CG – T2
Introduction to CG

L:CC, MI:ERSI

Miguel Tavares Coimbra

(course and slides designed by Verónica Costa Orvalho)

U. PORTO FC

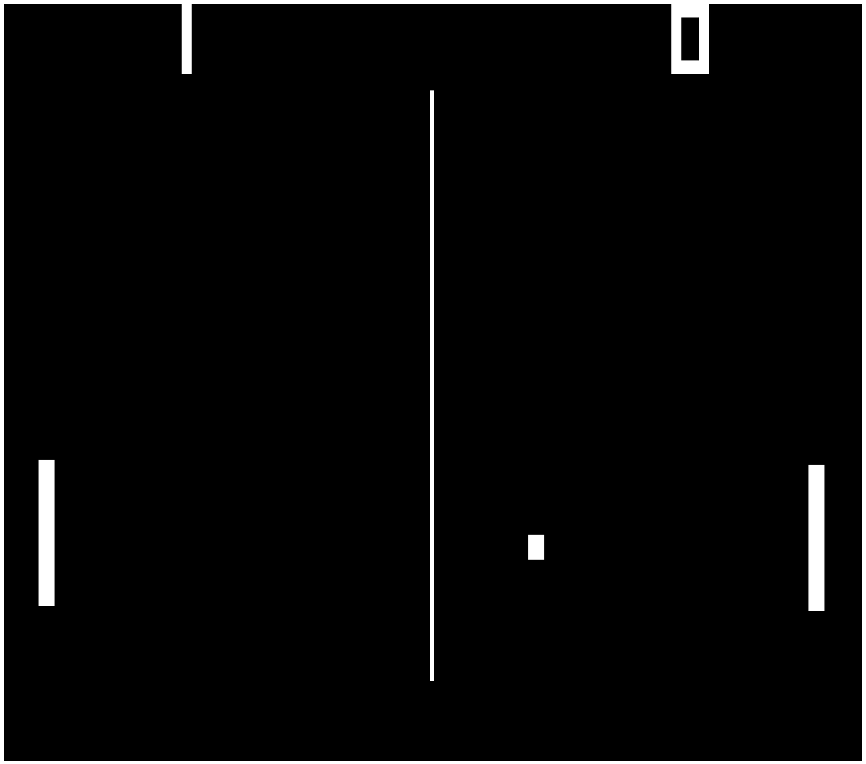
the beginning: 2D

1st CG displayed 2D graphics (flat lines, circles, polygons)

