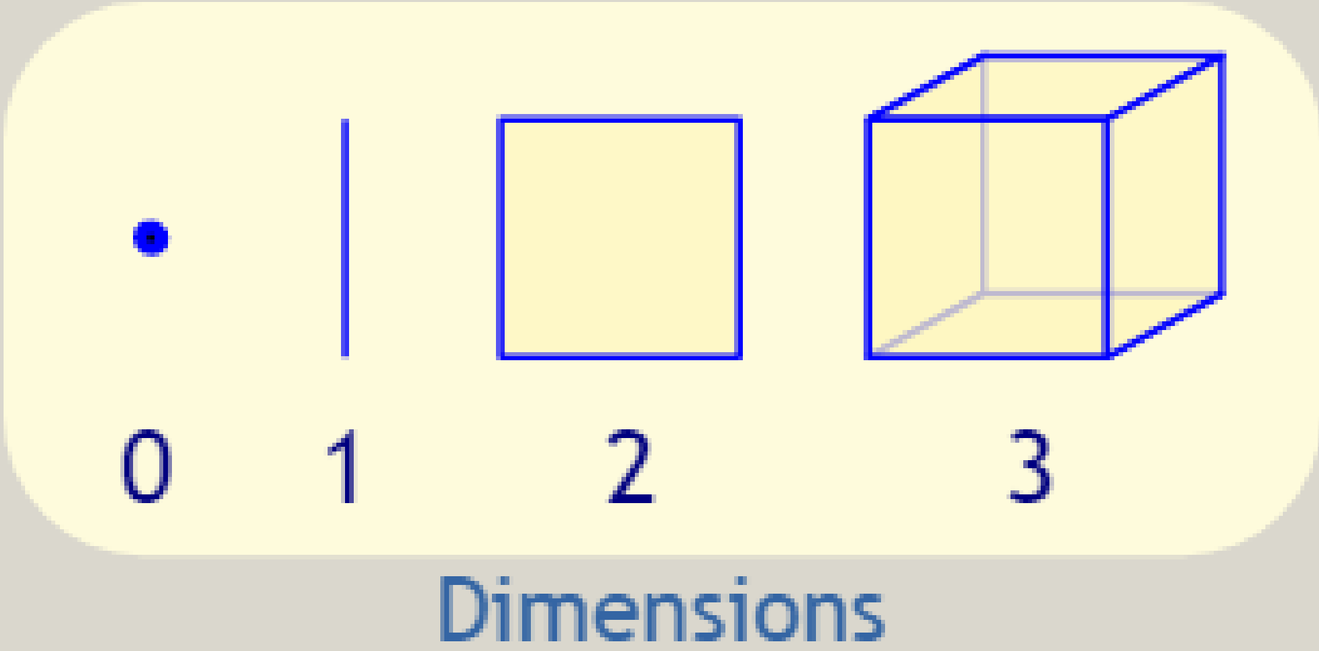
GEOMETRY

Geometry will guide you through among other things points, lines, planes, angles, parallel lines, triangles, similarity, trigonometry, quadrilaterals, transformations, circles and area.



