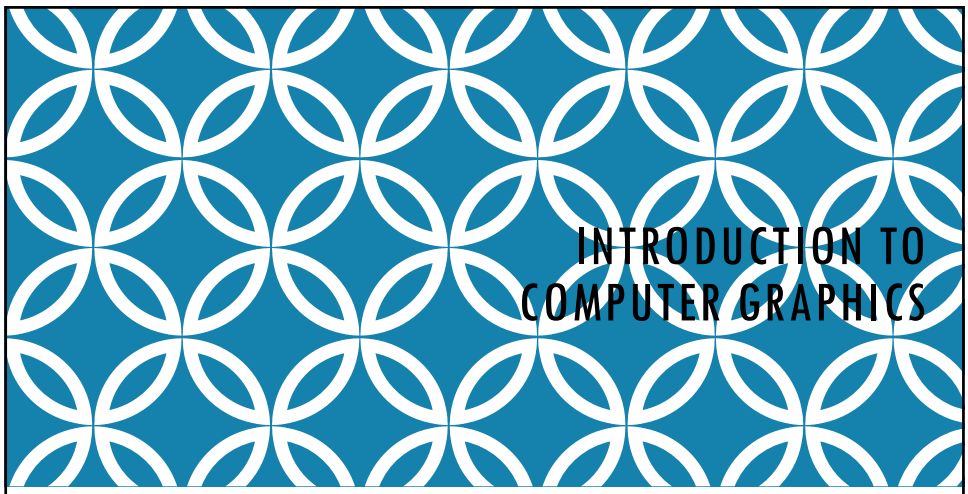


INTRODUCTION TO COMPUTER GRAPHICS

OpenGL Rendering Pipeline

COURSE GOAL

Learn the fundamental techniques involved in computer graphics

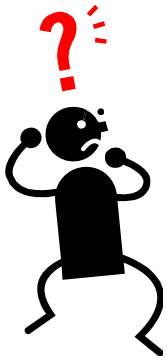
- Primitives
- Coordinate system & Transformations
- Projection
- Rasterization
- Clipping & Culling
- Texture mapping
- Lighting & Shading

Learn the standard computer graphics API

- OpenGL

RENDERING PIPELINE

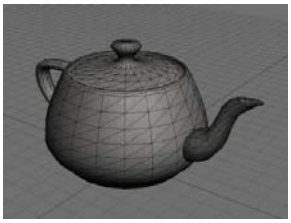
How do we display 3D scene on a computer screen?

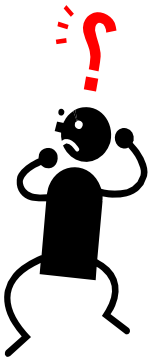


RENDERING PIPELINE

How do we display 3D scene on a (2D) computer screen?

1. Create a 3D scene
2. Choose a viewing angle & take a picture of the scene



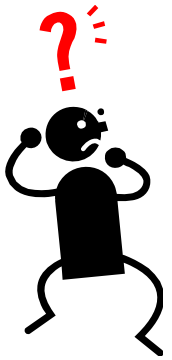


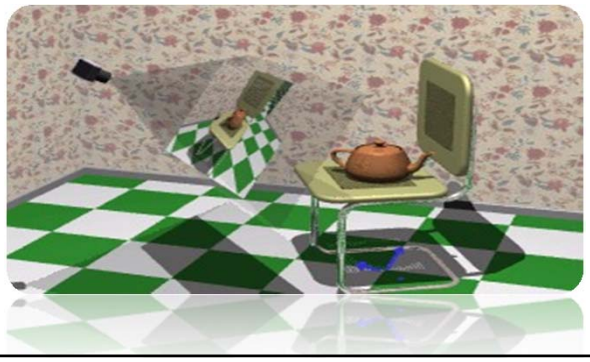
Objects & viewers

OBJECTS & VIEWERS

How do we display 3D scene on a (2D) computer screen?

