

# Computer Game Technologies

## 3D Graphics Concepts

# Java 3D and Games Programming

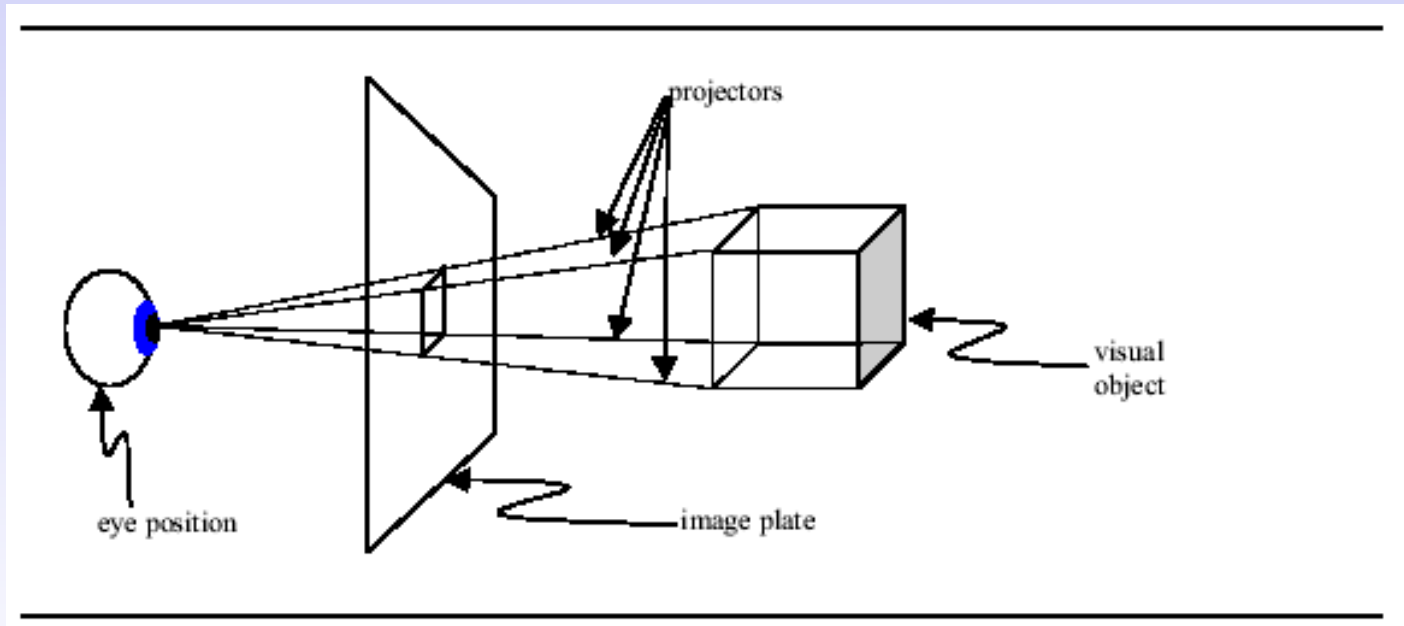
- Introduction to 3D graphics concepts
  - 3D Coordinates
- 3D graphics programming
  - Low and high level approaches
- Java 3D
  - The scene graph
- Modelling 3D objects
  - Polygons
- Interaction and animation
  - Behaviours
  - Collision detection
- Lighting
  - Shading models



# How Do We Create 3D Scenes?



# Project 3D World to 2D Screen



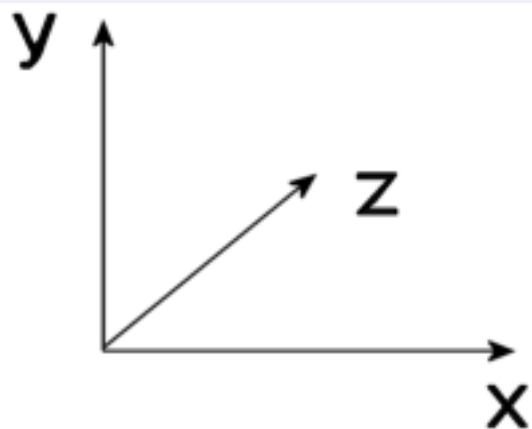
(Java 3D Tutorial Chapt 1 Fig 1-9)

# 3D Co-ordinate Systems - 1

Previous material discussed 2D gaming with 2 co-ordinates denoted by 'x' and 'y' that specify positions along the 'x' and 'y' axis.

