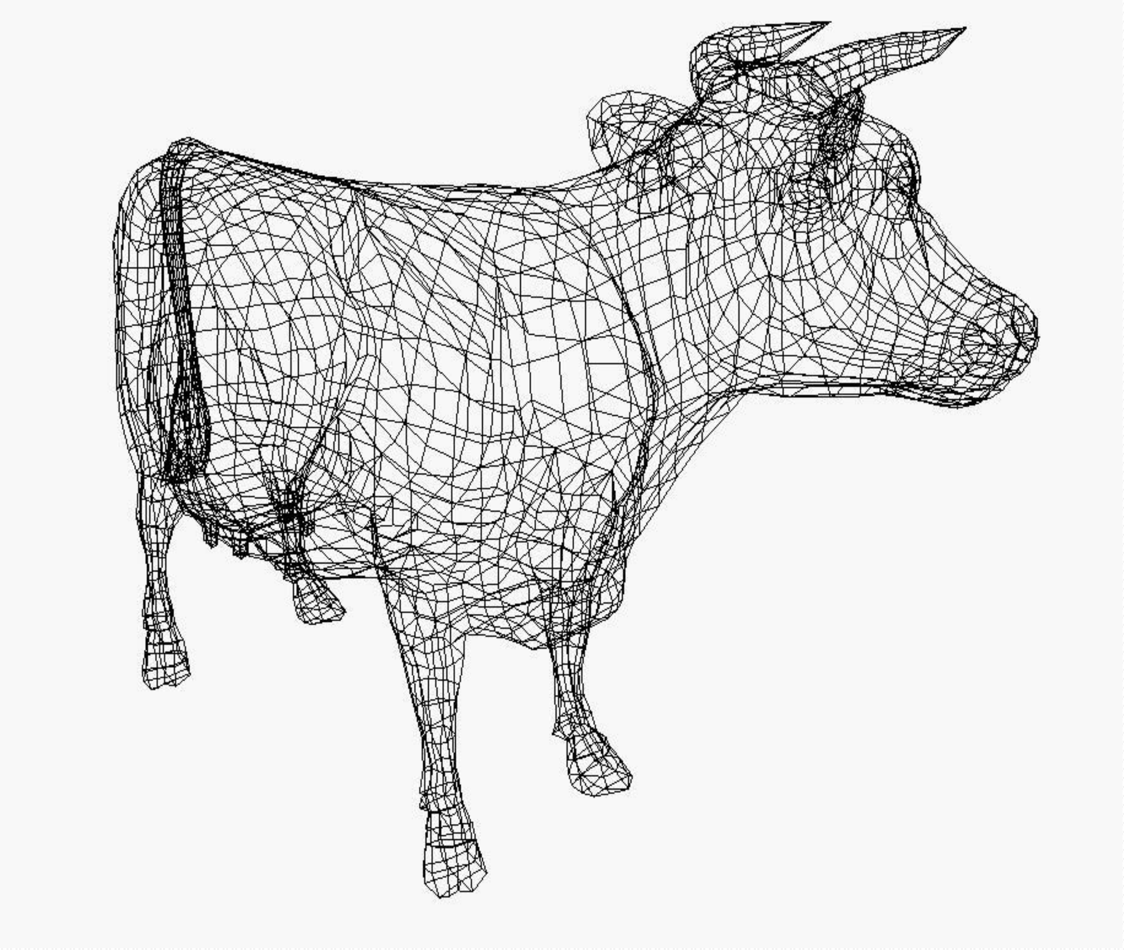
Planes, Polygons and Objects

No More Spheres

- Most things in computer graphics are not described with spheres!
- Polygonal meshes are the most common representation
- Look at how polygons can be described and how they can used in ray-casting

