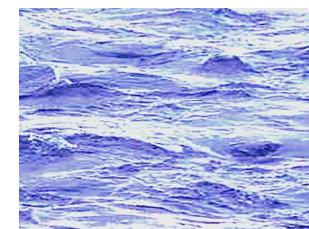


# Representing Objects

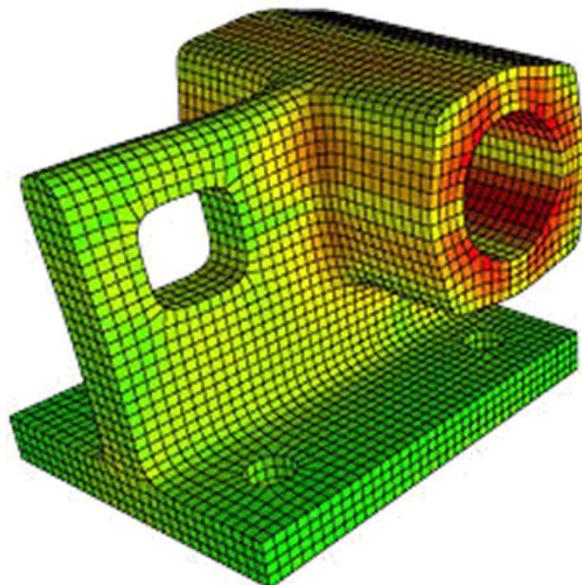
# Objects

- Graphics scenes contain
  - Solid geometric objects
  - Trees, flowers, clouds, rocks, water
- Creation of models
  - Surface ↔ interior models
  - Explicit ↔ procedural models
  - Heuristically ↔ physically based models



# Polygon Surfaces

- set of surface polygons enclose object interior  
= *Boundary Representation*  
("B-Rep")



