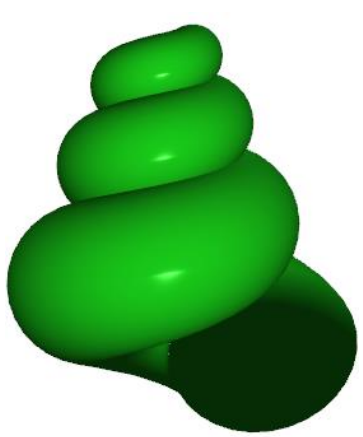
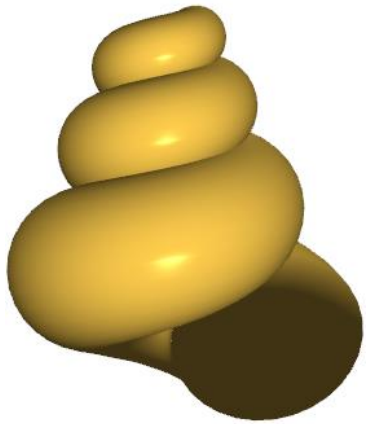
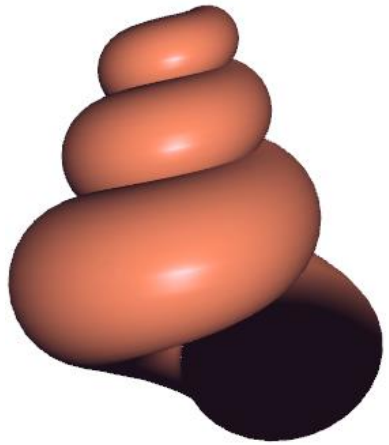
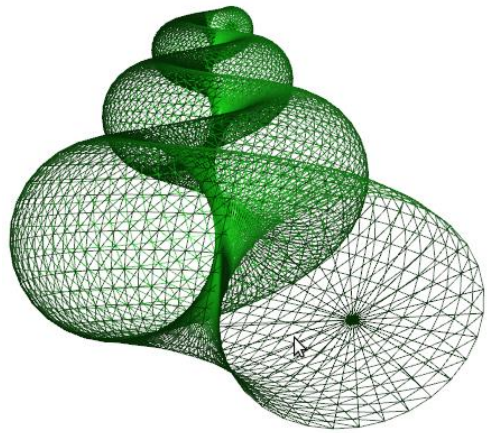
សាកលវិទ្យាល័យភូមិន្ទភ្នំពេញ
អប់រំ ស្រាវជ្រាវ និង សេវាសង្គម
មហាវិទ្យាល័យវិស្វកម្ម

Coordinate Systems

Department of IT Engineering

Administrivia

- Class
  - Theory: T002
    - ✓ Tuesday: 1:00pm – 2:30pm
  - Lab
    - ✓ Monday: 1:00pm – 2:30pm (G1)
    - ✓ Monday: 2:30pm – 4:00pm (G2)
- Exams
  - Final Exams: 60%
  - Assignment: 20%
  - Homework: 10%
  - Attendance: 10%

Contents

- Coordinate System
  - Cartesian Coordinates
  - Polar Coordinates
  - Cylindrical Coordinates
  - Spherical Coordinates
  - World-Window-Image-Screen

Introduction

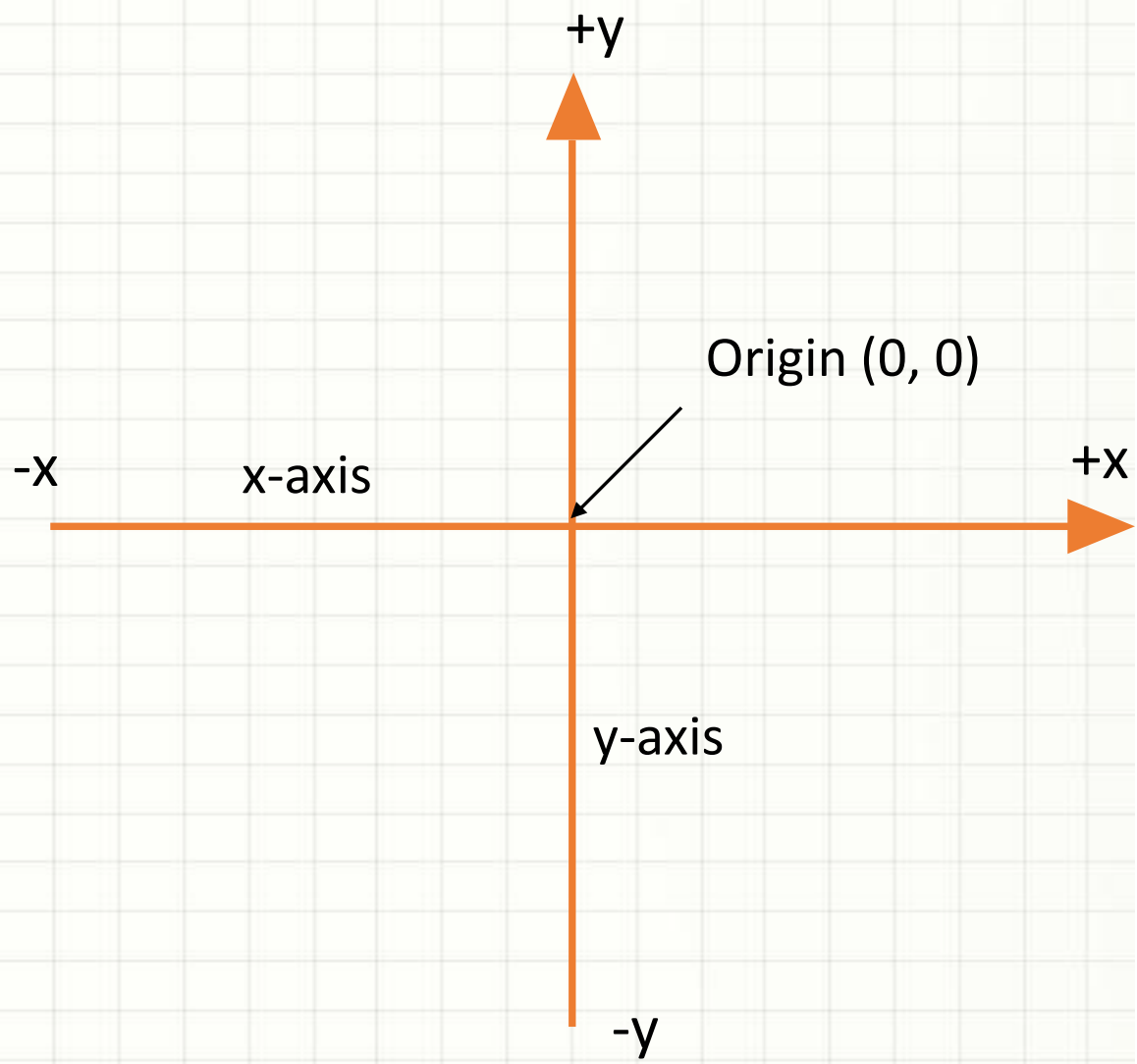
- What is Coordinate System?

In geometry, a coordinate system is a system using one or more numbers, or coordinates, to uniquely determine the position of a point or other geometric element on a manifold such as **Euclidean Space**