Simple arcade games: Pong

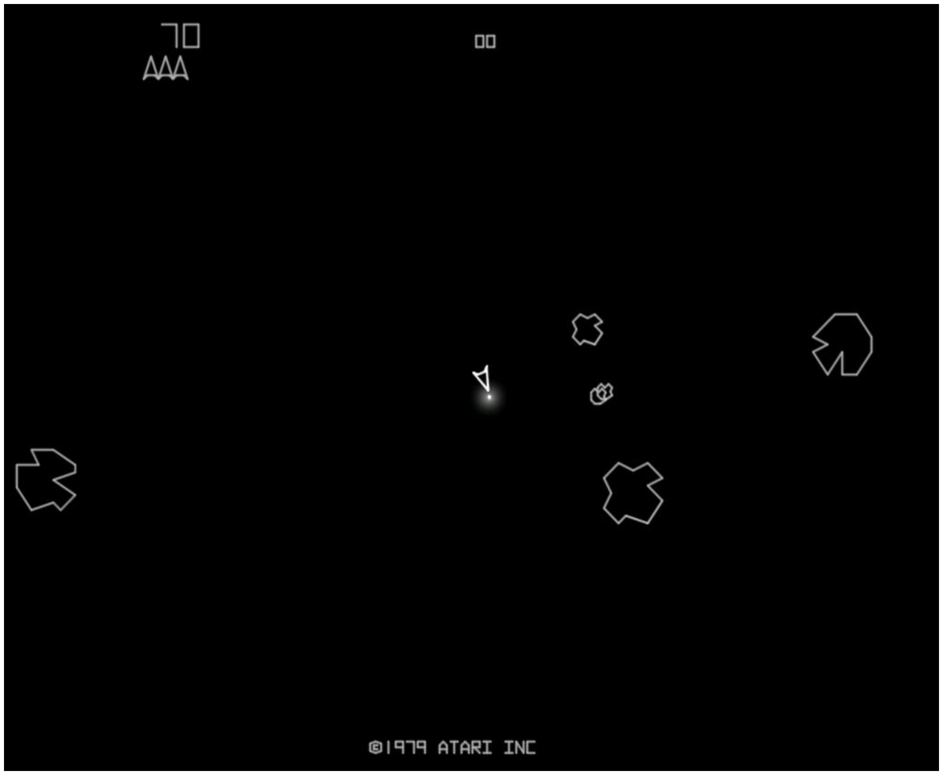
Real-time: CG that were animated

pong



1972

lunar lander

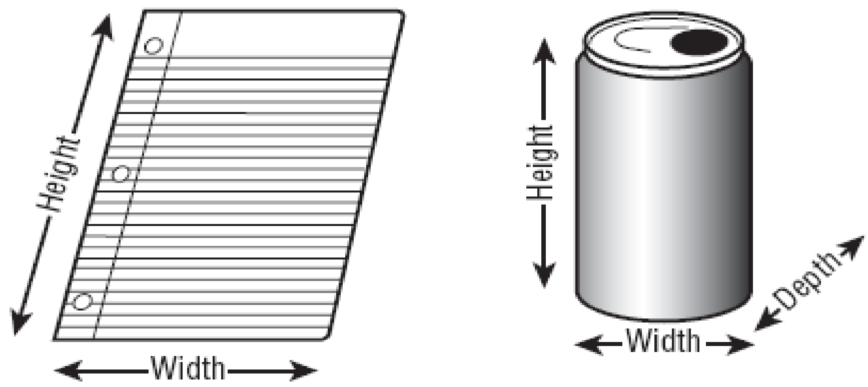


1979

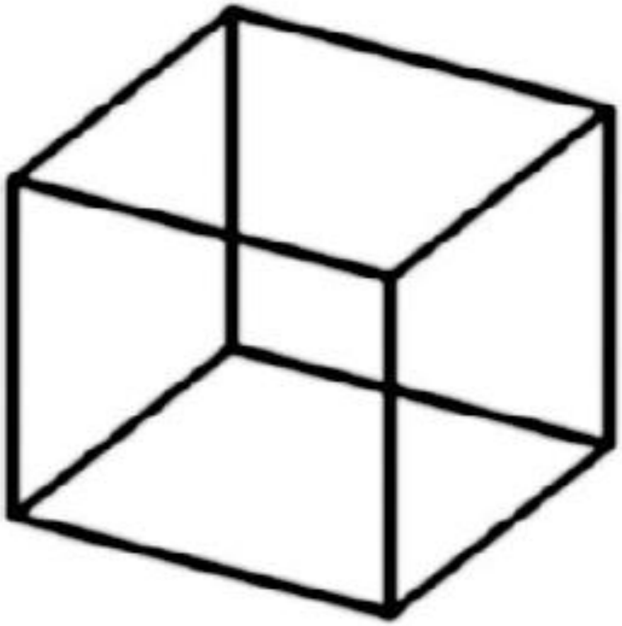
why and how 3D?

3D has 3 dimensions of meassurement

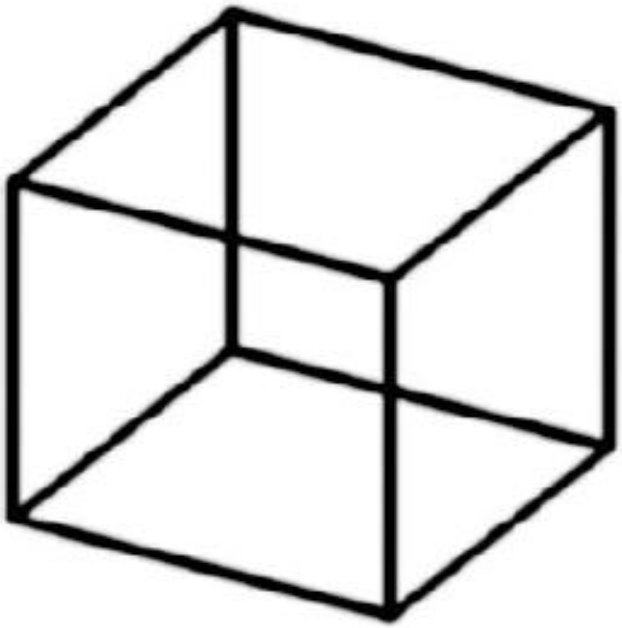
width, height and depth



what is this?



what is this?



This is a 2D image of a **drawing** of a cube

3D illusion

3D computer graphics are actually 2D images on a flat screen

what makes the cube look 3D ?