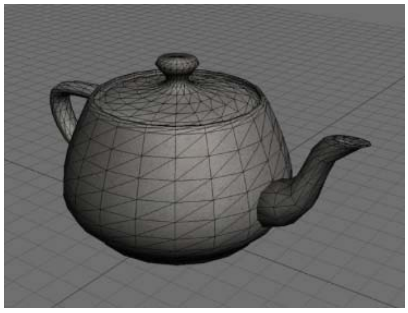
1. Create a 3D scene
2. Choose a viewing angle & take a picture of the scene





OBJECTS

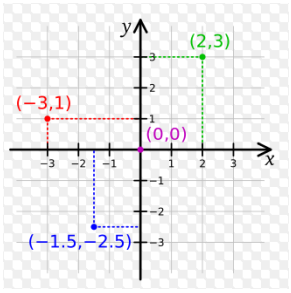
How do we describe an object in 3-Dimensional space?



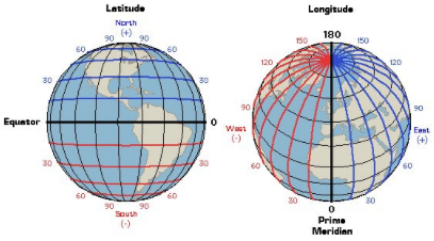
OBJECTS

In 2D

- we describe a point with (x,y)
- to uniquely determine the position of a point or other geometric element



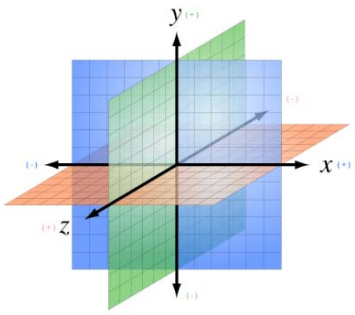
coordinate system



OBJECTS

In 3D

- we describe a vertex with (x,y,z)
  - E.g., Cartesian coordinate system
    - X-axis, Y-axis, Z-axis

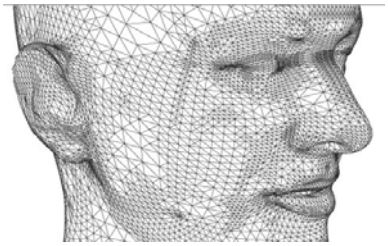


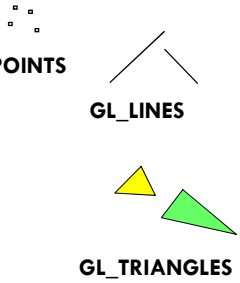
OpenGL Coordinate system

PRIMITIVES

How do we describe an object in 3-Dimensional space?

- Point (Vertex)
- Line
- Face (Triangles、~~Quad、Polygon~~)





OBJECT FILE

 # some text

- Line is a comment until the end of the line

v float float float

- A single vertex's geometric position in space. The first vertex listed in the file has index 1, and subsequent vertices are numbered sequentially.

f int int int …

- polygonal face.

vn float float float

- A normal. The first normal in the file is index 1, and subsequent normals are numbered sequentially.

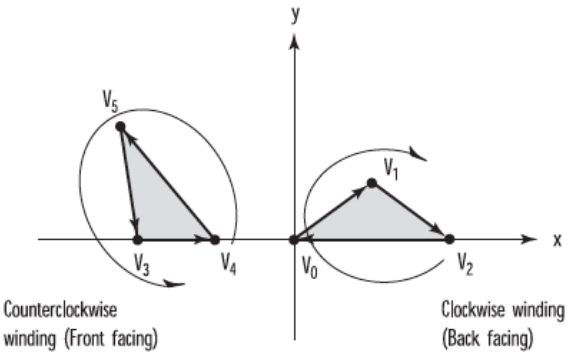
vt float float

- A texture coordinate. The first texture coordinate in the file is index 1, and subsequent textures are numbered sequentially.