*example:  
machine part surface  
represented by quadrilaterals*

# Polygon Surfaces

- More polys = better approximation



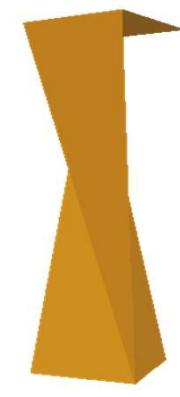
10,108 polys



1,383 polys

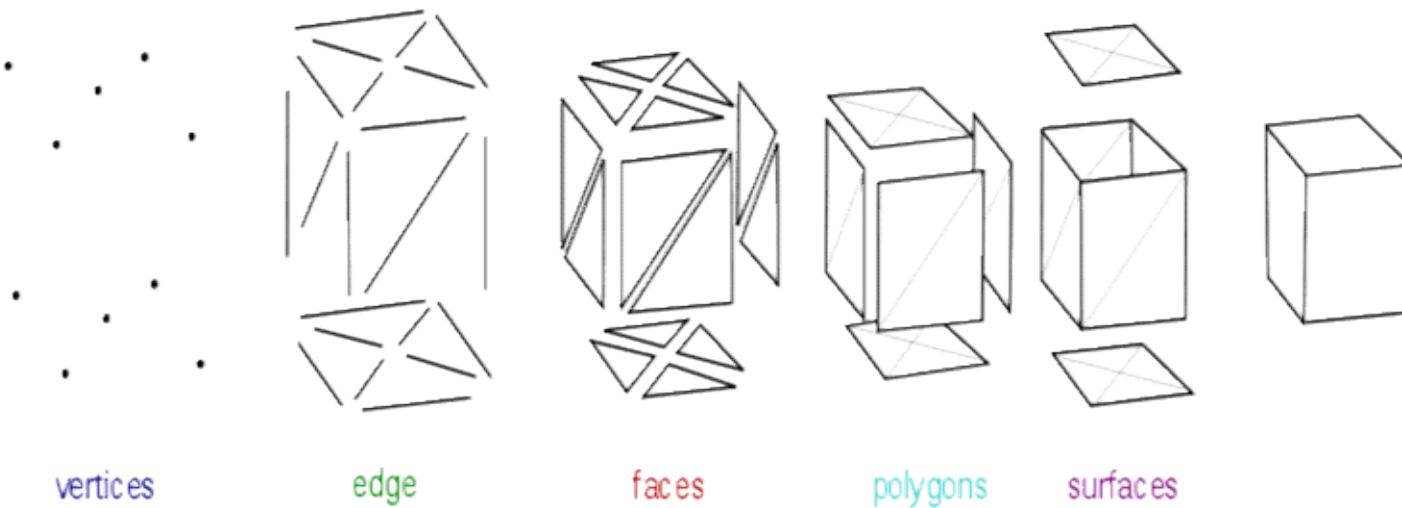