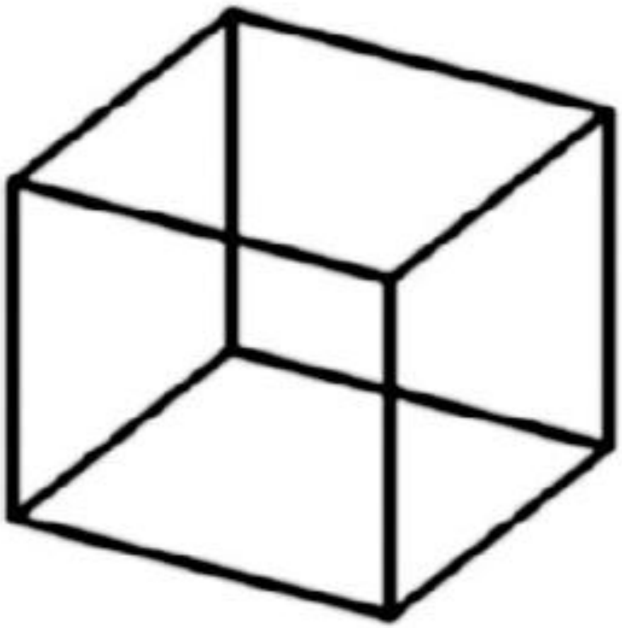
3D illusion

3D computer graphics are actually 2D images on a flat screen

<u>what makes the cube look 3D ?</u>
is **perspective** or the angle between the lines (illusion)

3D illusion

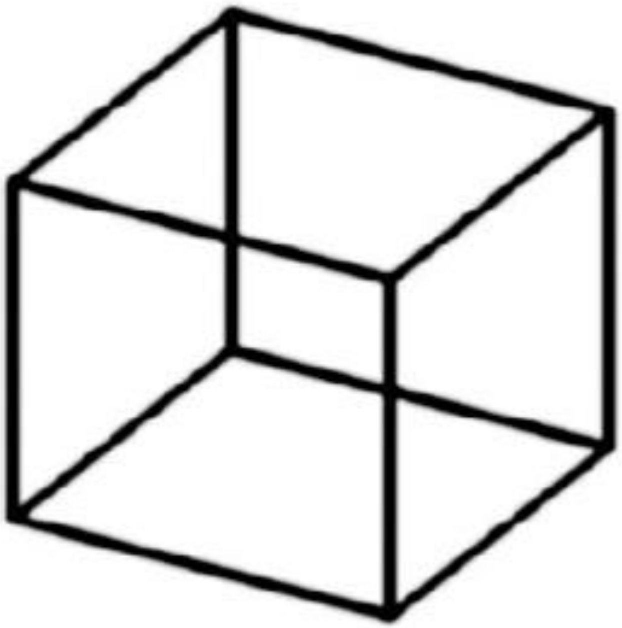
perspective is not enough

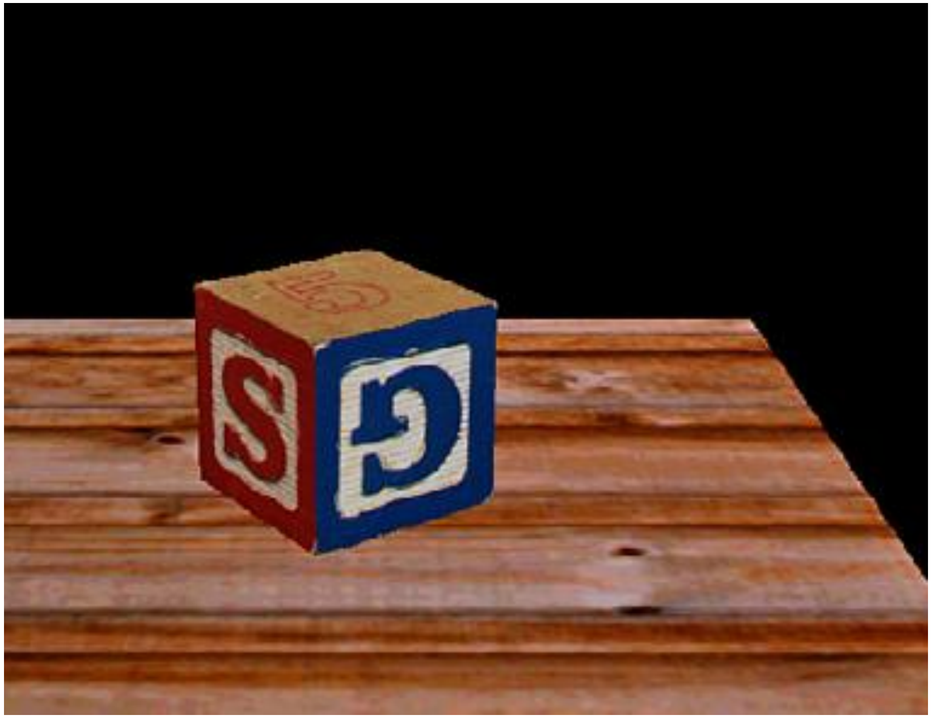


what else?