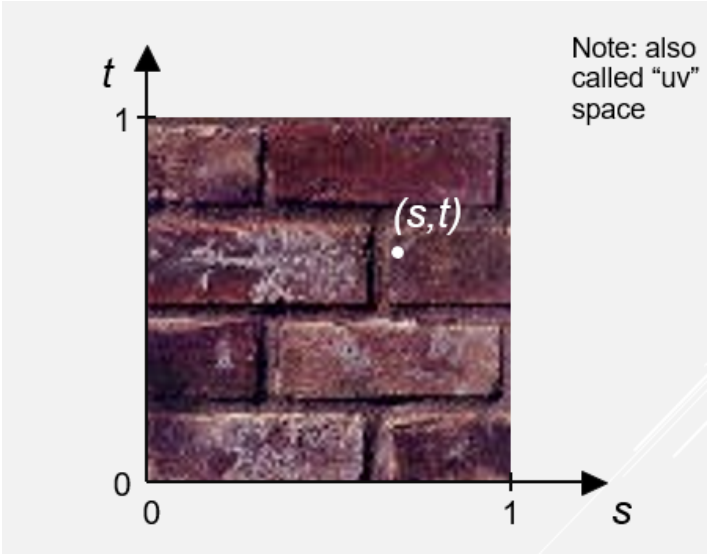
```
v 1 1 1
v 1 1 -1
v 1 -1 1
v 1 -1 -1
v -1 1 1
v -1 1 -1
v -1 -1 1
v -1 -1 -1
f 1 3 4 2
f 5 7 8 6
f 1 5 6 2
f 3 7 8 4
f 1 5 7 3
f 2 6 8 4
```

VERTEX WINDING MATTERS…

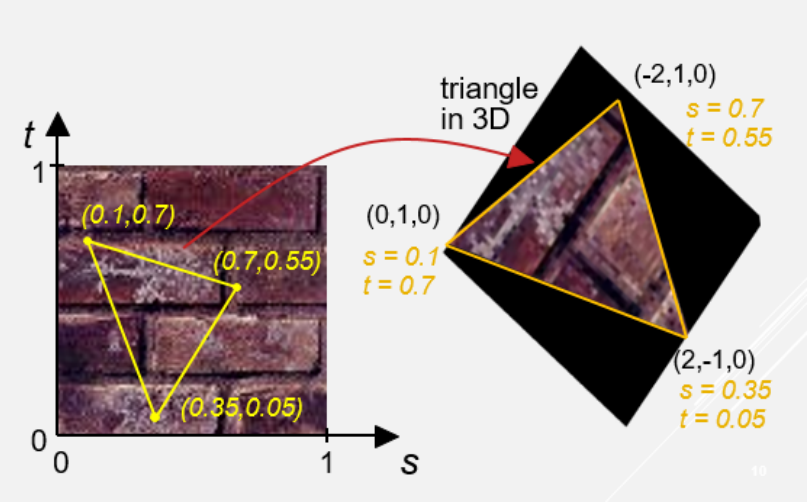
Face Culling
- Front face
- Back face



THE "ST" COORDINATE SYSTEM



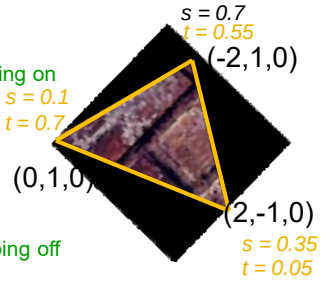
TEXTURE MAPPING: KEY SLIDE



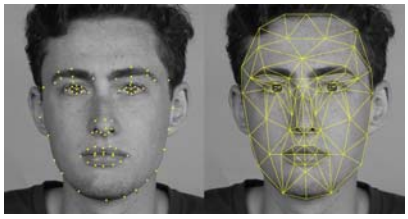
Specifying texture coordinates in OpenGL

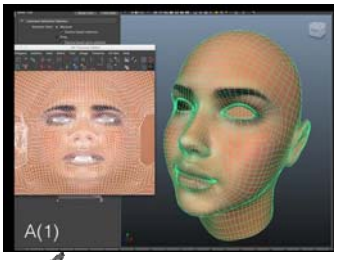
- Use glTexCoord2f(s,t)
- State machine: Texture coordinates remain valid until you change them