Polygonal Meshes



Polygons

- A polygon (face) Q is defined by a series of points

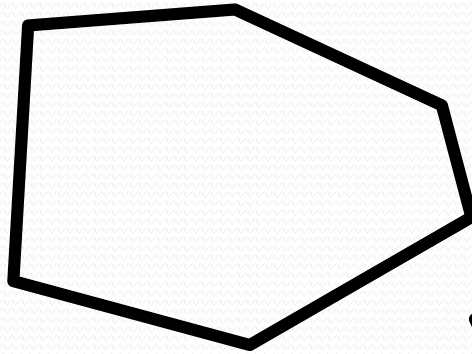
$$\left[p_0, p_1, p_2, \ldots, p_{n-1}, p_n\right]$$

$$p_i = \left(x_i, y_i, z_i\right)$$

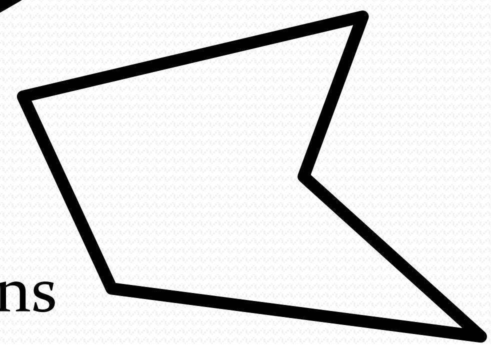
- The points are must be **co-planar**
- 3 points define a plane, but a 4th point need not lie on that plane

Convex, Concave

- Convex



- Concave



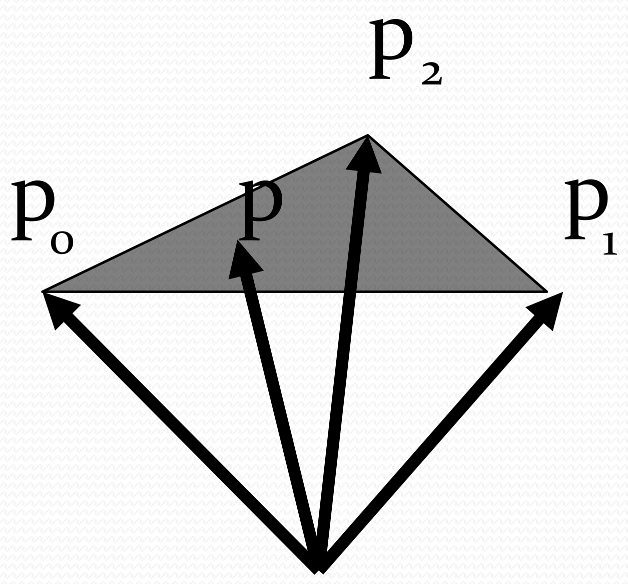
  - CG people dislike concave polygons
  - CG people would prefer triangles!!
    - Easy to break convex object into triangles, hard for concave

Equation of a Plane

$$ax + by + cz = d$$

- a,b,c and d are constants that define a unique plane and x,y and z form a vector P.

Deriving a,b,c & d (1)



- The cross product

$$n = (p_1 - p_0) \times (p_2 - p_0)$$

defines a **normal** to the plane

- There are two normals (they are opposite)
- Vectors in the plane are all orthogonal to the plane normal vector

Deriving a,b,c & d (2)

- So $p\text{-}p_o$ is normal to n therefore

$$n \cdot (p - p_0) = 0$$

- if $p=(x, y, z)$, $po=(x_{o,} y_{o,} z_o)$, $n = (n_1,n_2,n_3)$
  - $n_1(x\text{-}x_o)+ n_2(y\text{-}y_o)+n_3\ (z\text{-}z_o)=o$
  - $a= n_1$ $b= n_2$ $c= n_3$
  - $d = n_1{}^*x_o + n_2{}^*y_o + n_{3*}z_o$