3D illusion

perspective is not enough

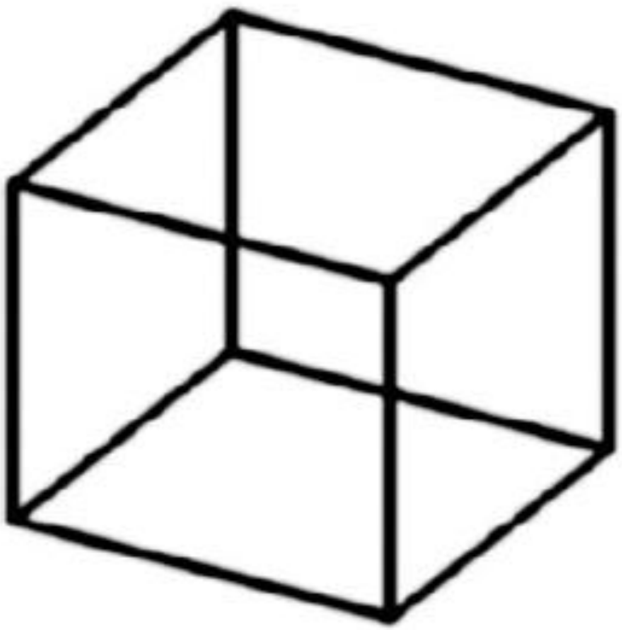


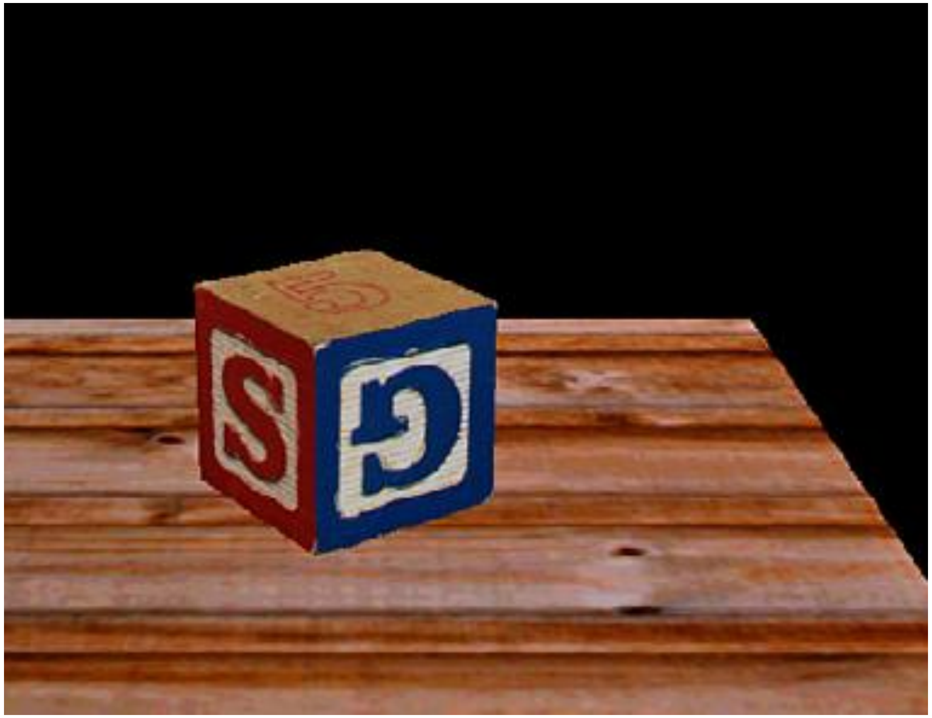


Ref. R. S. Wright, et al. OpenGL SuperBible, 5th edition, Addison-Wesley, July 2010