- `drawLine(x1,y1,x2,y2);`

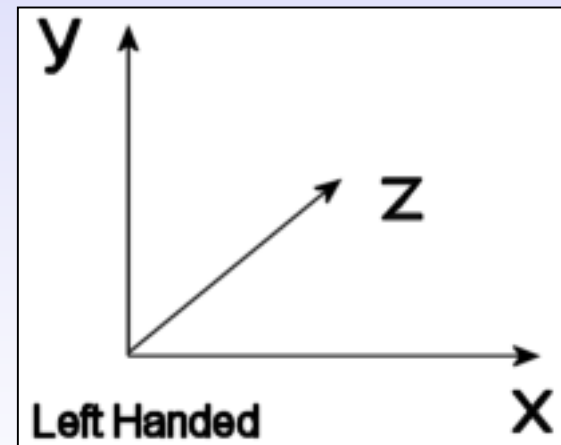
We now wish to add a 3rd dimension on what we shall call the 'z' axis. The 'z' axis effectively defines the depth of a point.



# 3D Co-ordinate Systems - 2

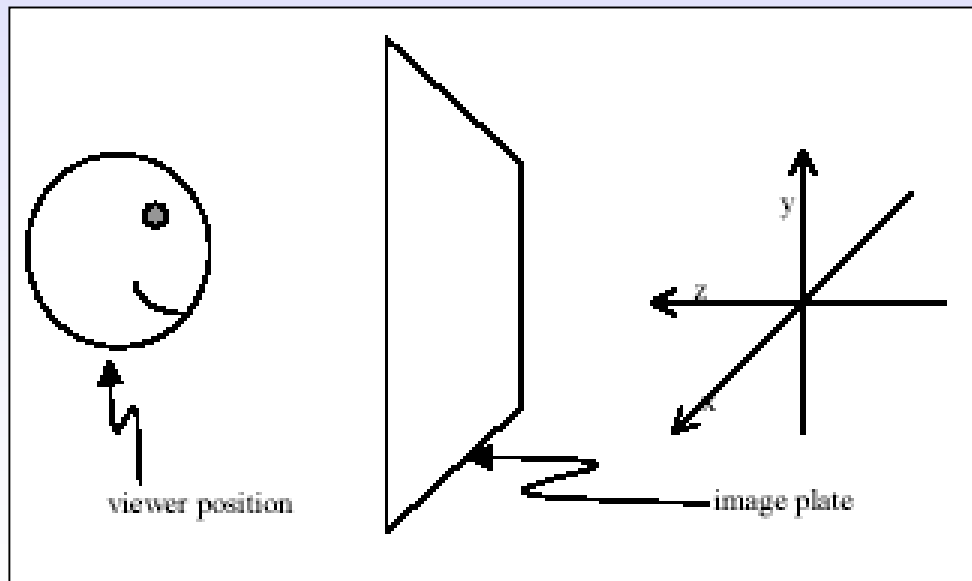
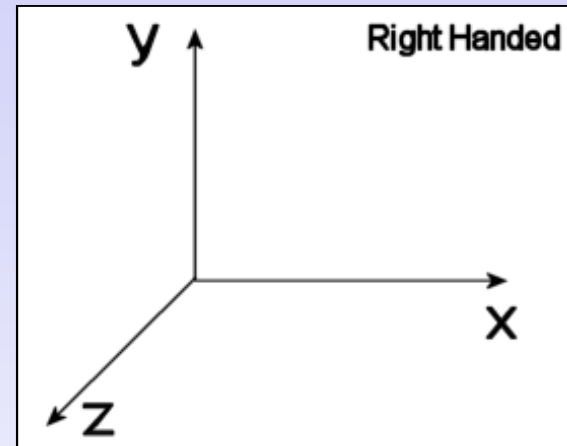
There are 2 main co-ordinate systems in 3D programming which effectively define whether positive values of 'z' move a point towards you or away from you.