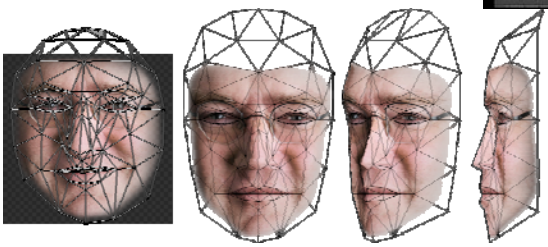
```
glEnable(GL_TEXTURE_2D); // turn texture mapping on
glBegin(GL_TRIANGLES);
 glTexCoord2f(0.35,0.05); glVertex3f(2.0,-1.0,0.0);
 glTexCoord2f(0.7,0.55); glVertex3f(-2.0,1.0,0.0);
 glTexCoord2f(0.1,0.7); glVertex3f(0.0,1.0,0.0);
glEnd();
glDisable(GL_TEXTURE_2D); // turn texture mapping off
```



FACE MODEL



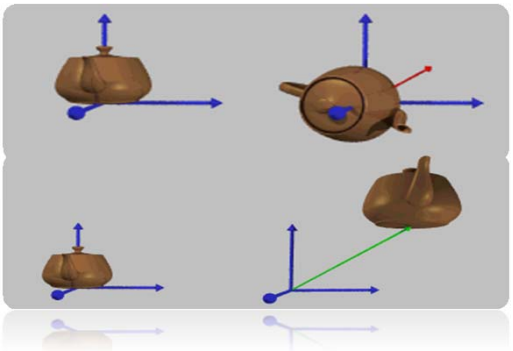




TRANSFORMATION

To determine the location, orientation of a 3D object

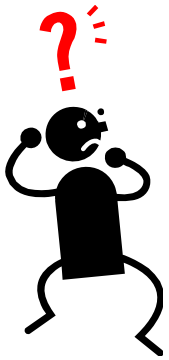
- Translation
- Rotation
- Scale

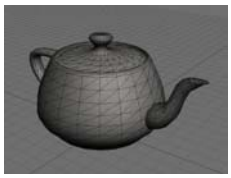


OBJECTS & VIEWERS

How do we display 3D scene on a (2D) computer screen?

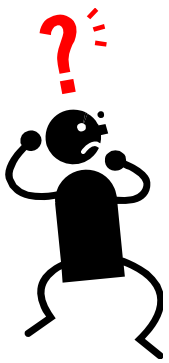
1. Create a 3D scene
2. Choose a viewing angle & take a picture of the scene



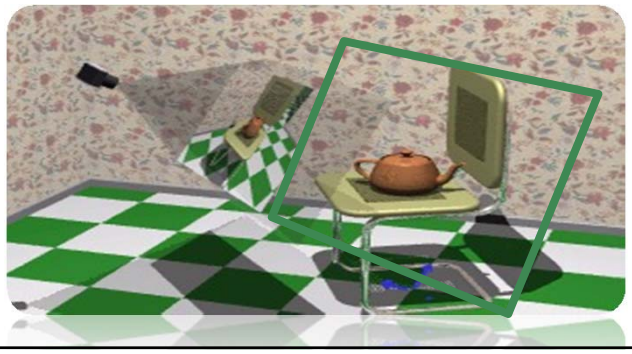


OBJECT & VIEWER

- How do we display 3D scene on a (2D) computer screen?
  1. Create a 3D scene
  2. Choose a viewing angle & take a picture of the scene



Projecting 3D objects to a 2D plane



OBJECTS & VIEWERS

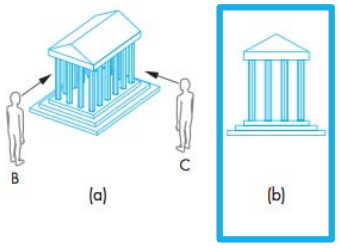
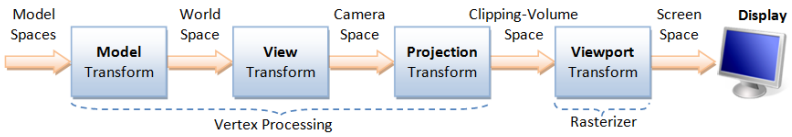
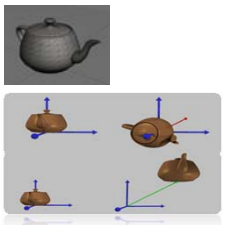


Image seen by three different viewers. (a) A's view. (b) B's view. (c) C's view.