3D illusion

perspective is not enough



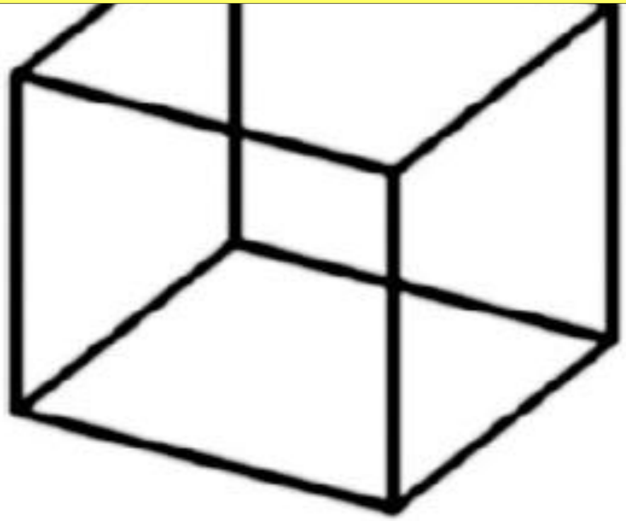


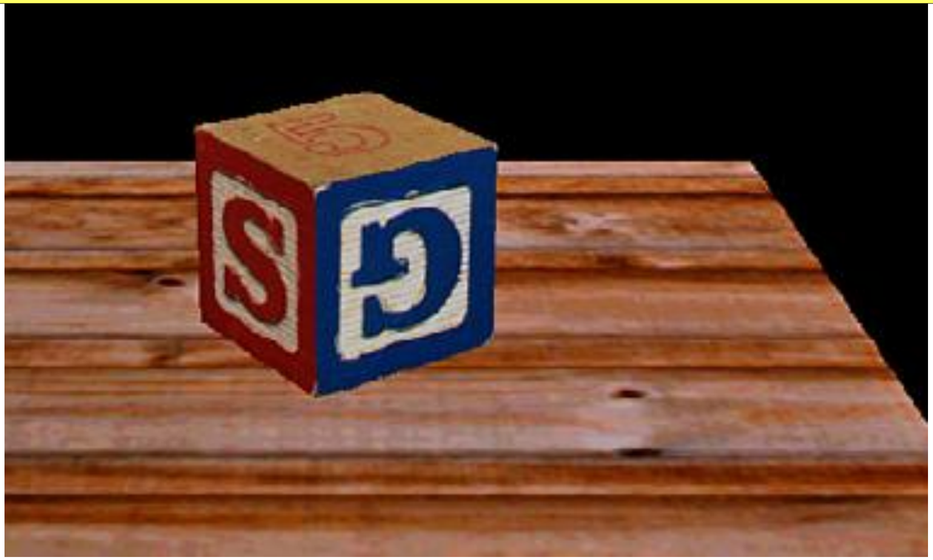
color changes, textures, shading, color intensity....

3D illusion

perspective is not enough

perception of a 3D image

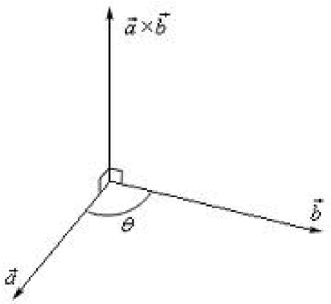


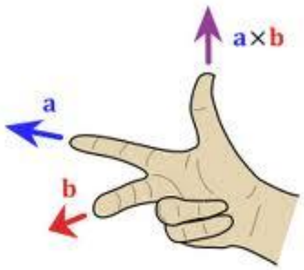


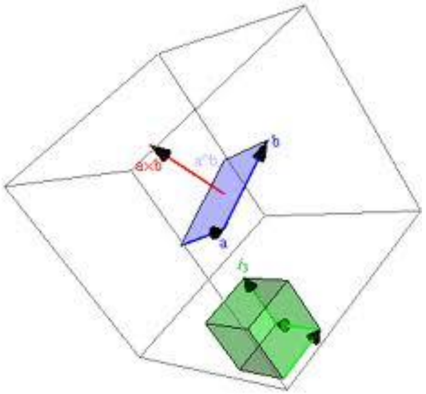
color changes, textures, shading, color intensity....

math you might need today

cross product

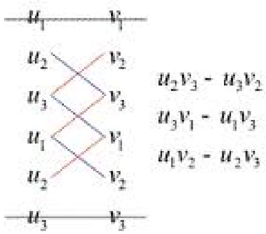






not commutative,
order is important

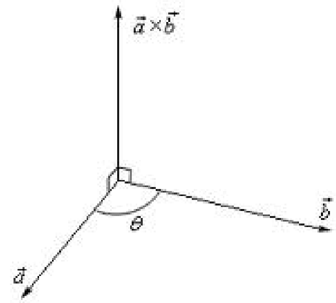
$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - b_y a_z \\ a_z b_x - b_z a_x \\ a_x b_y - b_x a_y \end{pmatrix}$$



math you might need today

cross product

```
Vector crossproduct(Vector &v)
{
    Vector vector;
        vector.x = (y * v.z) - (z * v.y);
        vector.y = (z * v.x) - (x * v.z);
        vector.z = (x * v.y) - (y * v.x);
    return vector;
}
```



$$\begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \times \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} a_y b_z - b_y a_z \\ a_z b_x - b_z a_x \\ a_x b_y - b_x a_y \end{pmatrix}$$