- Left handed, z increases as you move away from viewer  
Positive rotation is clockwise about axis of rotation



# 3D Co-ordinate Systems - 3

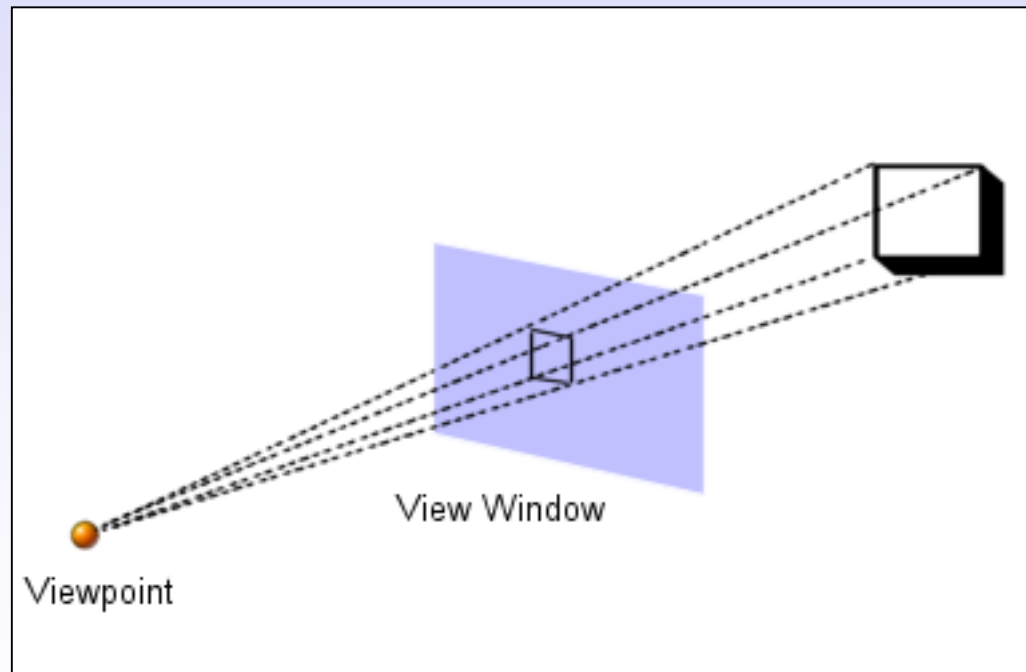
- Right handed, z increases as you move toward viewer  
Positive rotation is anti-clockwise about axis of rotation
  - Java 3D



(Java 3D Tutorial Chapt. 2 Fig. 2-1)

# The View Window

When displaying a 3D scene, an application provides a 2D 'View Window' into the 3D world. As the user moves around, this 2D view changes.

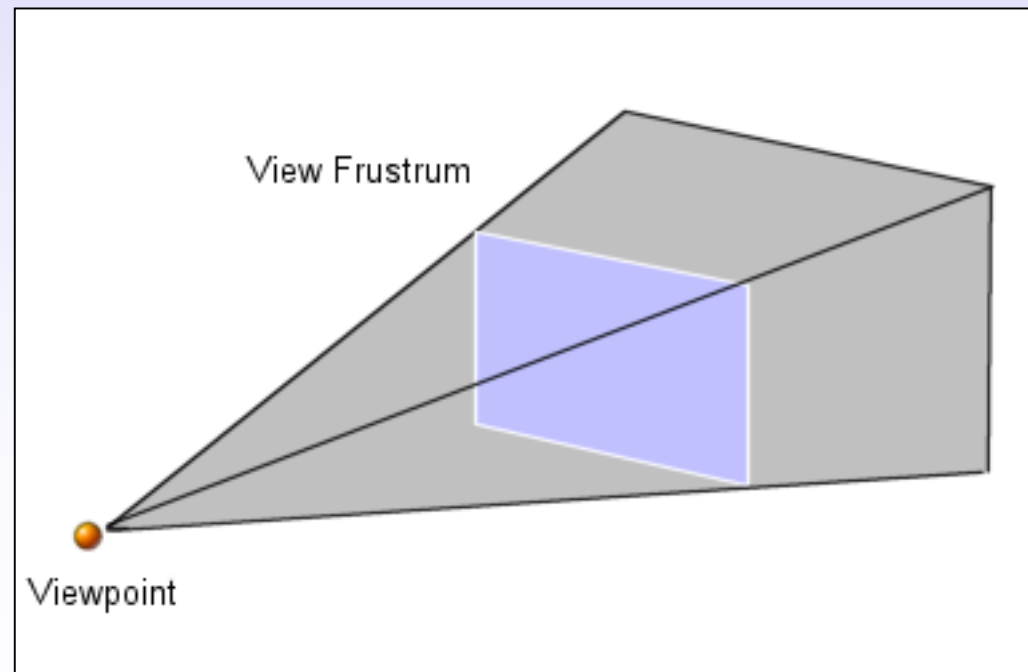


(DGJ Chapt. 7)