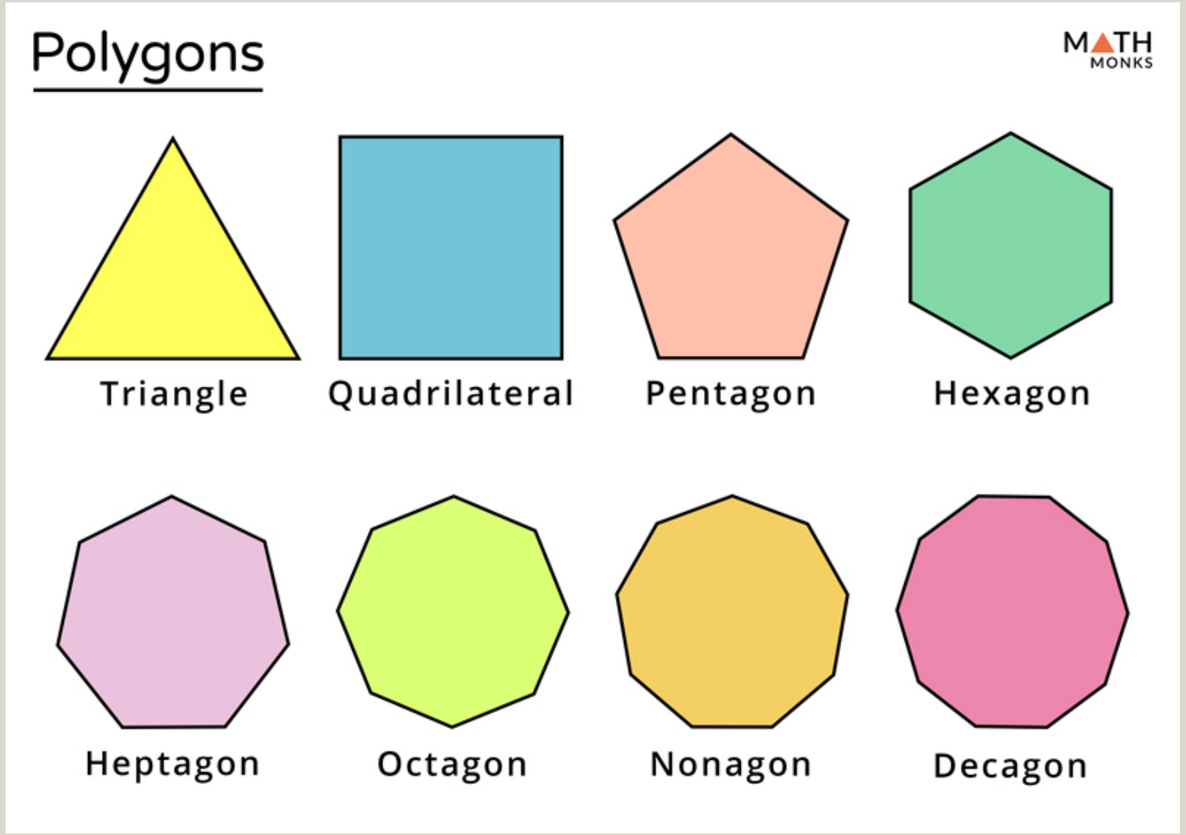
POINT, LINE, PLANE AND SOLID

- A Point has no dimensions, only position
- A Line is one-dimensional
- A Plane is two dimensional (2D)
- A Solid is three-dimensional (3D)



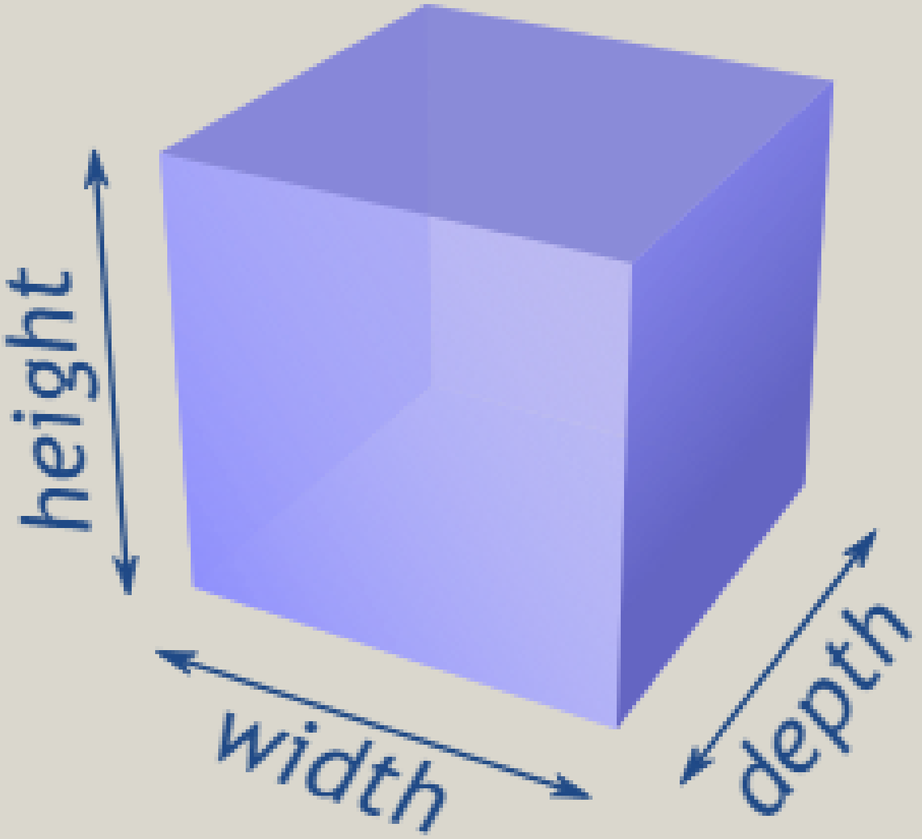
REGULAR POLYGONS

- A polygon is a closed figure where the sides are all line segments. Each side must intersect exactly two others sides but only at their endpoints.