COORDINATES TRANSFORMATION



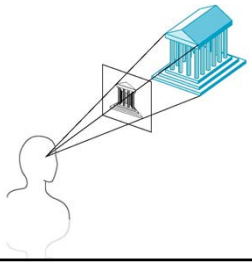
Coordinates Transform Pipeline

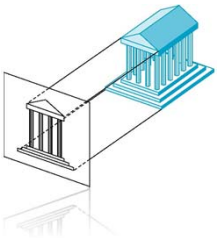


PROJECTION

The process that combines the 3D viewer with the 3D objects to produce the 2D image

- Perspective projections (透視投影)
  - all projectors meet at the center of projection
- Parallel projection(平行投影)
  - projectors are parallel, center of projection is replaced by a direction of projection

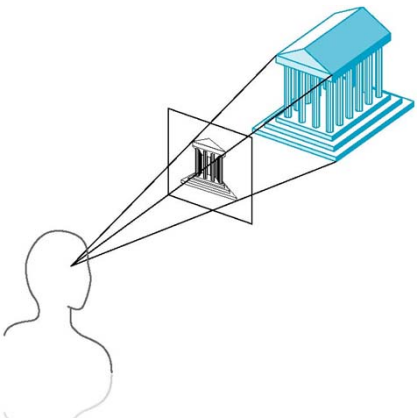




PERSPECTIVE PROJECTION

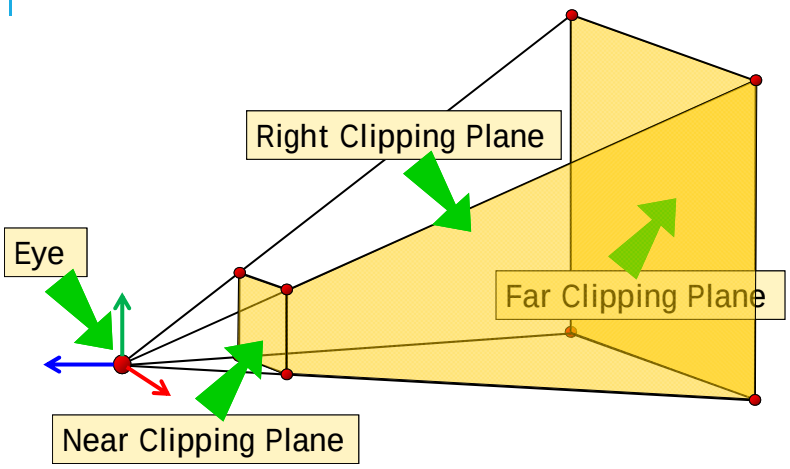
特性: 會隨著物體的距離遠近改變外觀大小

- 因為較接近真實所以較常被使用



all projectors meet at the center of projection

PERSPECTIVE PROJECTION

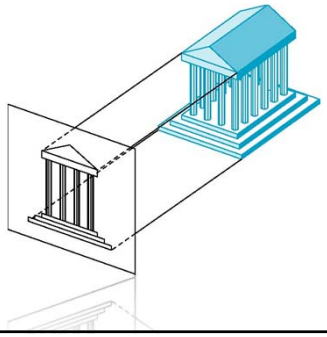


ORTHOGRAPHIC PROJECTION

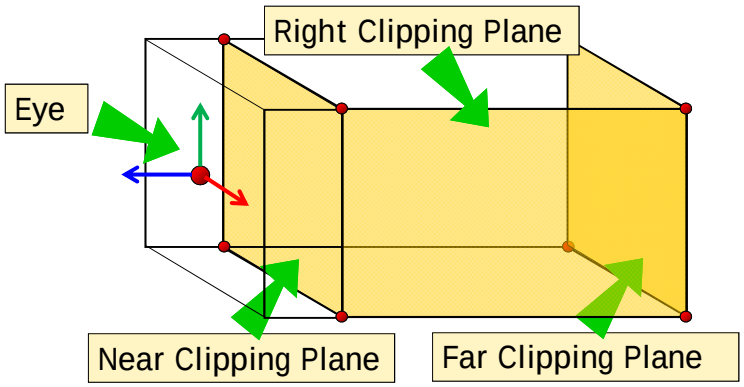