474 polys

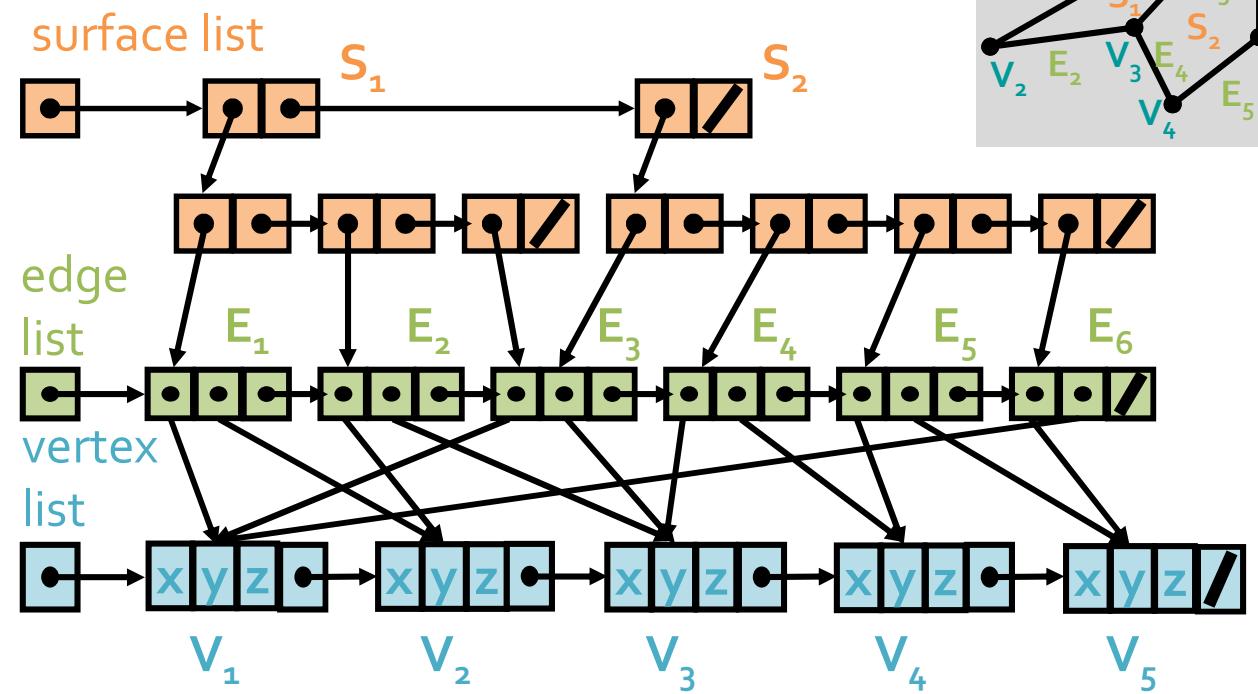


6 polys

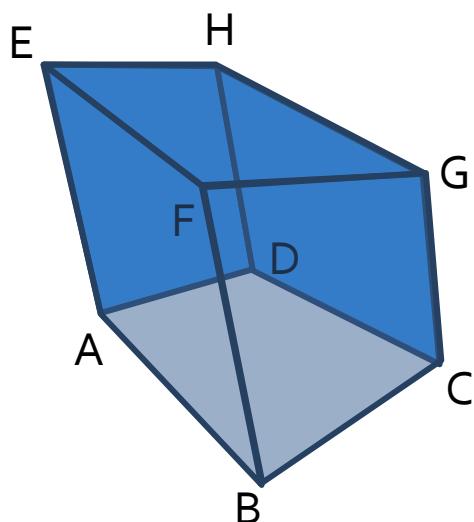
# B-Rep (Boundary Representation)



# Lists for B-Reps



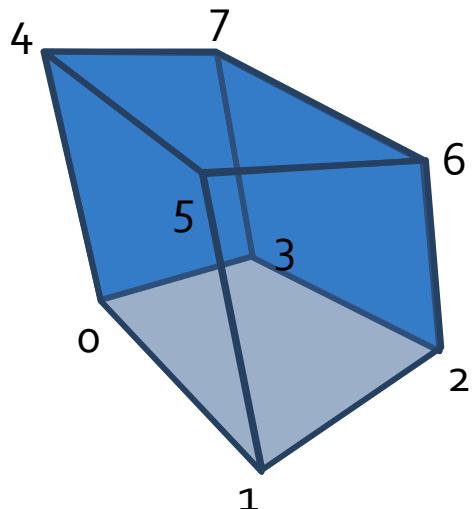
# Face-Vertex List (Indexed Format)