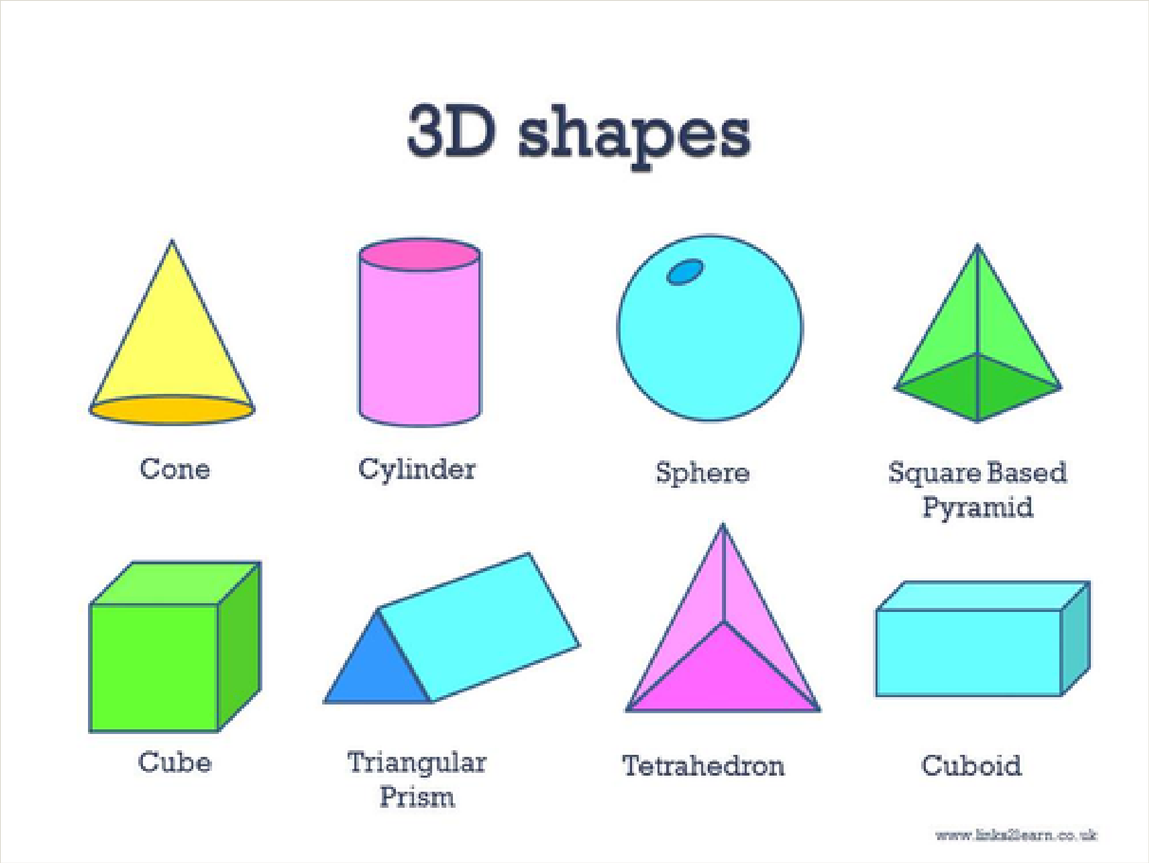


SOLID GEOMETRY

- Solid Geometry is the geometry of three-dimensional space,
- the kind of space we live in.
- three dimensions: width, depth and height.