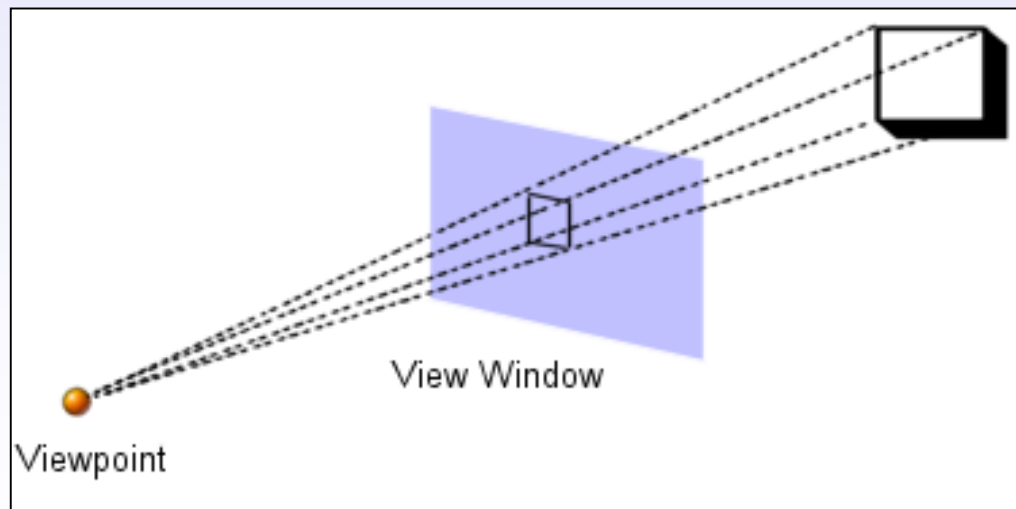
# The View Frustum

Depending on how close to the 'View Window' the observer view point is projected to be, we get a broad or narrow 4 sided pyramid defining the visible area the user can see. This is called the 'View Frustum'.



# Projecting a Point - 1

It can be seen from the diagram of the view window that we can draw a line from a point on our 3D object that passes through the View Window. The View Window is a 2D space which means we are projecting from a 3D co-ordinate into a 2D co-ordinate. How do we do this?



# Projecting a Point - 2

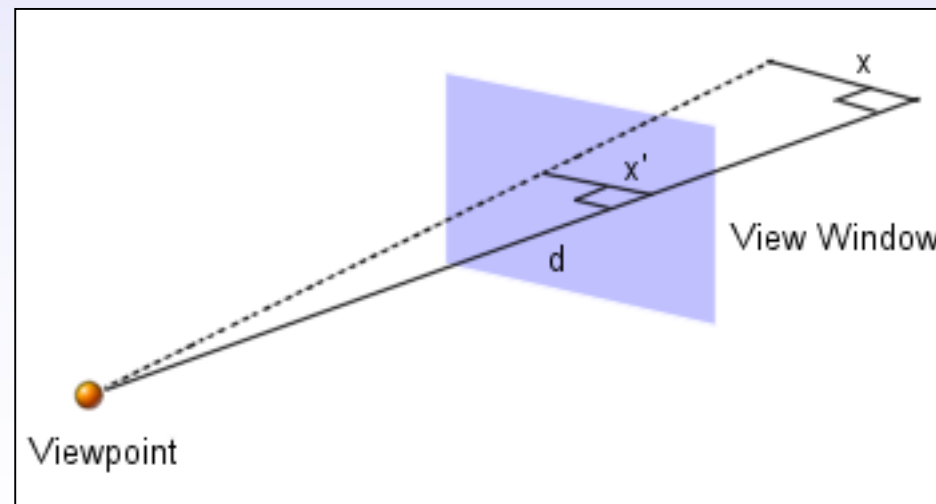
Object is distance  $z$  along the  $z$ -axis from viewer.

And distance  $x$  along the  $x$ -axis.

View window is distance  $d$  from viewer.

What is point  $x'$  along the  $x$ -axis on the window?