Vertex List	
A	(0,0,0)
B	(0,0,1)
C	(1,0,1)
D	(1,0,0)
E	(0,1,0)
F	(0,1,1)
G	(1,1,1)
H	(1,1,0)

Index List	
(A,B,C,D)	
(A,B,F,E)	
(B,C,G,F)	
(E,F,G,H)	
(A,D,H,E)	
(D,C,G,H)	

# Face-Vertex List (Indexed Format)

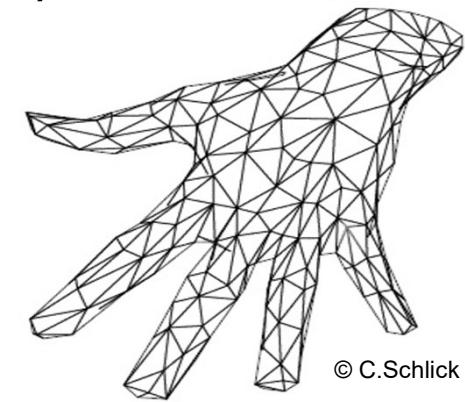


Vertex List	
0	(0,0,0)
1	(0,0,1)
2	(1,0,1)
3	(1,0,0)
4	(0,1,0)
5	(0,1,1)
6	(1,1,1)
7	(1,1,0)

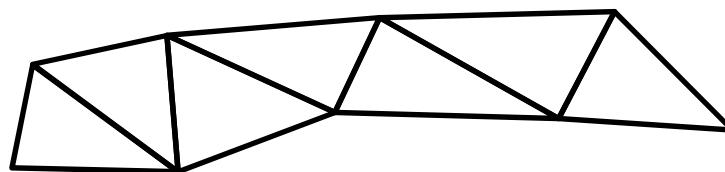
Index List	
(0,1,2,3)	
(0,1,5,4)	
(1,2,6,5)	
(4,5,6,7)	
(0,3,7,4)	
(3,2,6,7)	

# Triangle Meshes

- Most often used (directly rendered by hardware)
- Why triangles?
  - Simplest polygon
  - Always on a plane