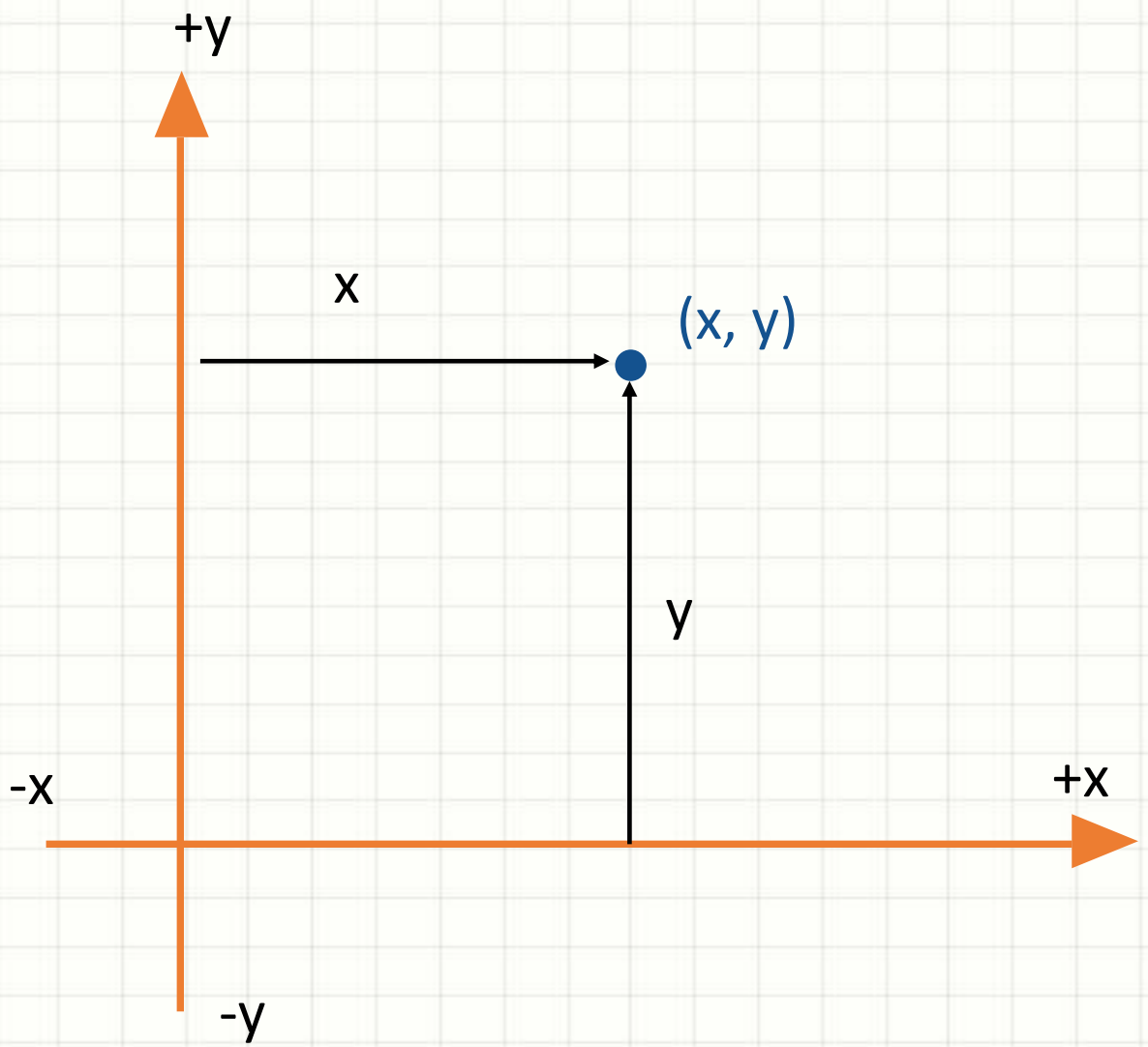
- Common Coordinate Systems
  - Cartesian Coordinate System
  - Polar Coordinate System
  - Cylindrical Coordinate System
  - Spherical Coordinate System

2D Coordinate Spaces

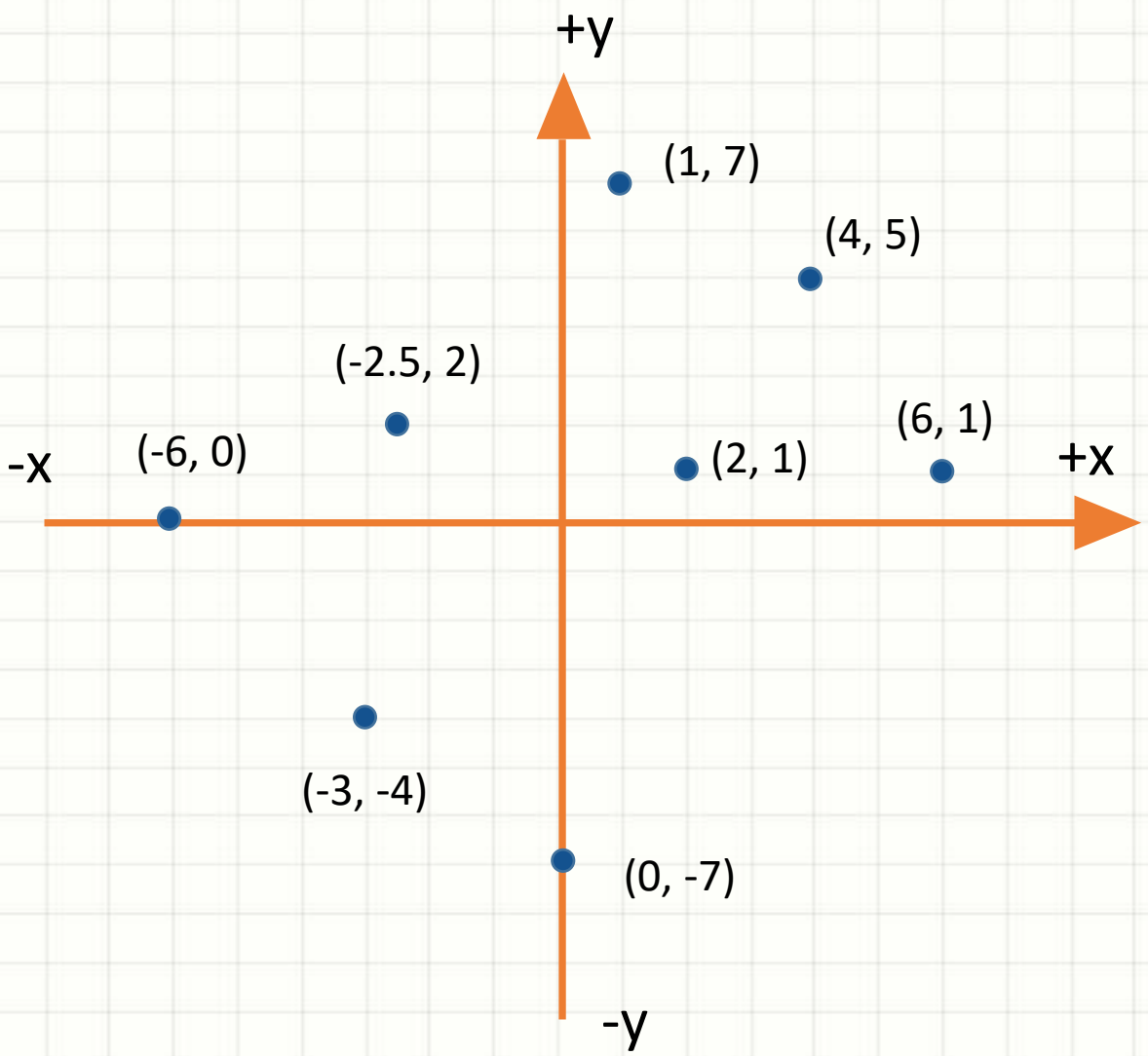
- Consist of a special location called the origin
- Exist two straight lines that pass through the origin
- Two axes are perpendicular to each other