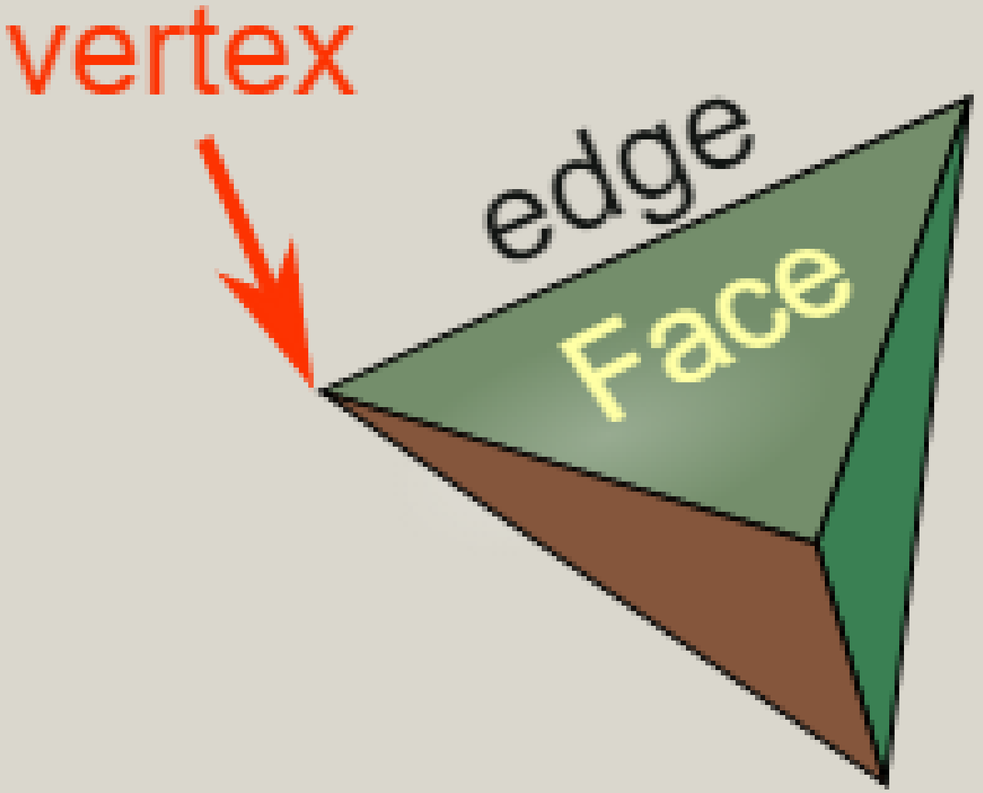


COMMON 3D SHAPES



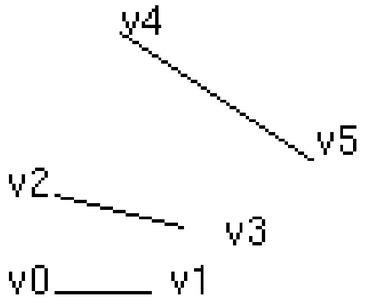
VERTICES, EDGES AND FACES

- A vertex is a corner.
- An edge is a line segment between faces.
- A face is a single flat surface.

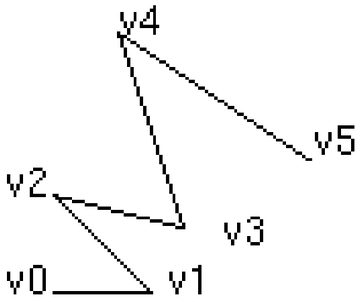


PRIMITIVES

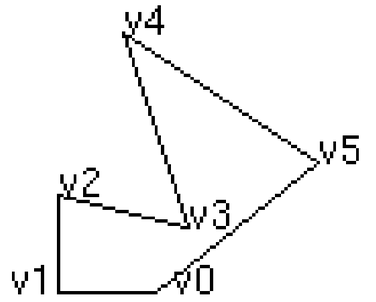
- In OpenGL, an object is made up of geometric primitives such as triangle, quad, line segment and point. A primitive is made up of one or more vertices.
- A geometric primitive is defined by specifying its vertices via glVertex function, enclosed within a pair glBegin() and glEnd().