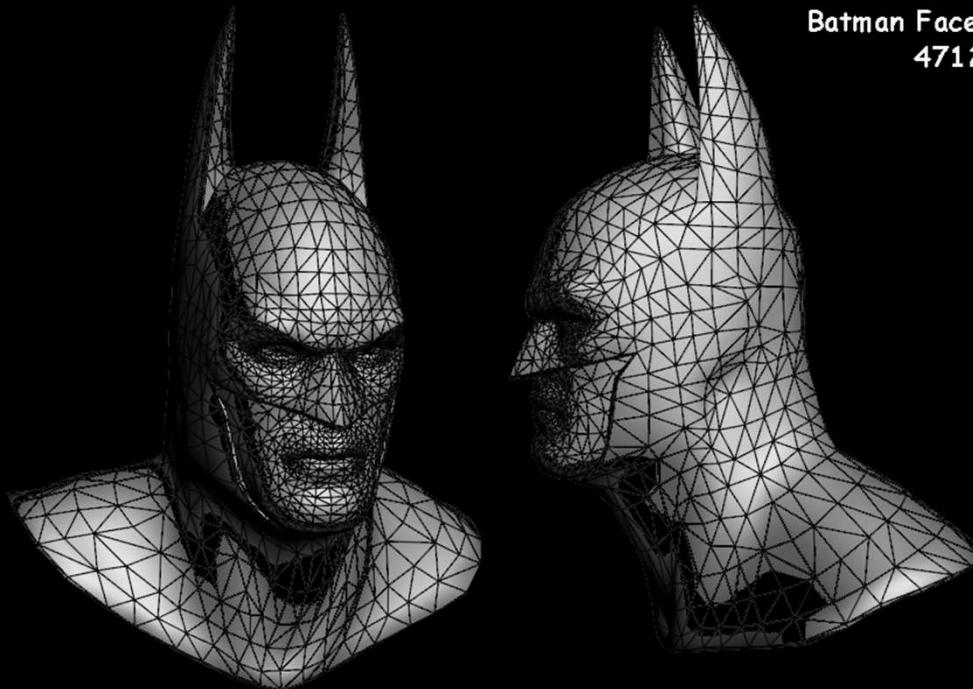


© C.Schlick



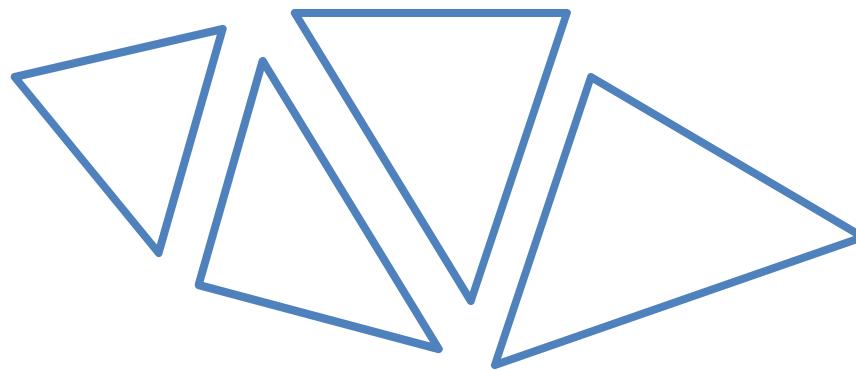
# Triangle Meshes

Batman Face Mesh  
4712 Tri's



# Polygon (Triangle) Soup

- Duplicate indices or even triangle coordinates



# Triangle Strips

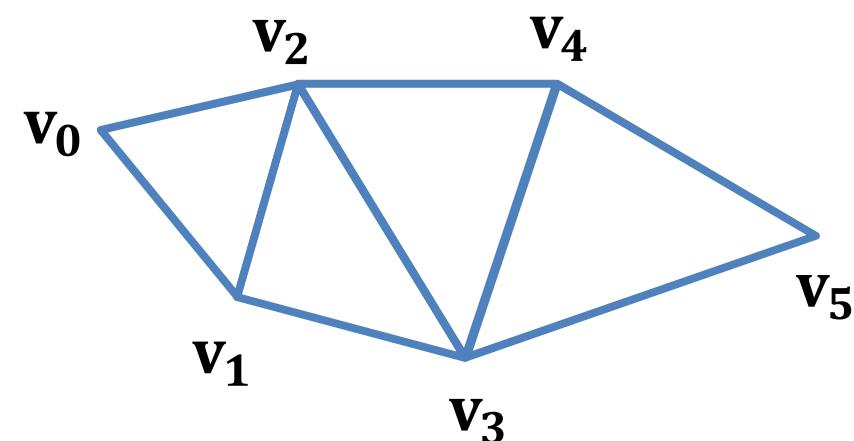
- Reduces redundancy by sharing vertices (indices)

Triangle 1:  $v_0, v_1, v_2$

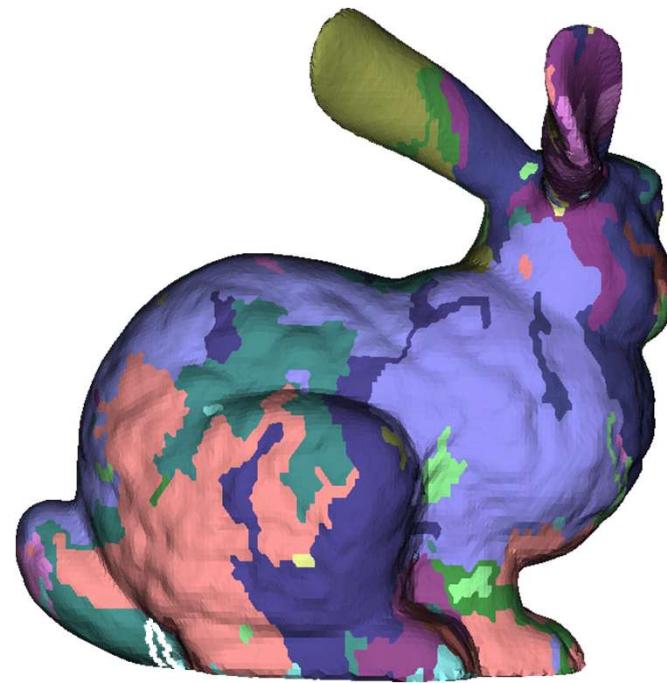
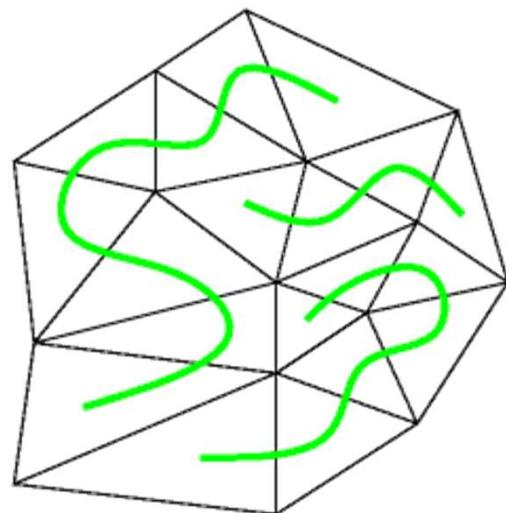
Triangle 2:  $v_1, v_2, v_3$