Specify a point in 2D Cartesian

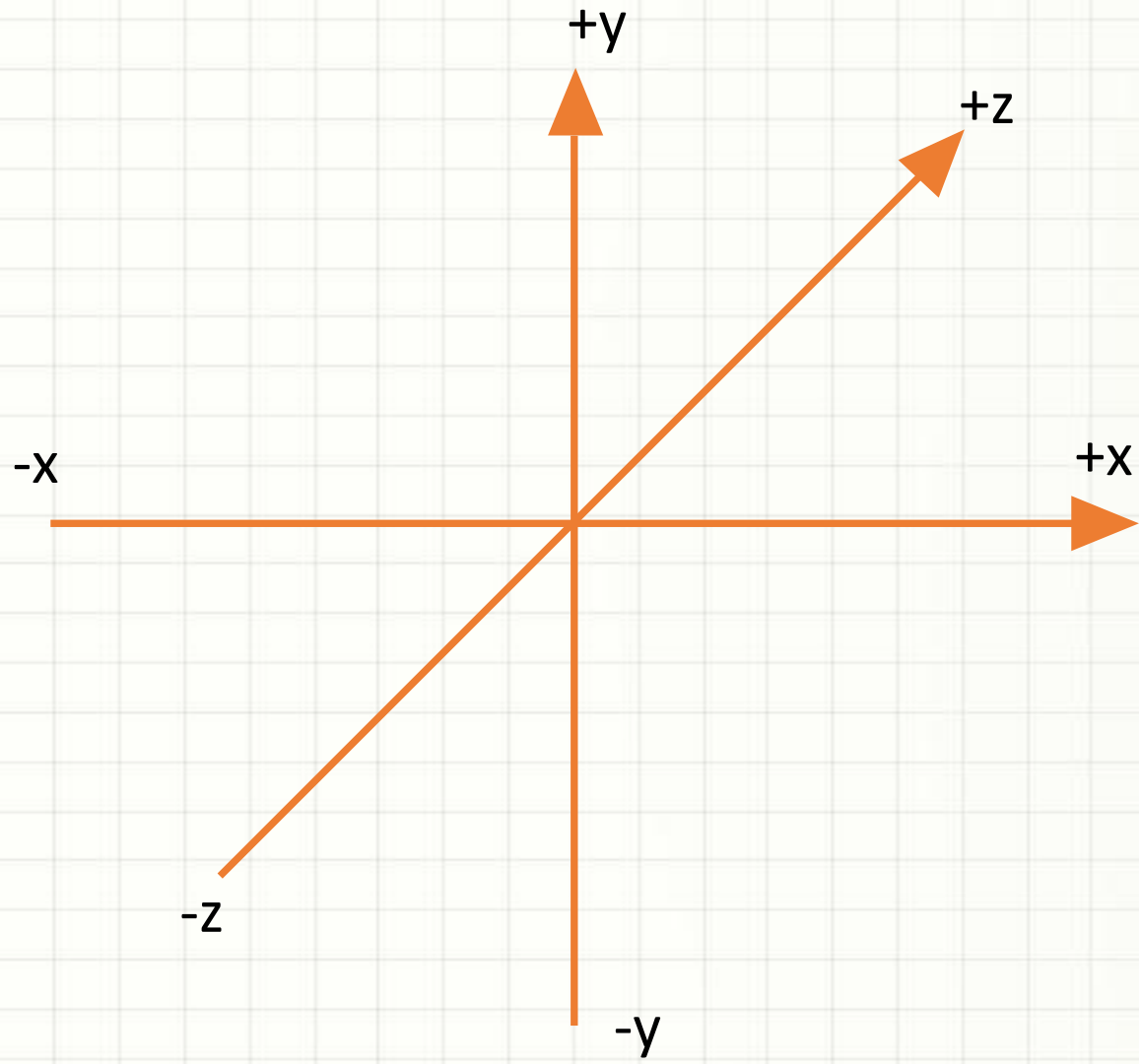


Example points labeled with 2D Cartesian Coordinates

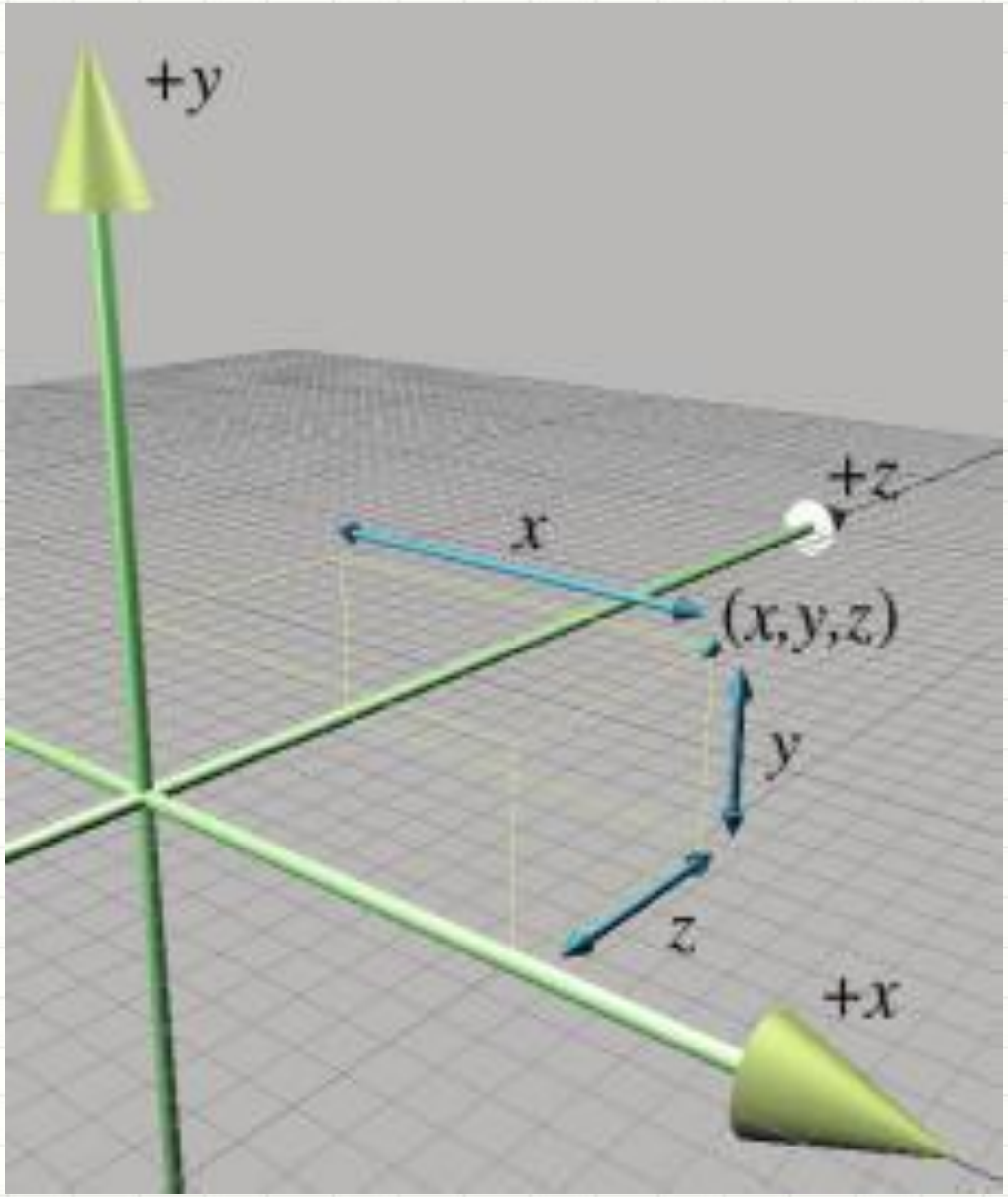


3D Cartesian Space

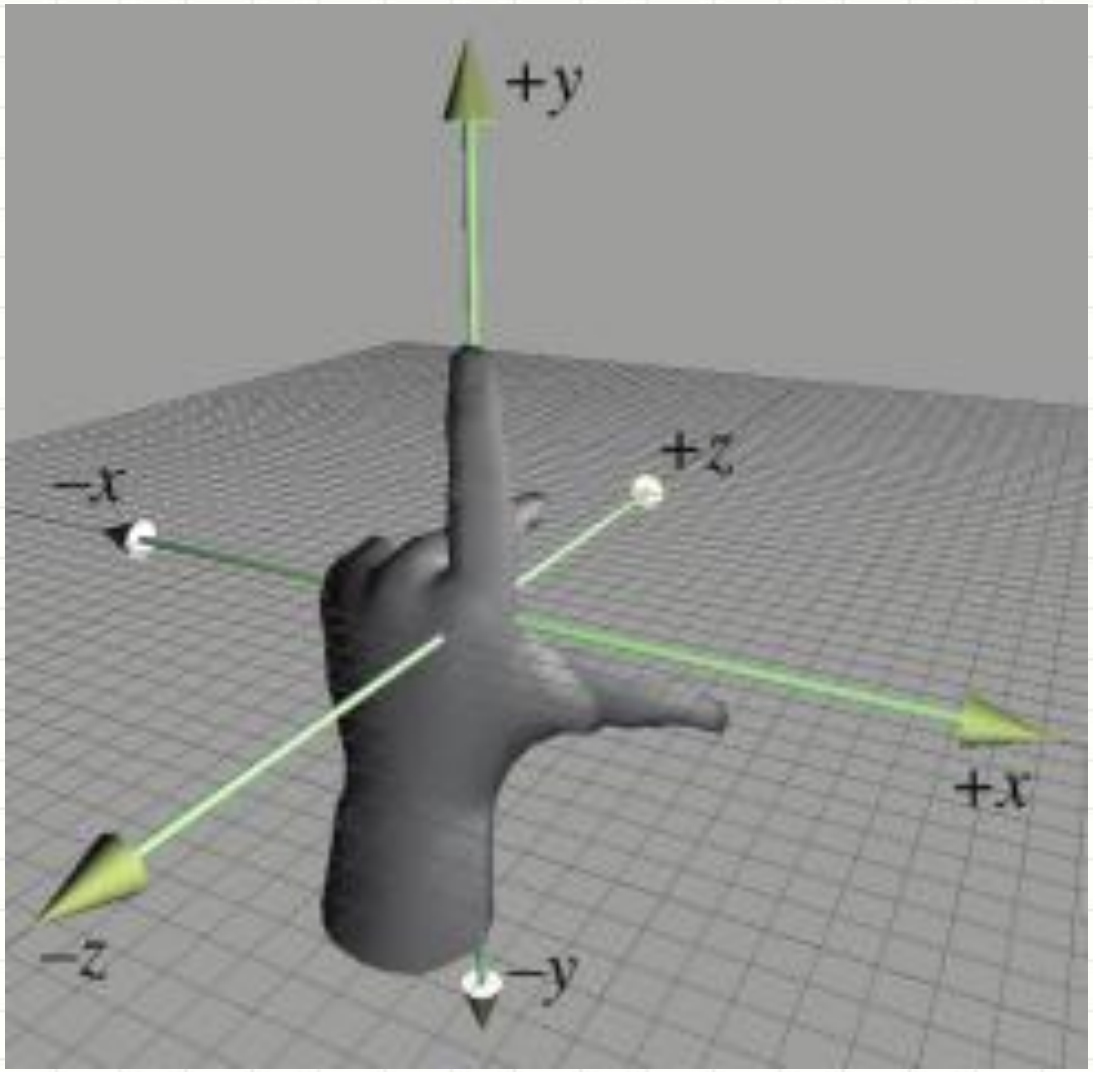
- Extend 2D into 3D by adding a 3rd axis
- **xy**, **xz**, and **yz** planes