不管物體距離遠近，其尺寸都不變

- E.g, 視窗中的文字說明，要保持相同大小

projectors are parallel, center of projection is replaced by a direction of projection

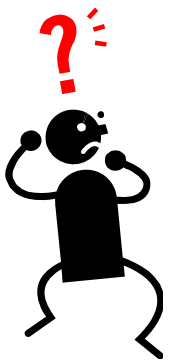


ORTHOGRAPHIC PROJECTION

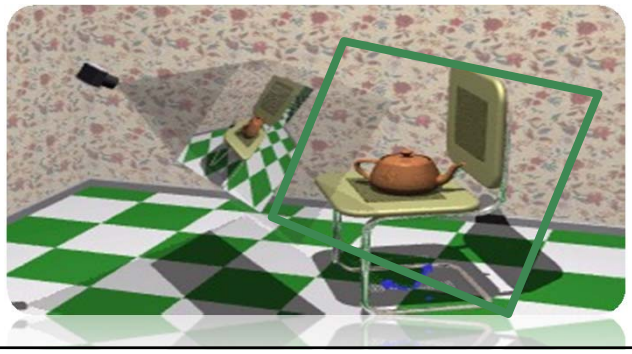


PROJECTION

Project to a 2D plane and then do rasterization



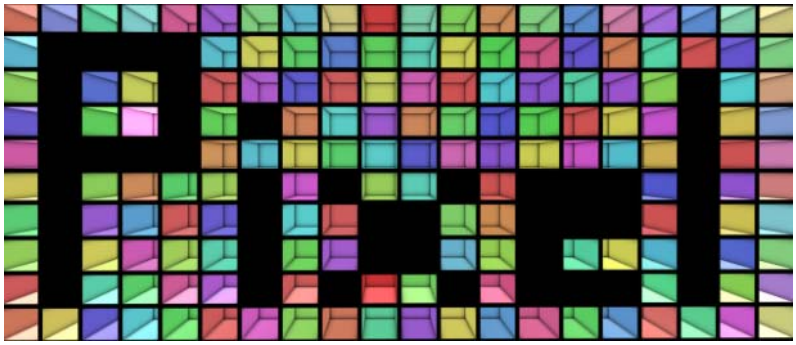
Projecting 3D objects to a 2D plane



FRAME BUFFER

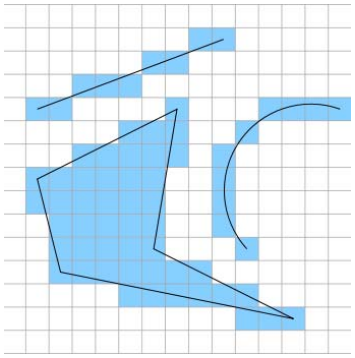
Display on the screen

- Pixel



RASTERIZATION

- 在畫線條時，將每個頂點/線條之間的像素(pixels)填滿

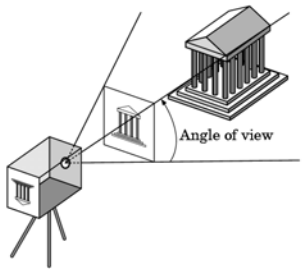


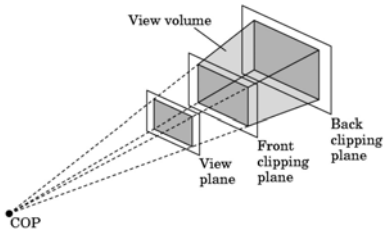
BEFORE RASTERIZATION…

Clipping

Just as a real camera cannot "see" the whole world, the virtual camera can only see part of the world or object space