Triangle 3:  $v_2, v_3, v_4$

Triangle 4:  $v_3, v_4, v_5$

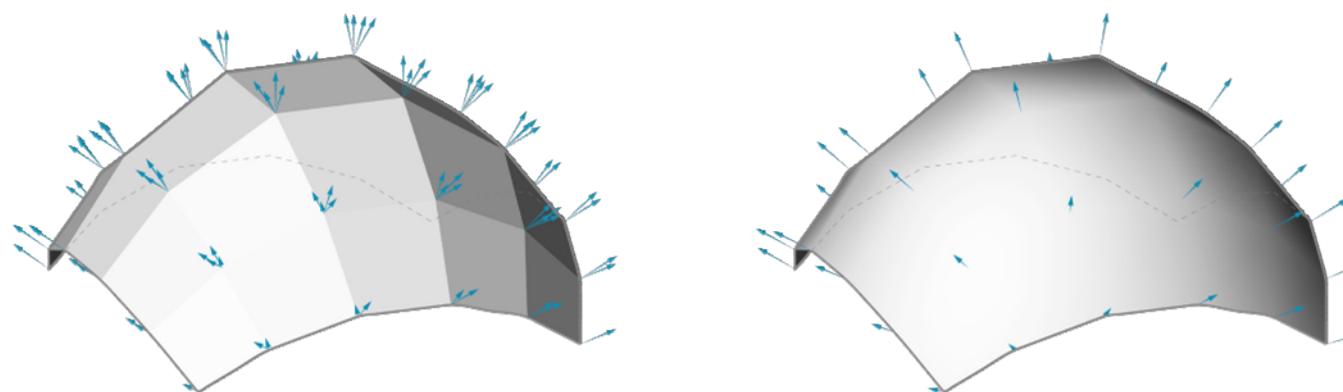


# Polygon(Triangle) Strip



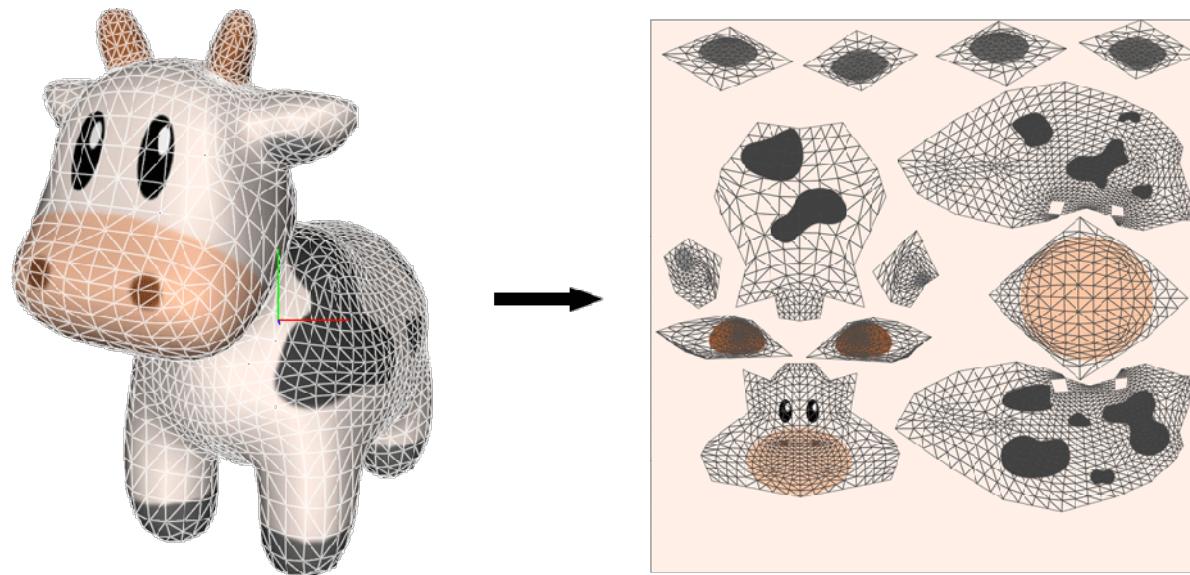
# (Vertex) Attributes of a Mesh

- Position
- Normal



# (Vertex) Attributes of a Mesh

- Texture coordinate



## (Vertex) Attributes of a Mesh

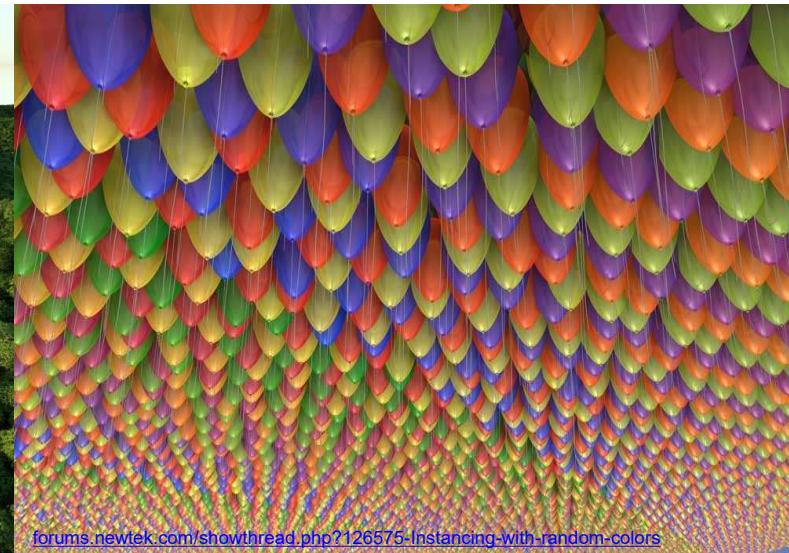
- Position
- Normal
- Texture coordinate
- Color
- ...

# Instancing

- Same object with differing transformations



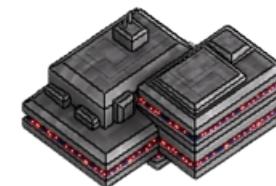
[thearender.com/site/index.php/features/engines/presto-gpu-cpu.html](http://thearender.com/site/index.php/features/engines/presto-gpu-cpu.html)



[forums.newtek.com/showthread.php?126575-Instancing-with-random-colors](http://forums.newtek.com/showthread.php?126575-Instancing-with-random-colors)