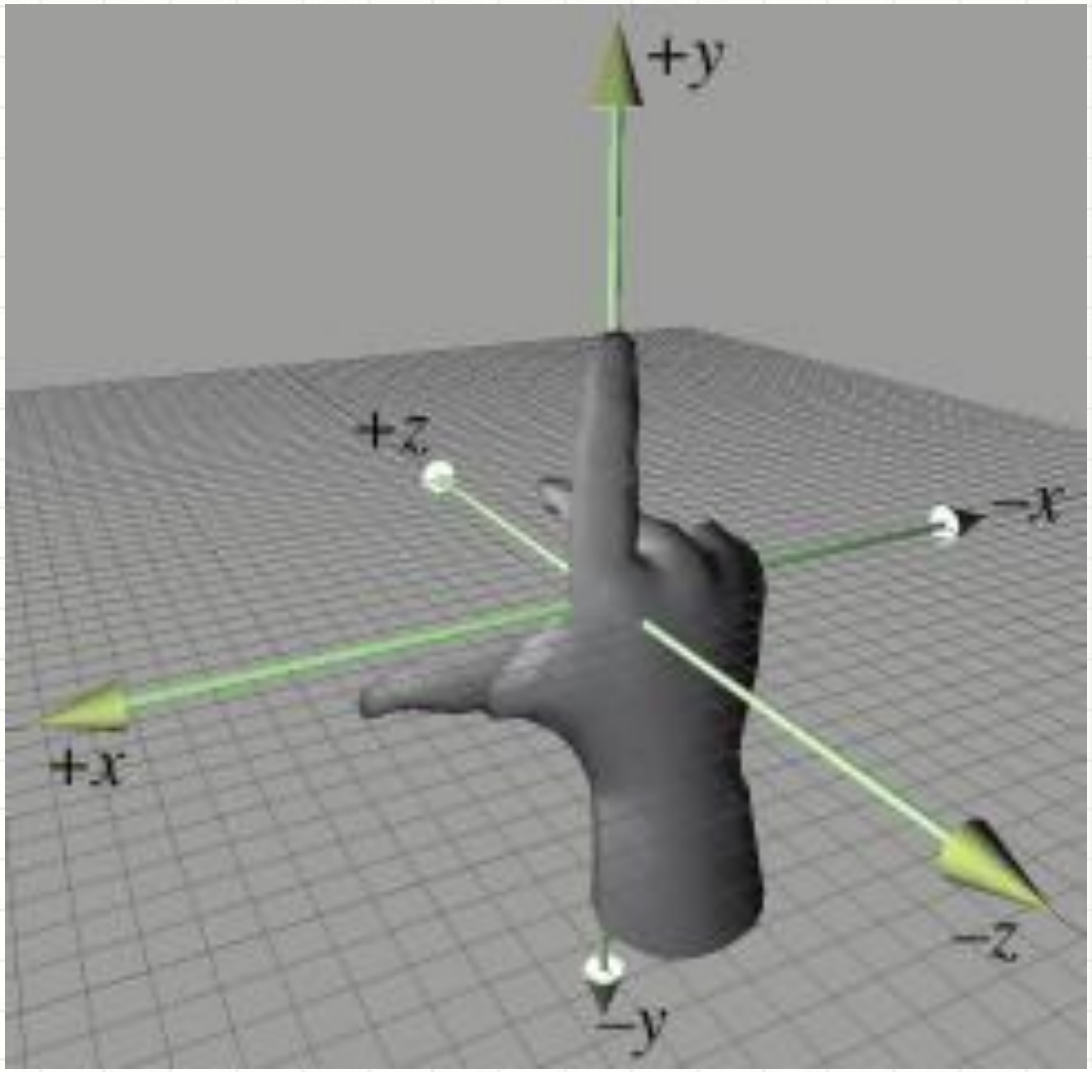


Specify a point in 3D Cartesian

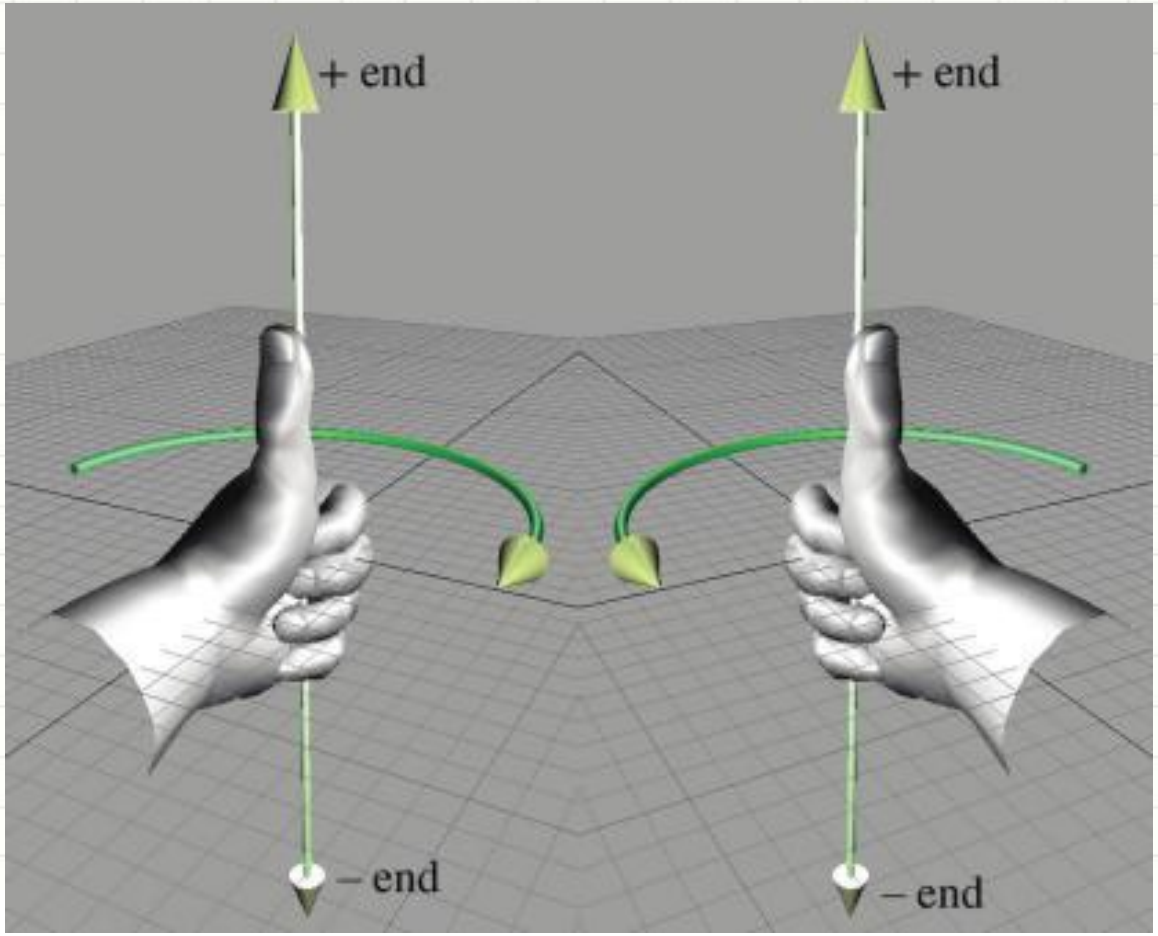


Left-handed & Right-handed Rule



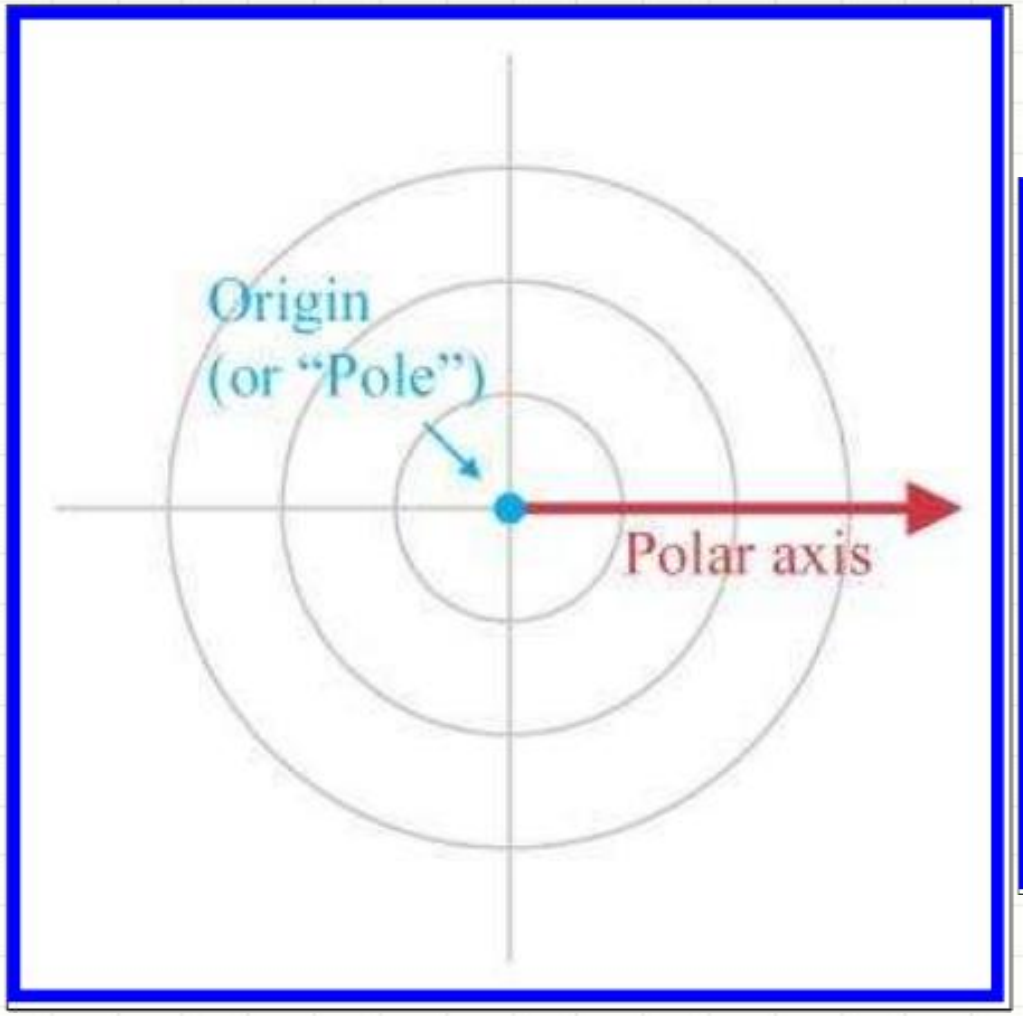


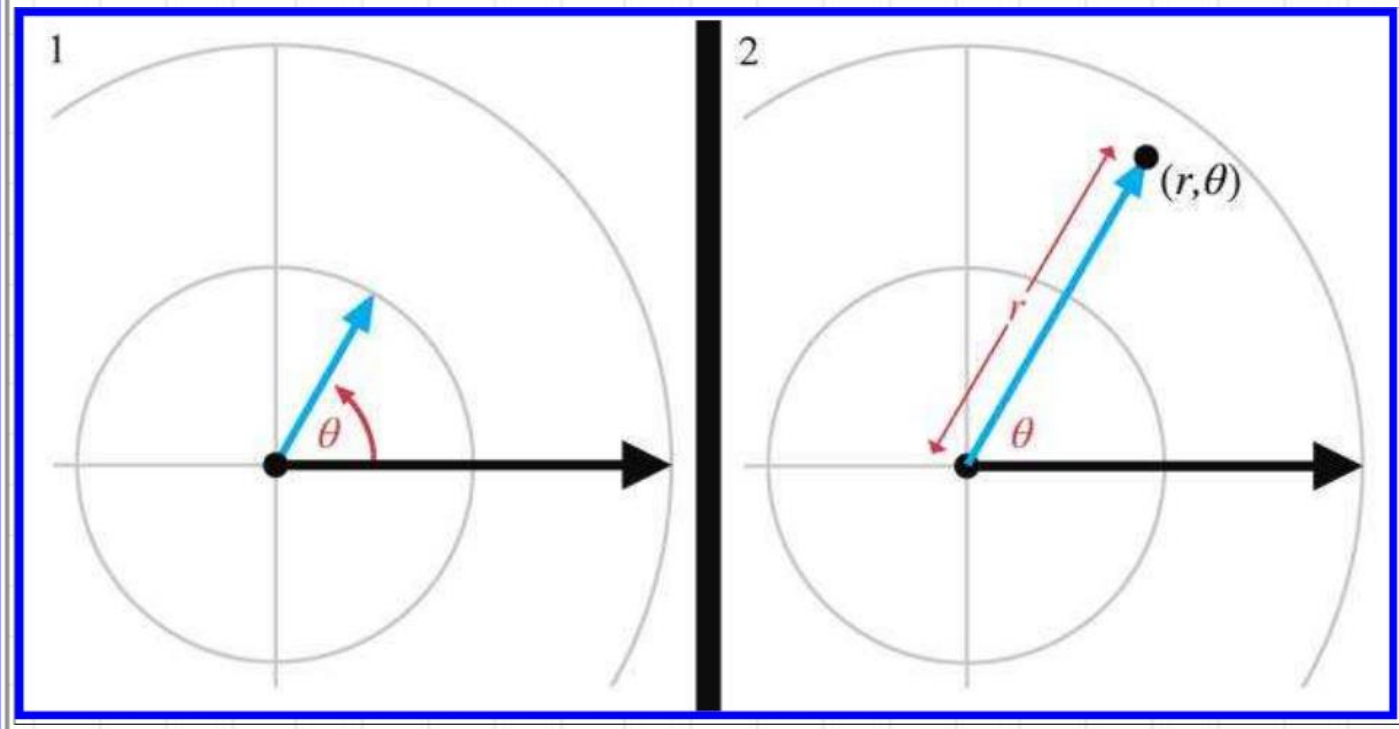
Positive Rotation for Left-Right Hand



When looking towards the origin from…	**Positive rotation** Left-handed: Clockwise Right-handed: Counterclockwise	**Negative rotation** Left-handed: Counterclockwise Right-handed: Clockwise
$+x$	$+y \to +z \to -y \to -z \to +y$	$+y \to -z \to -y \to +z \to +y$
$+y$	$+z \to +x \to -z \to -x \to +z$	$+z \to -x \to -z \to +x \to +z$