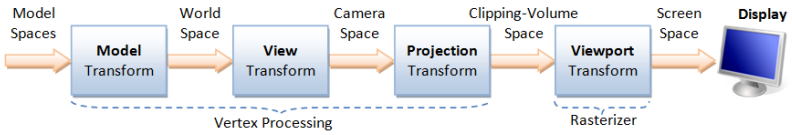
- Objects that are not within this volume are said to be *clipped* out of the scene



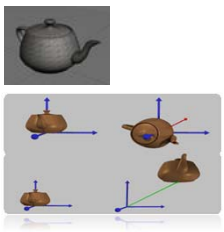


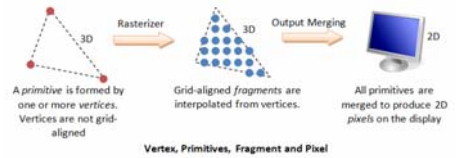
3
4

COORDINATES TRANSFORMATION



Coordinates Transform Pipeline

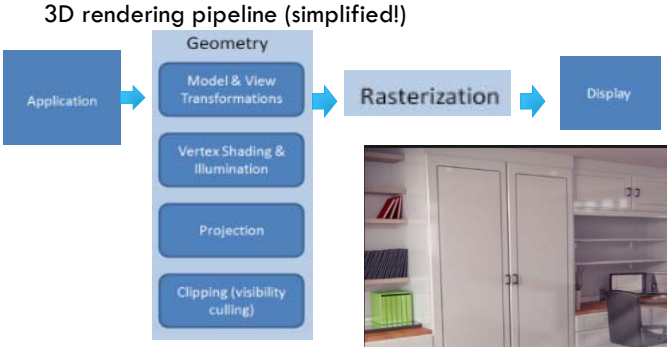




Vertex, Primitives, Fragment and Pixel

COMPUTER GRAPHICS INTRODUCTION

3D rendering pipeline (simplified!)



COMPUTER GRAPHICS INTRODUCTION

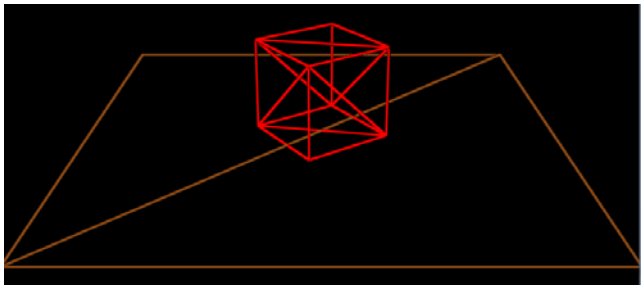
Texture mapping/ lighting/Rendering

COMPUTER GRAPHICS INTRODUCTION

OpenGL example

OPENGL EXAMPLE

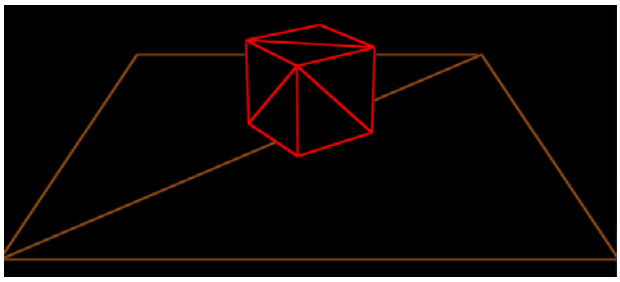
- Wireframe



OPENGL EXAMPLE

Back face culling

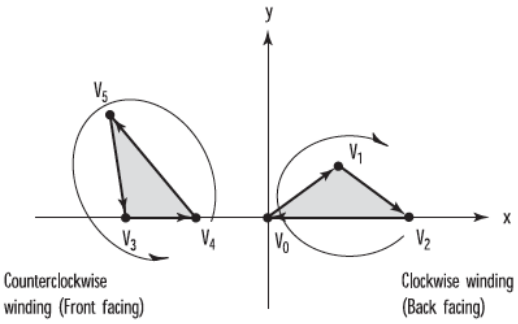
- 接著透過壓按空白鍵來切換，可發現該block後方消失了，其開始逼近真實的3D實體感覺