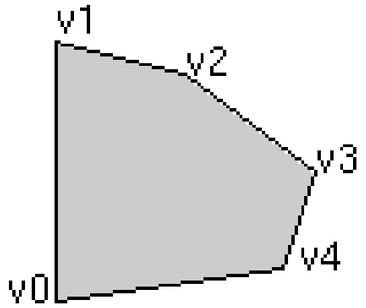
GL_LINES



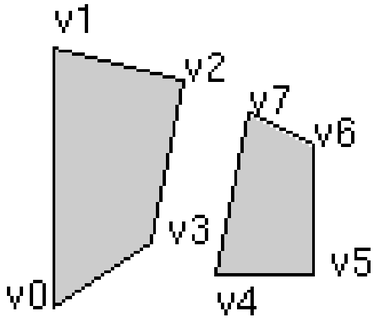
GL_LINE_STRIP



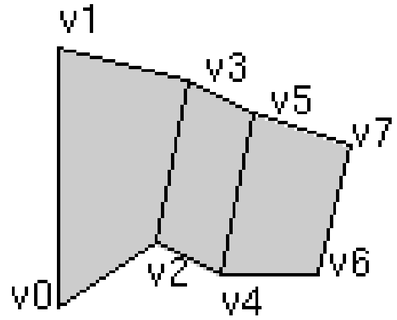
GL_LINE_LOOP



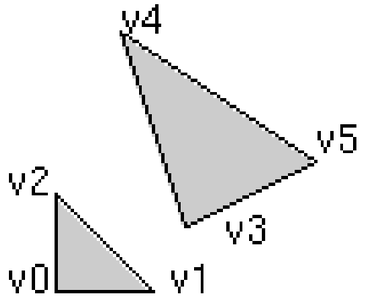
GL_POLYGON



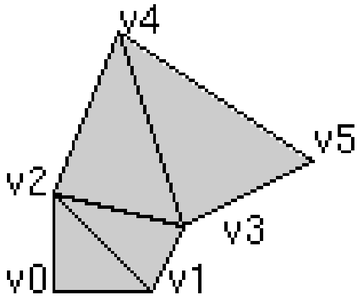
GL_QUADS



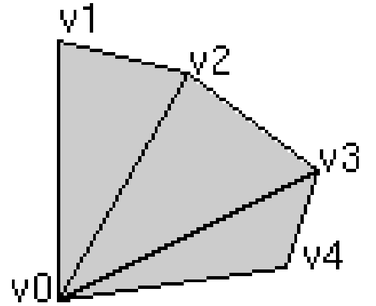
GL_QUAD_STRIP



GL_TRIANGLES



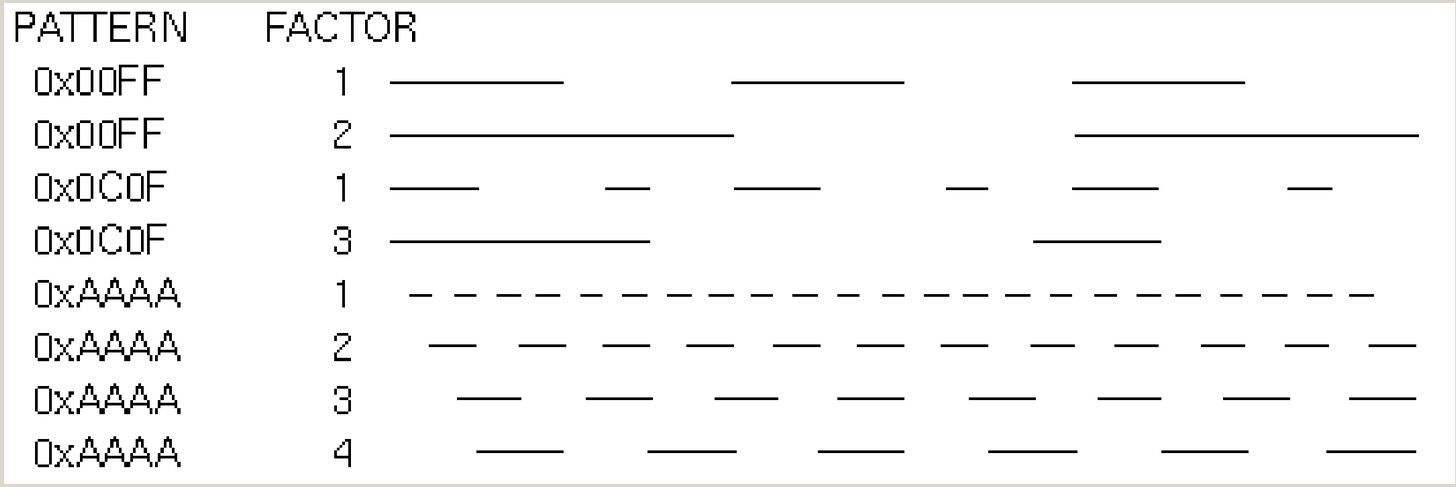
GL_TRIANGLE_STRIP



GL_TRIANGLE_FAN

DISPLAYING POINTS, LINES

- void glPointSize(GLfloat size); - set the width in pixels for rendered point; size > 0.0, default is 1.0
- void glLineWidth(GLfloat width); -set the width in pixels for rendered line; width > 0.0, default is 1.0
- glLineStipple (GLint factor, GLushort pattern);
  - sets the current stippling pattern for lines
  - pattern is 16-bit number that sets the pattern, 1 => drawing; 0 => no drawing
  - facotr (1 to 256), for repeating pattern



EXTRA LEARNING LINKS

High school geometry course

https://www.khanacademy.org/math/geometry

Vector Math for 3D Computer Graphics

https://chortle.ccsu.edu/vectorlessons/vectorindex.html

“You do not need to leave your room. Remain sitting at your table and listen. Do not even listen, simply wait, be quiet, still and solitary. The world will freely offer itself to you to be unmasked, it has no choice, it will roll in ecstasy at your feet.”

— Franz Kafka