$+z$	$+x \to +y \to -x \to -y \to +x$	$+x \to -y \to -x \to +y \to +x$

Polar Coordinates





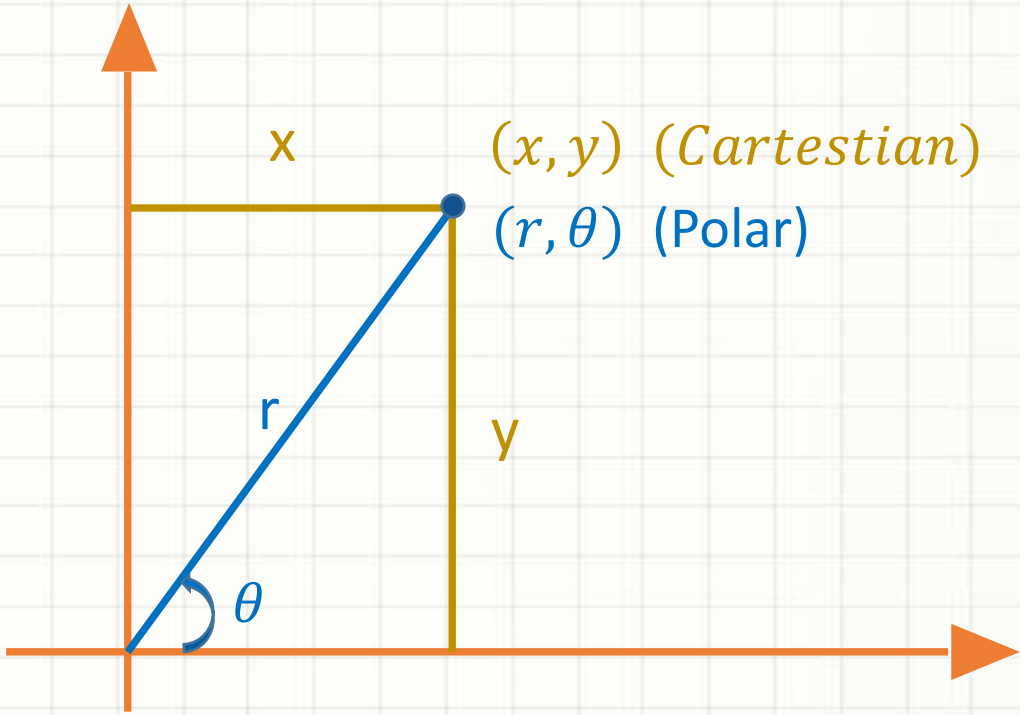
Convert between Cartesian and Polar

- Convert from Polar to Cartesian

$$x = r\cos\theta$$
$$y = r\sin\theta$$

- Convert from Cartesian to Polar

$$r = \sqrt{x^2 + y^2}$$
$$\theta = atan2(y, x)$$



$$atan2(y,x) = \begin{cases} 0, & x = 0, y = 0 \\ +90^o, & x = 0, y > 0 \\ -90^o, & x = 0, y < 0 \\ arctan(y/x), & x > 0, \\ arctan(y/x) + 180^o, & x < 0, y \geq 0 \\ arctan(y/x) - 180^o & x < 0, y < 0 \end{cases}$$