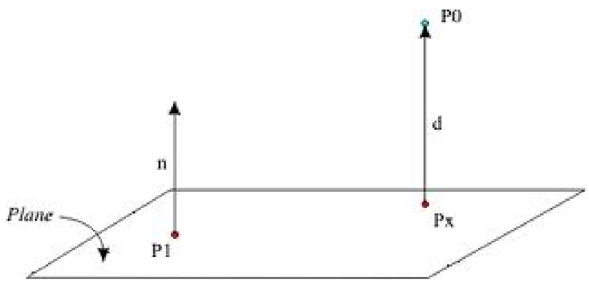
math you might need today

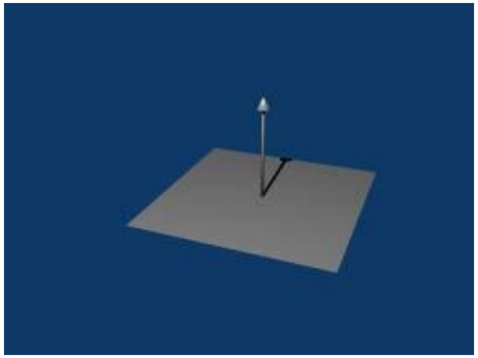
the plane equation

A plane is defined as:

> a set of points perpendicular to a normal vector **n = (a,b,c)**
> that also contains the point **P0=(x0,y0,z0)**
> if a point **P** lies on the plane, then vector **v=P-P0** also lies on the plane
> then **n.v=0 (dot product)**

n.v => (x * v.x)+(y * v.y)+(z * v.z);





$$A.B = \|A\|\|B\|\cos\theta$$

a(x-x0)+b(y-y0)+c(z-z0)=0

math you might need today

more about vectors

magnitude (length):

|a| = sqrt((ax * ax) + (ay * ay) + (az * az))

unit Vector – normalization

1 calculate its length, then,

2 divide each of its (xyz) components by its length.

x = ax/|a|
y = ay/|a|
z = az/|a|

Values between [0,1]

magnitued = sqrt(9 + 1 + 4) = 3.742

x = 3.0 / 3.742 = **0.802**
y = 1.0 / 3.742 = **0.267**
z = 2.0 / 3.742 = **0.534**

http://www.fundza.com/vectors/normalize/index.html

terms you must know: the beginning

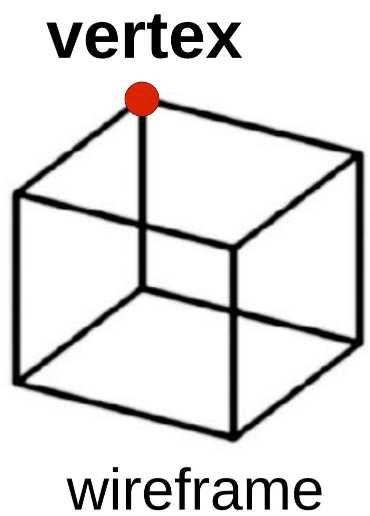
vertex: 3D point in space

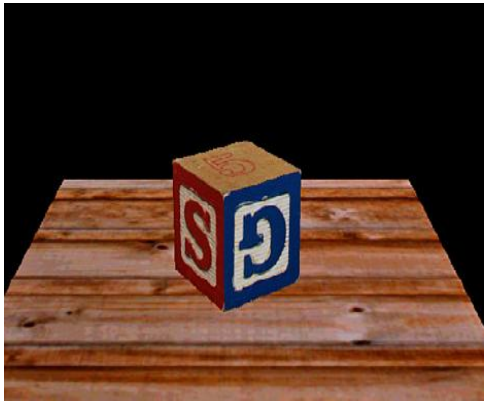
transformation matrix: move vertex around in space

projection matrix: turn 3D coordinates into 2D screen coordinates

transforming points around and creating lines between them we create the 3D illusion

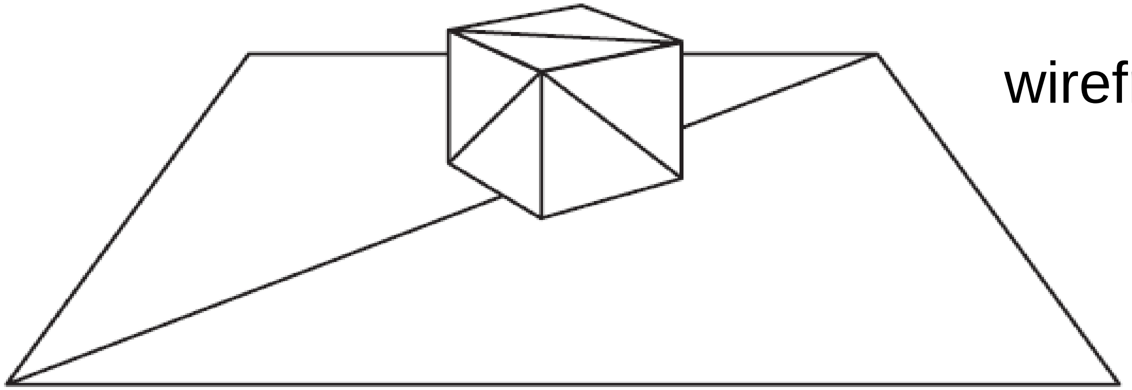
rasterization: drawing or filling the pixels between each vertex