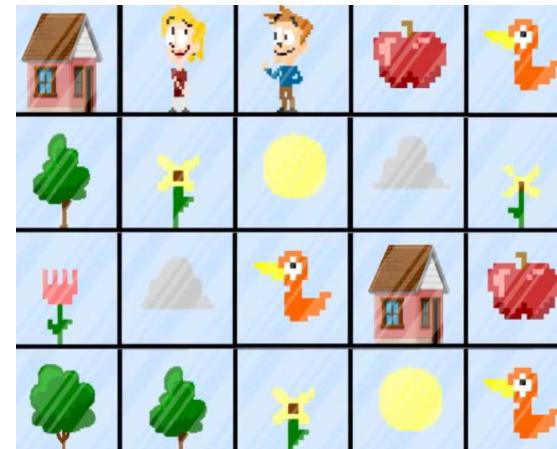
# Sprites

- Image or animation of object

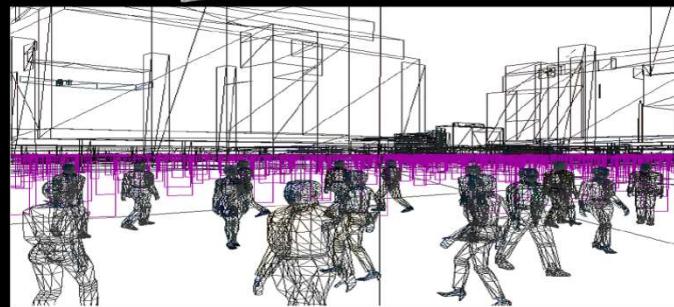


# Sprites

- Layer many to generate scene



# Impostor





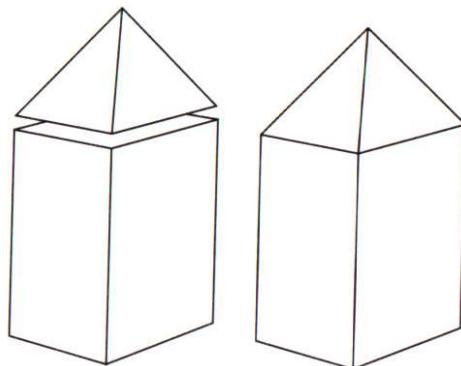
**Impostor**

# Impostor

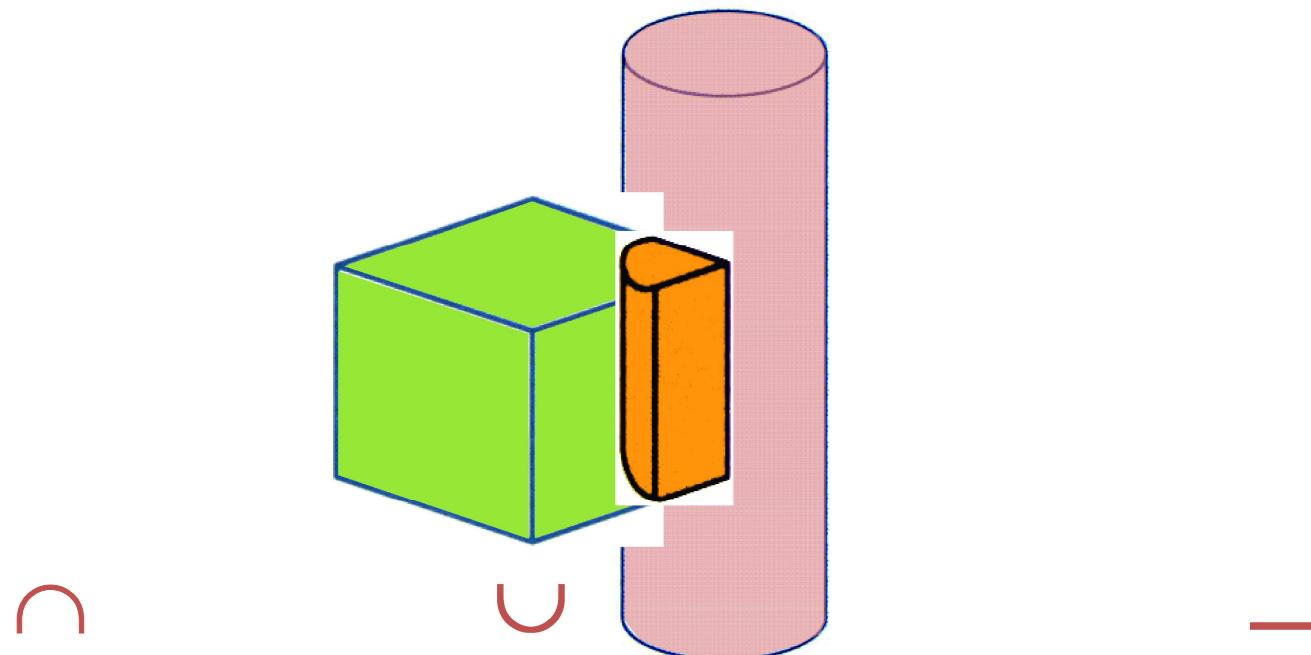


# Constructive Solid Geometry

- Constructive Solid Geometry (CSG)
  - boolean set operations on 3D objects
  - union, intersection, difference operation