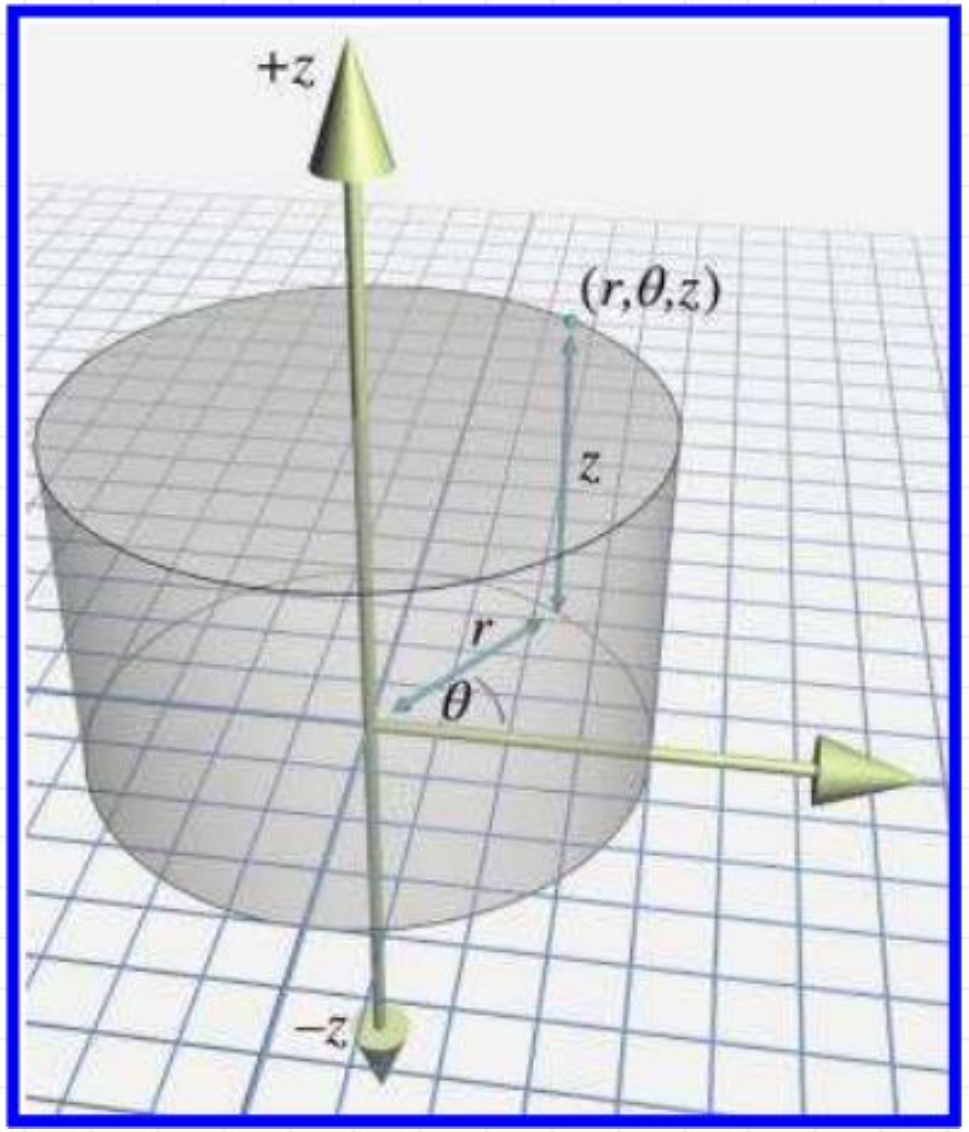
Cylindrical Coordinates

- Convert from Cylindrical to Cartesian Coordinate

$$x = r\cos\theta$$

$$y = r\sin\theta$$

$$z = z$$



Spherical Coordinates

- Convert from Spherical to Cartesian

$$x = r\sin\Phi\cos\theta$$

$$y = r\sin\Phi\sin\theta$$

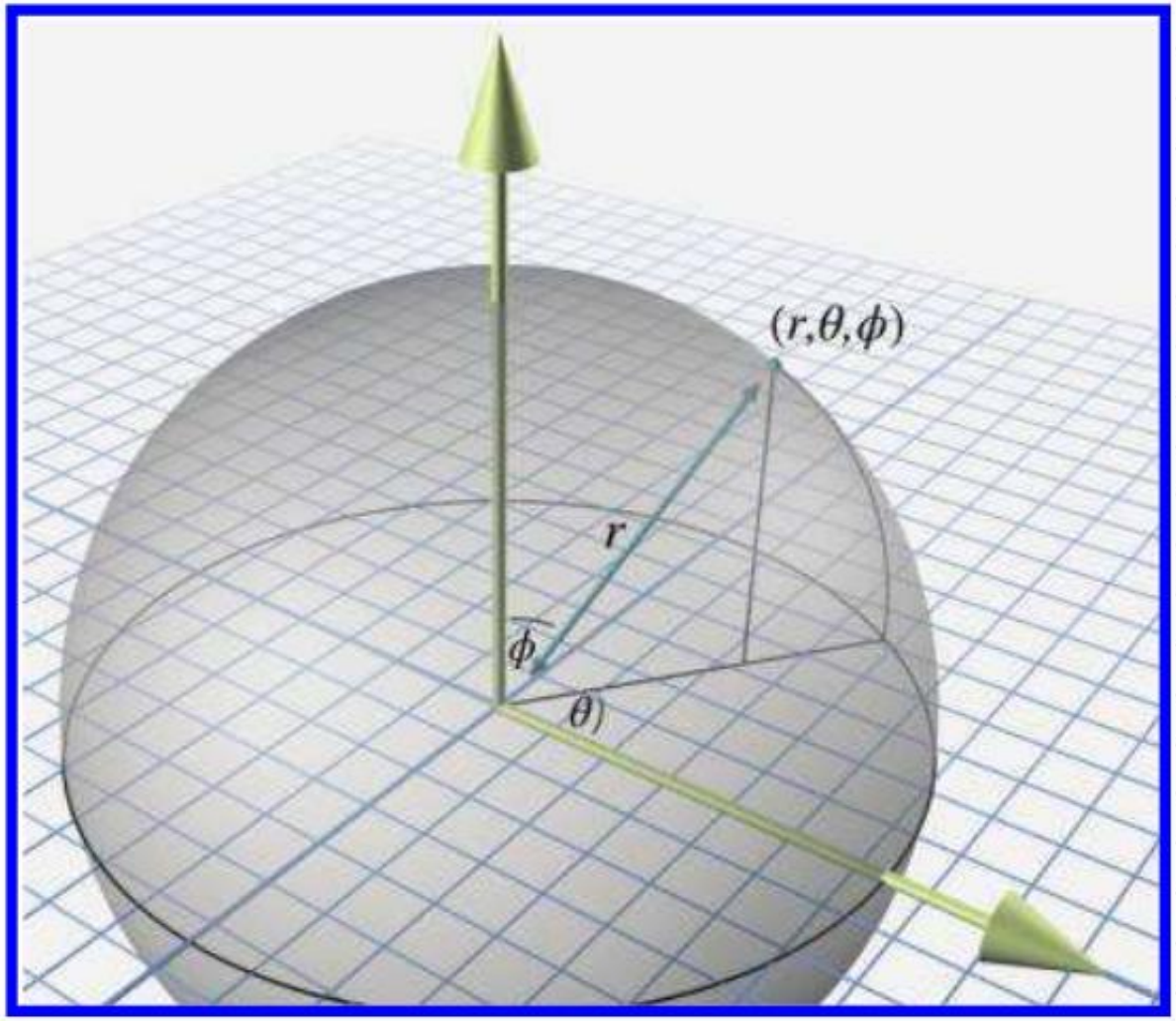
$$z = r\cos\Phi$$

- Convert from Cartesian to Spherical

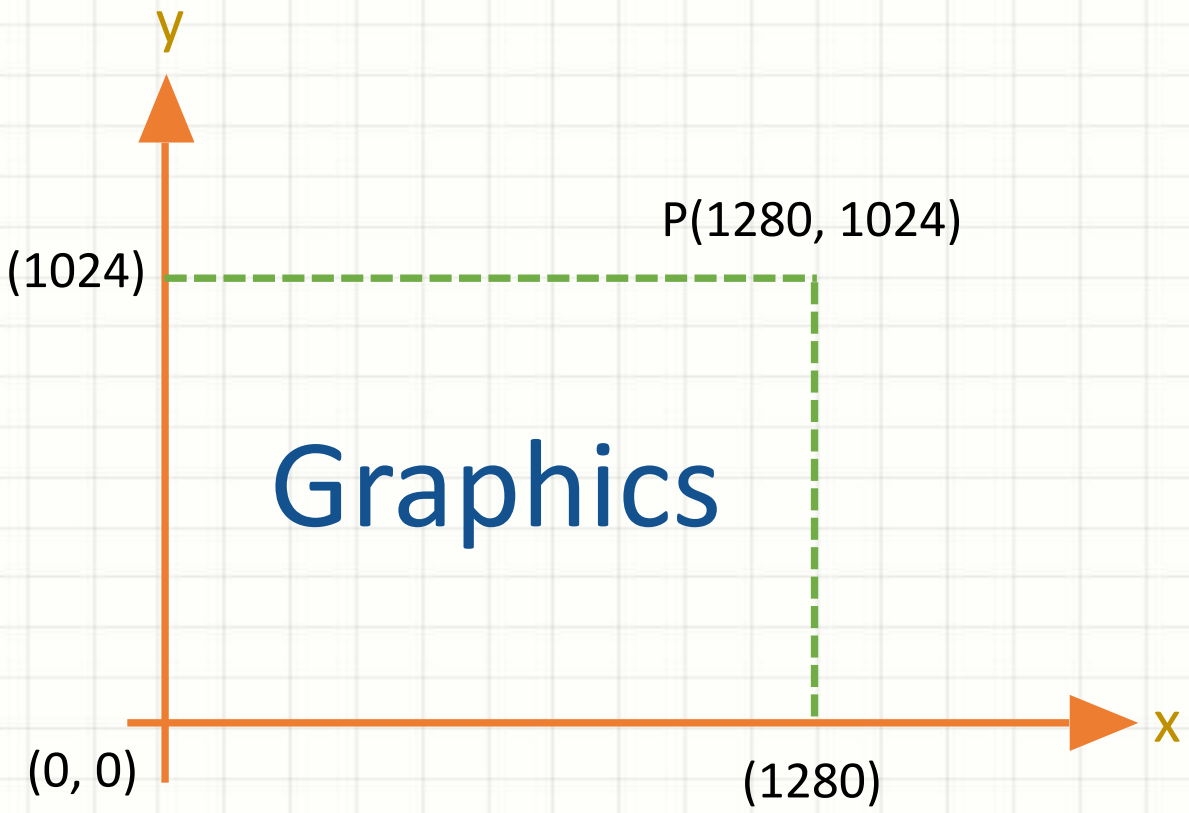
$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\Phi = \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right)$$