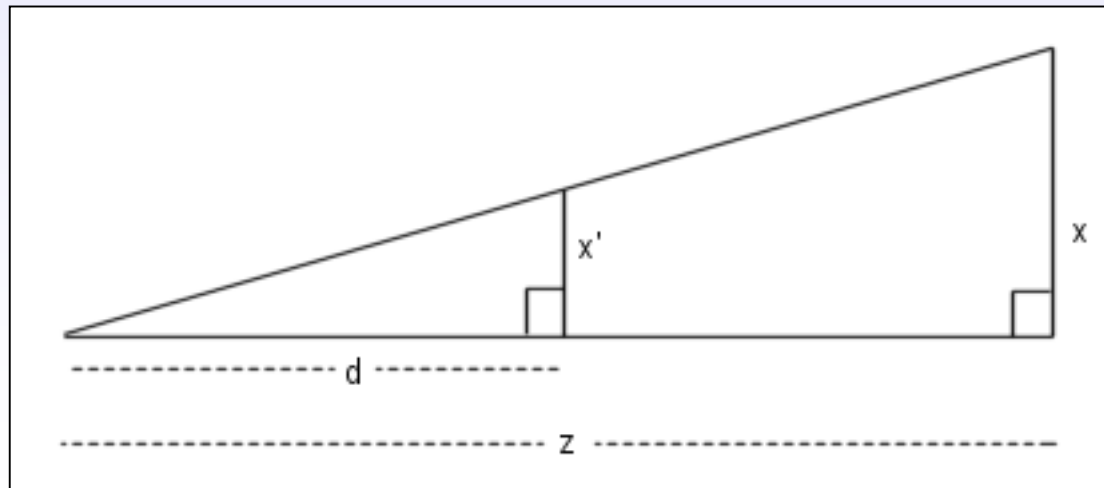
This is a right angle triangle problem...



# Projecting a Point - 3

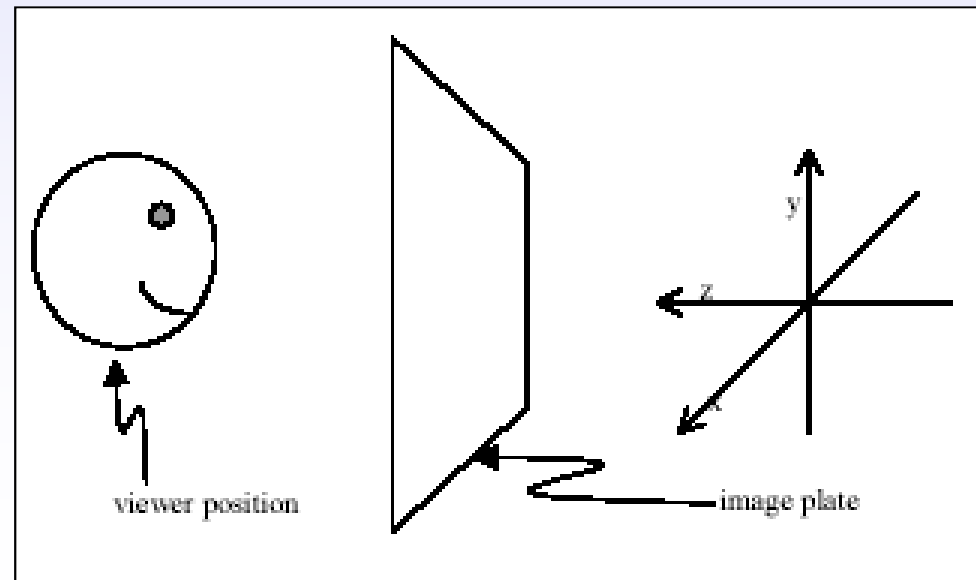
We need to work out what the proportional value of  $x$  on the View Window is (we call this  $x'$ ). It is in the same ratio as the viewing distance ' $d$ ' is to the actual distance ' $z$ '. We know both ' $d$ ' and ' $z$ ' so:

$$x' = (d / z) * x$$



# Formula for 3D Graphics

1. Establish a coordinate system and its origin (0,0,0)
2. Specify size and position of view window and viewpoint (establish view frustum)
3. Draw object in 3D world coordinates e.g.  
`drawCube(x,y,z,height,width,depth);`
4. Project 3D object onto 2D array of screen pixels



The End