Half-Space

- A plane cuts space into 2 **half-spaces**
- Define

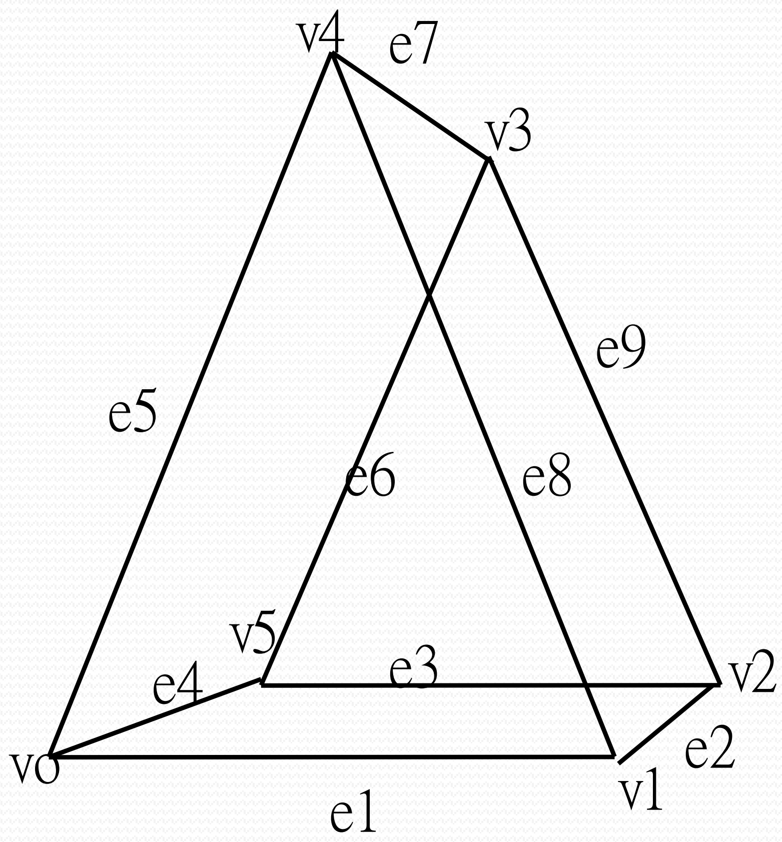
$$l(x, y, z) = ax + by + cz - d$$

- If l(p) =0
  - point on plane
- If l(p) > 0
  - point in **positive** half-space
- If l(p) <0
  - point in **negative** half-space

Polyhedra

- Polygons are often grouped together to form polyhedra
  - Each edge connects 2 vertices and is the join between two polygons
  - Each vertex joins 3 edges
  - No faces intersect
- V-E+F=2
  - For cubes, tetrahedra, cows etc…

Example Polhedron



- F0=v0v1v4
- F1=v5v3v2
- F2=v1v2v3v4
- F3=v0v4v3v5
- F4=v0v5v2v1

- V=6,F=5, E=9
- V-E+F=2

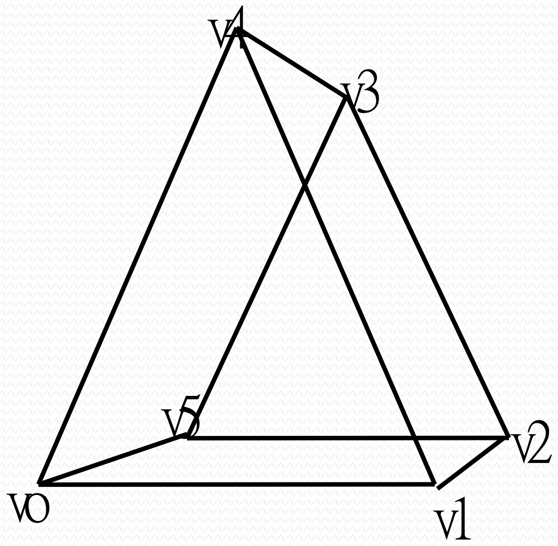
Representing Polyhedron (1)

- Exhaustive (array of vertex lists)
  - faces[1] = $(x_0,y_0,z_0),(x_1,y_1,z_1),(x_3,y_3,z_3)$
  - faces[2] = $(x_2,y_2,z_2),(x_0,y_0,z_0),(x_3,y_3,z_3)$
  - etc ….
- Very wasteful since same vertex appears at 3(or more) points in the list
  - Is used a lot though!

Representing Polyhedron (2)

- Indexed Face set
- Vertex array
  - vertices[0] = (x0,y0,z0)
  - vertices[1]=(x1,y1,z1)
  - etc ...
- Face array (list of indices into vertex array)
  - faces[0] = 0,2,1
  - faces[1]=2,3,1
  - etc ...

Vertex order matters



- Polygon v0,v1,v4 is NOT equal to v0,v4,v1
- The normal point in different directions
- Usually a polygon is only visible from points in its positive half-space
- This is known as **back-face culling**

Representing Polyhedron (3)

- Even Indexed face set wastes space
  - Each face edge is represented twice
- Winged edge data structure solves this
  - vertex list
  - edge list (vertex pairs)
  - face list (edge lists)