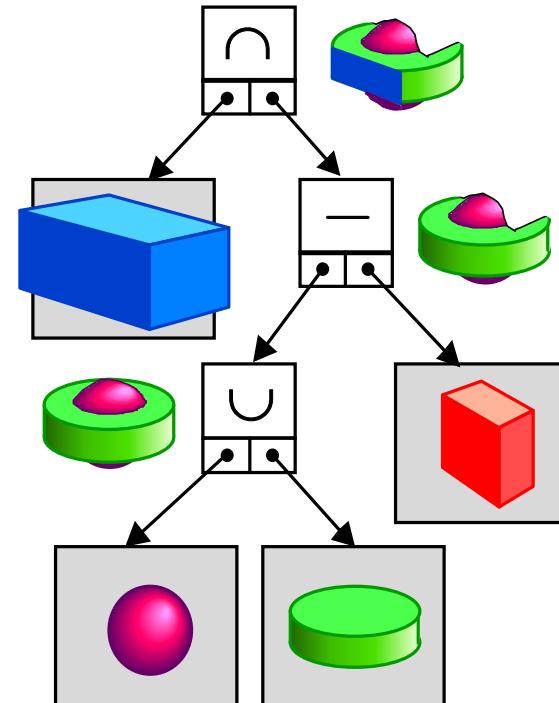


# CSG: Different Set Operations



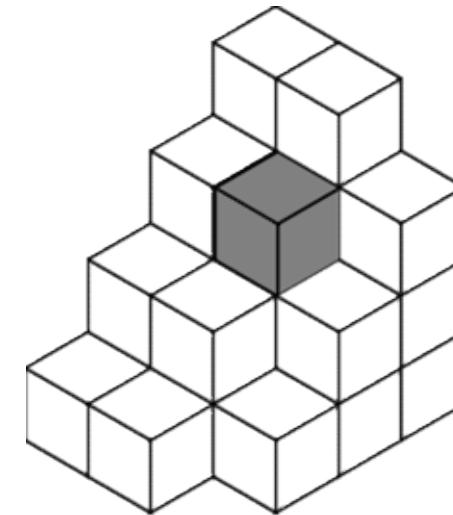
# CSG Data Structure

- Object assembled from simple solids with **set operations**
- data structure **binary tree**
- recursive evaluation

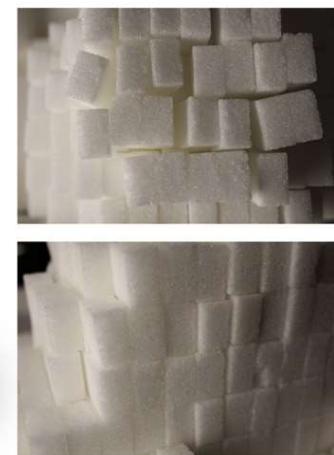


# Voxels

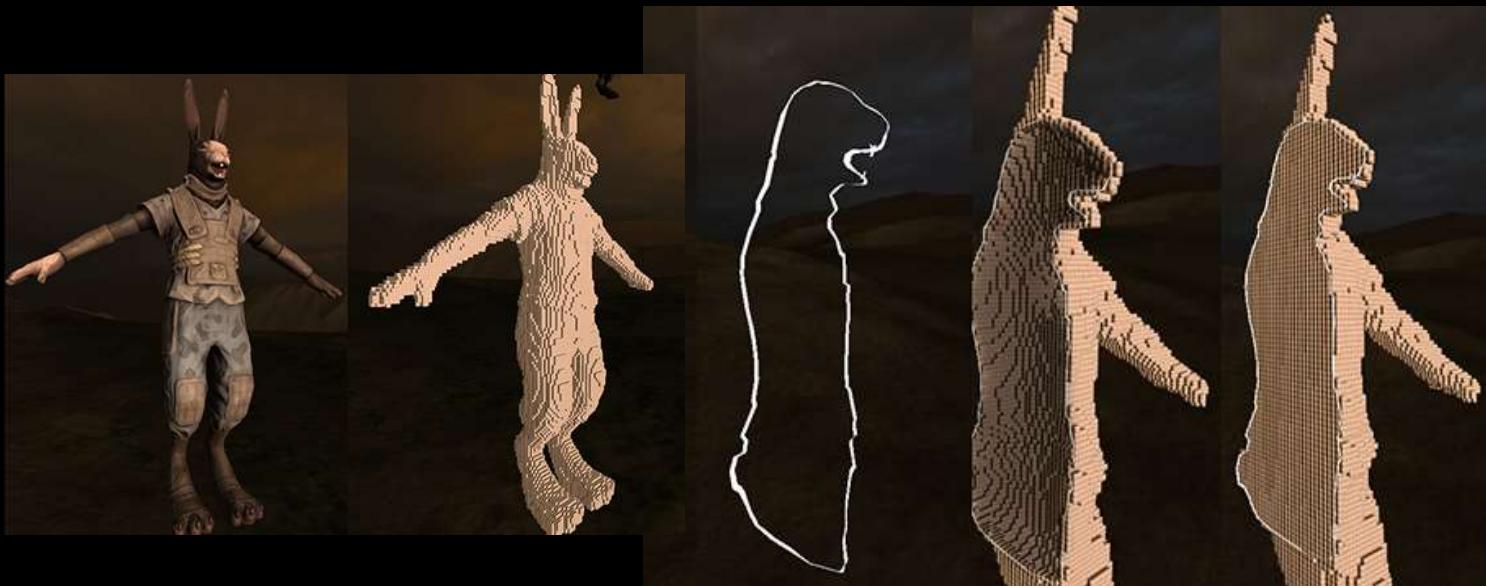
- Name is a combination of “volume” and “pixel”



# Real World



# Surface vs Solid Voxelization



# Voxels

- Not directly renderable by hardware
- Bad if lots of free space (memory!)
- But “fast” algorithms exists
  - Volume rendering
  - Ray casting
  - Marching cubes



# Procedural Modeling

- Use algorithm/rule to produce models







# Physically Based Modelling

- Procedural modeling with physically based rules