OPENGL EXAMPLE

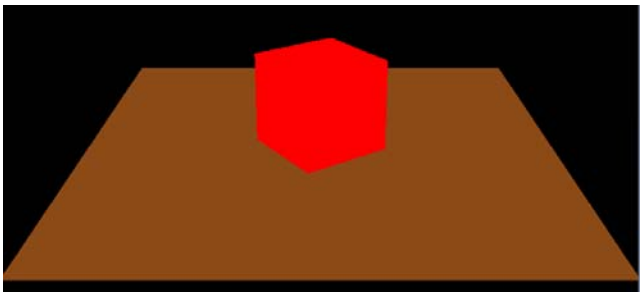
Culling

- Front face
- Back face



OPENGL EXAMPLE

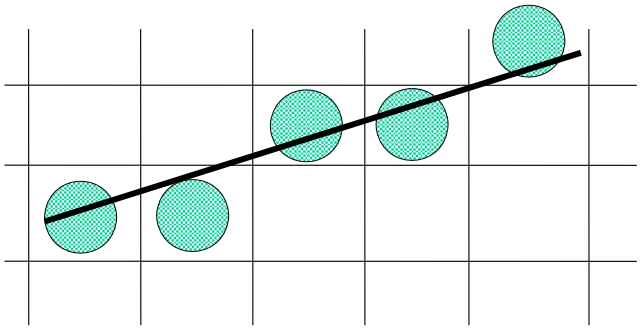
拿掉了原先的線條，改採用純色的三角面組成這Block，卻沒使該物體顯得逼真



Superbible 5th edition sample project

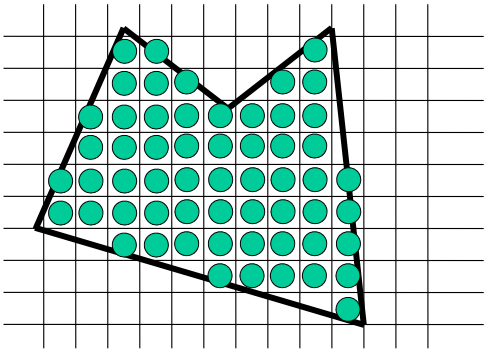
RASTERIZING LINES

Given two endpoints, $(x_0, y_0), (x_1, y_1)$
find the pixels that make up the line.



RASTERIZING POLYGONS

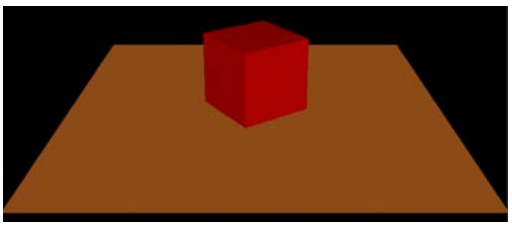
Given a set of vertices and edges,
find the pixels that fill the polygon.



OPENGL EXAMPLE

Shading(明暗度)

利用明暗程度的差異(打光技巧)來使原先的Block不同的面有了色差

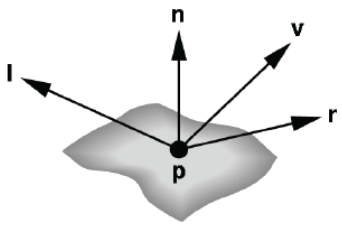


PHONG REFLECTION MODEL

The Phong model uses the four vectors, (l,v,n,r)

- to calculate a color for an arbitrary point p on a surface

Basic inputs are material properties and l, n, v



l = unit vector to light source
v = unit vector to viewer
n = surface normal
r = reflection of l at p
(determined by l and n)

OPENGL EXAMPLE

Texture Mapping(貼圖投影)

- 將一張圖案投影到三角形或多邊形上, 真實性提升

OPENGL EXAMPLE

Blending(混合)

- 此圖的反射效果, 預先複製一個顛倒的Block，加上半透明的效果出來，再與木版做混合