$$\theta = \arctan\left(\frac{y}{x}\right)$$



Coordinate Systems



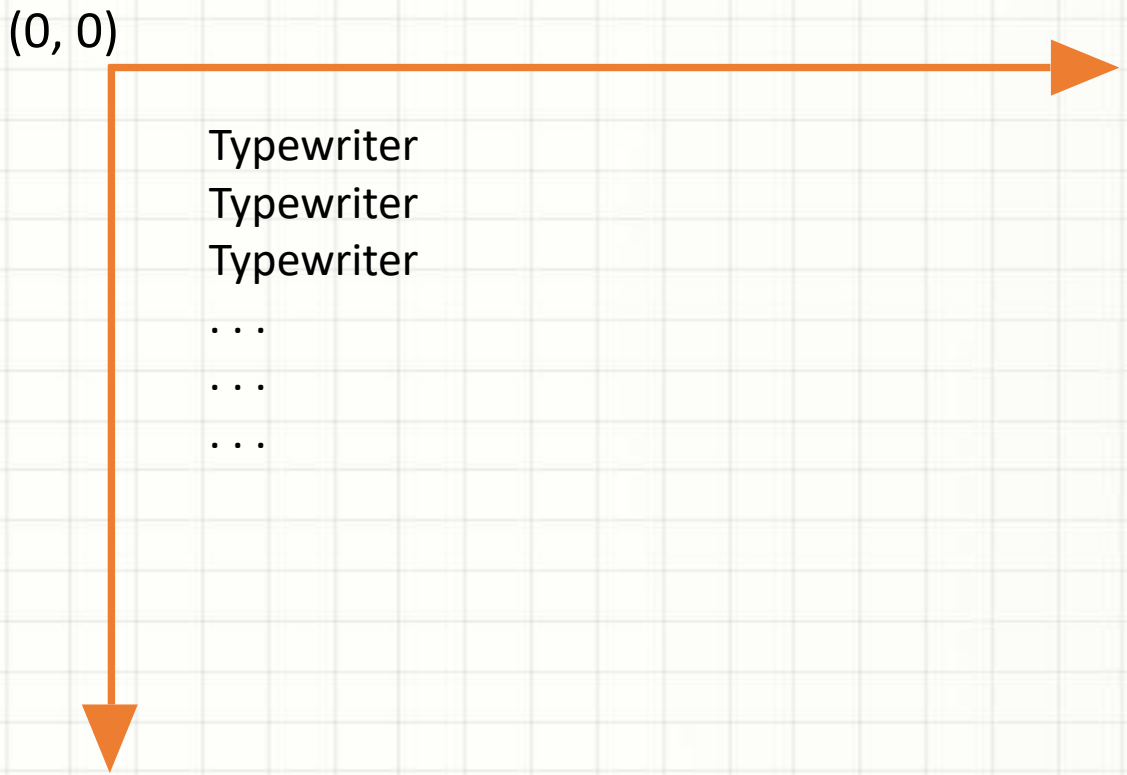
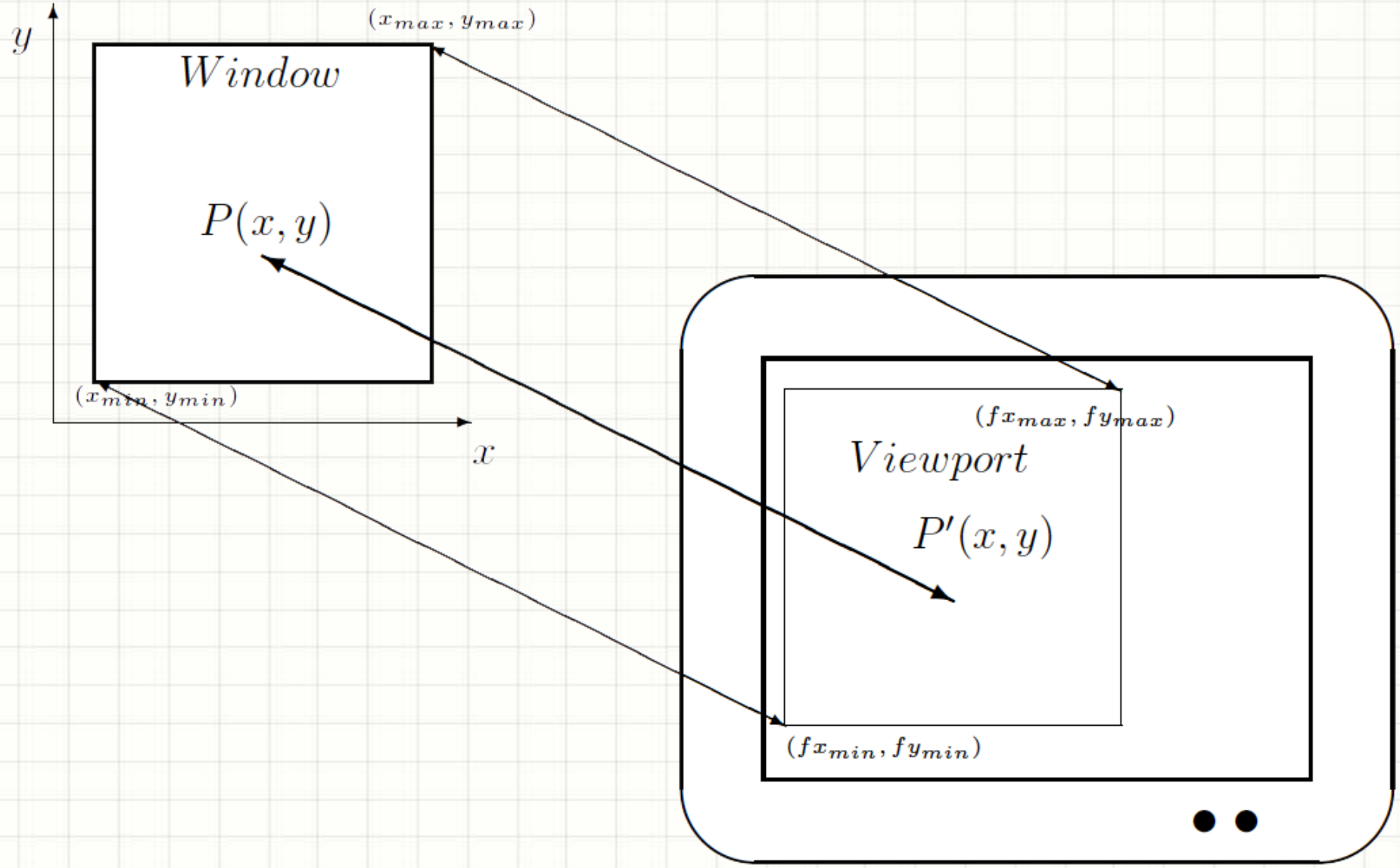


Image Processing

Word Processing

World – Window – Image - Screen



GLUT is able to handle windows

```
glutInitWindowSize(GLInt width, GLint height);
```

```
glutInitWindowPosition(GLint x, GLint y);
```

OpenGL acts in windows

```
glViewport(GLInt x, GLInt y, GLInt width, GLint height);
```