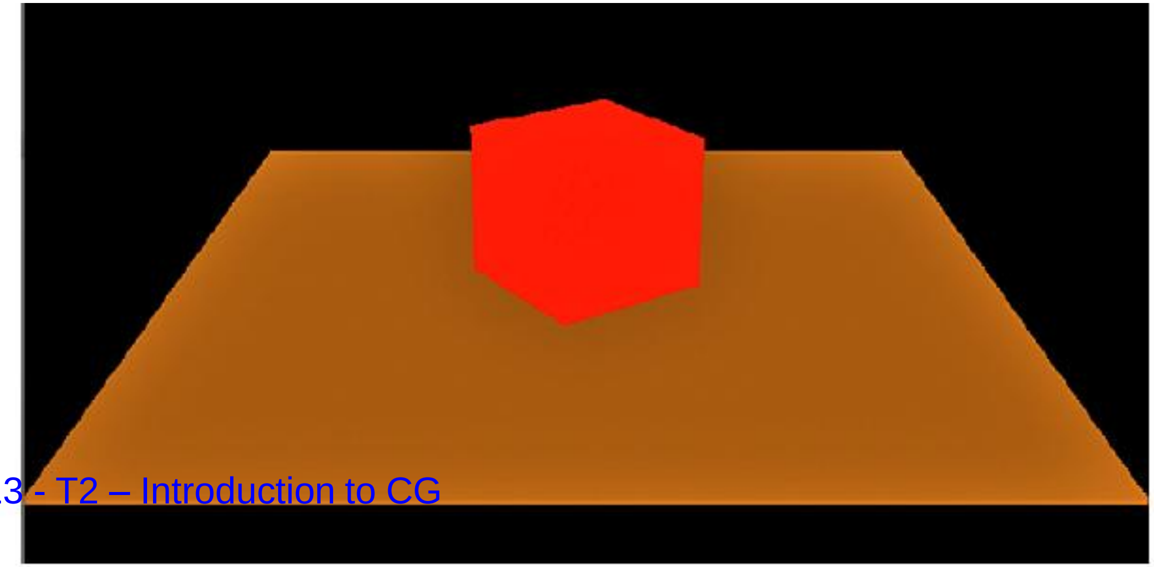
terms you must know: the beginning

rasterization:



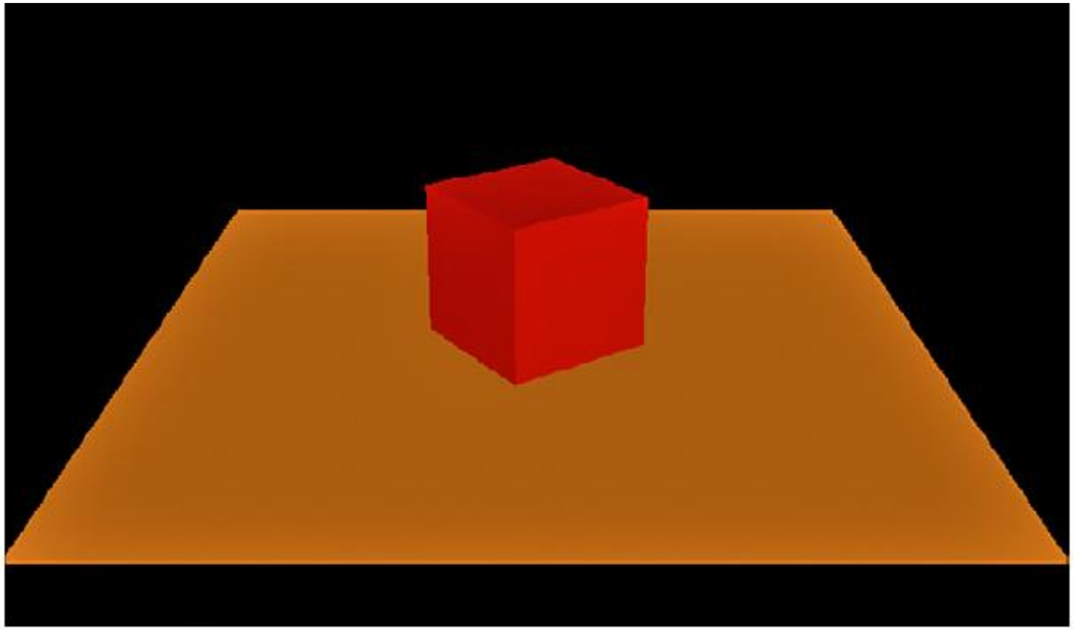
wireframe

filling with colors



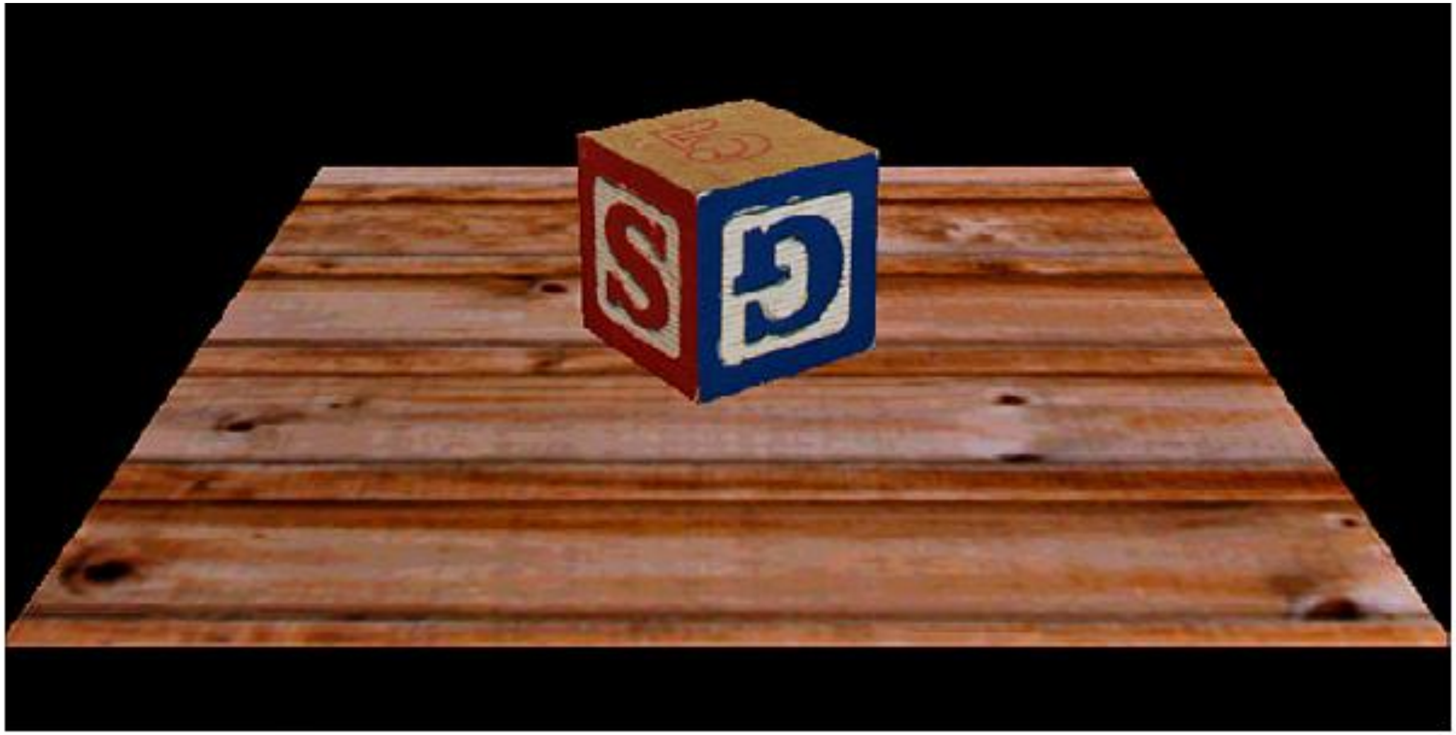
terms you must know: the beginning

Shading: varing the color values across the surface (between vertices). Create the <u>**effect**</u> of <u>**light shining**</u> on a red cube



terms you must know: the beginning

texture mapping: a picture that we map to the surface of a triangle or polygon. A texture can simulate an effect that could take thousands of triangles.



terms you must know: the beginning

blending: allows mixing different colors together. e.g. create reflections.

everything comes together

transformation + shading + texture + blending

Summary

- We will try create the illusion of a 3D world using a 2D screen
- Humans mentally build their 3D illusion based on two 2D images (but now we only have one…)
- We need maths
- We need structure: transformation, shading, texture, blending