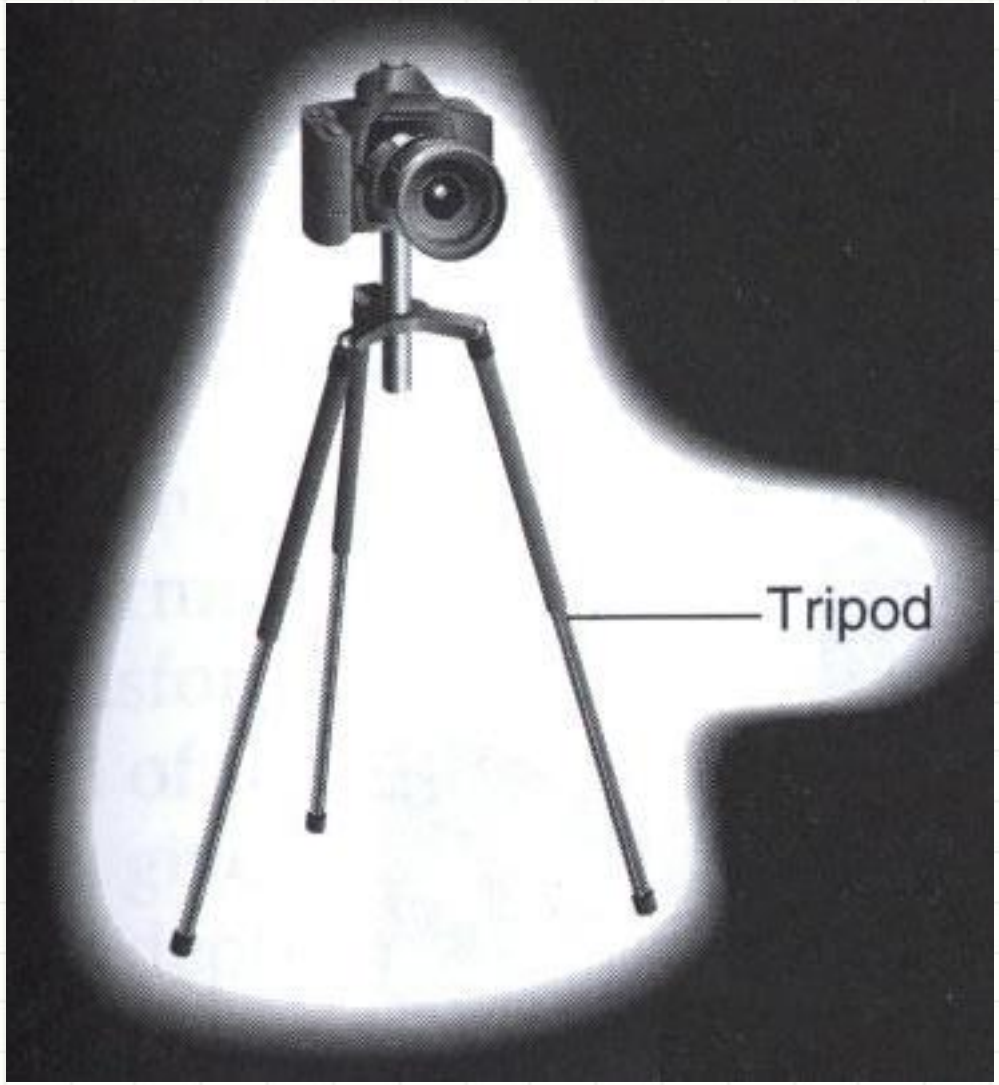
```
glViewport(0, 0, win_x, win_y);
```

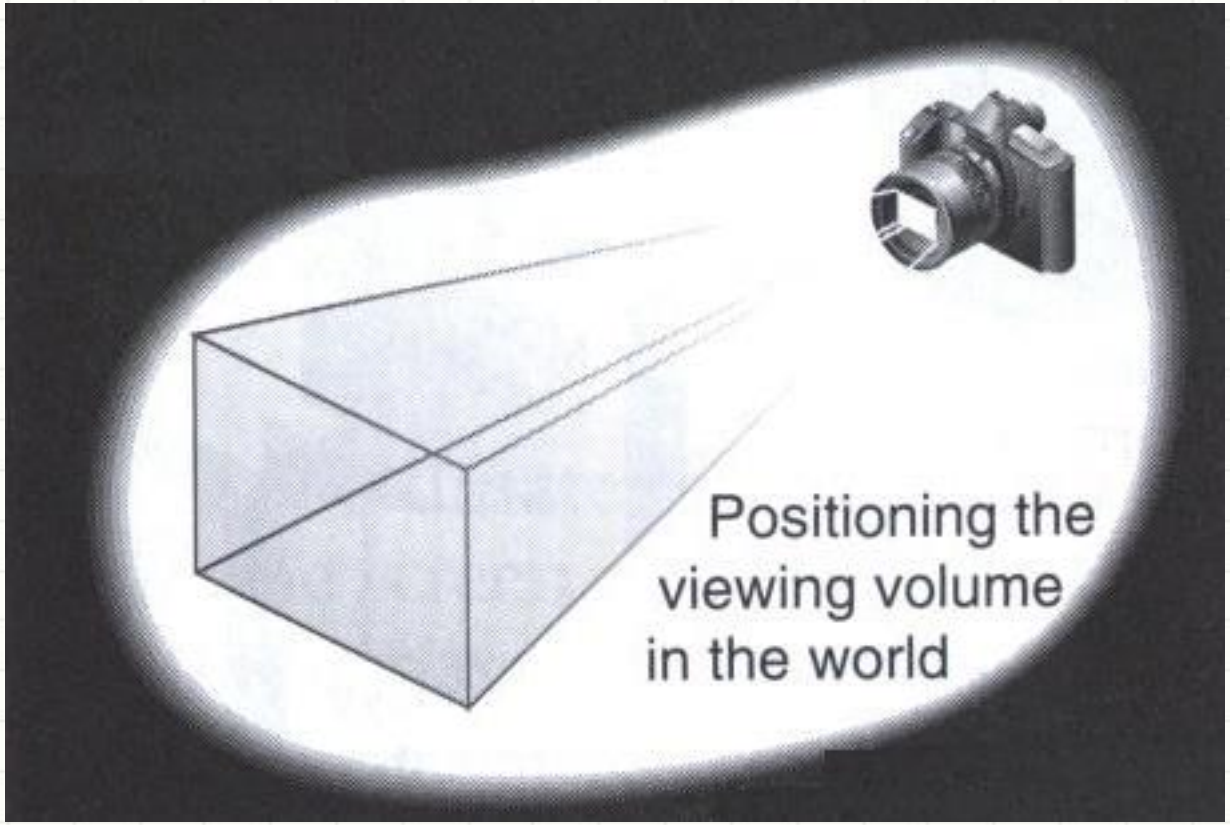
(x, y) specify the lower left corner of the viewport rectangle default **(0, 0)**

(width, height) specify the upper right corner of the viewport rectangle, default **(win_x, win_y)**

The Camera Analogy

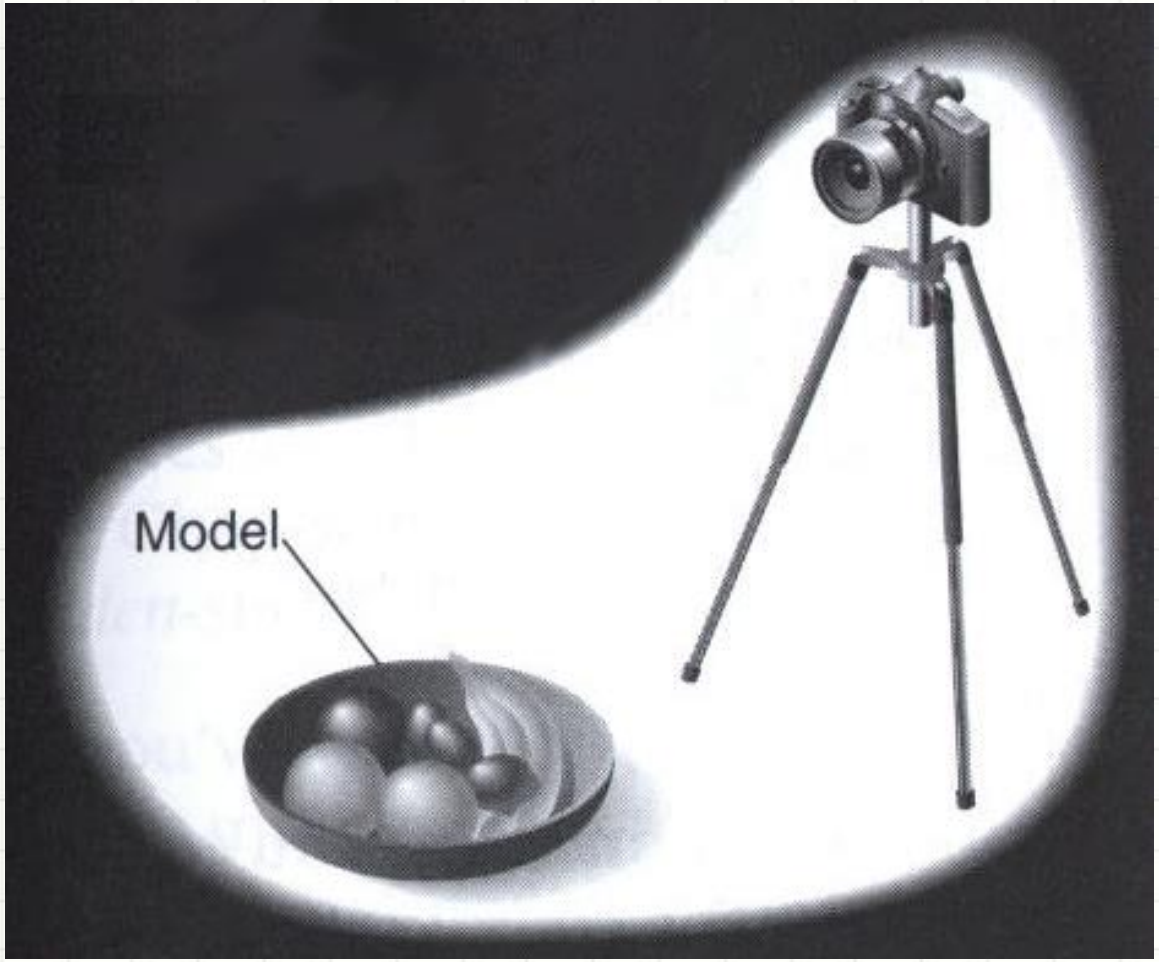
Placing the camera and pointing the viewing direction

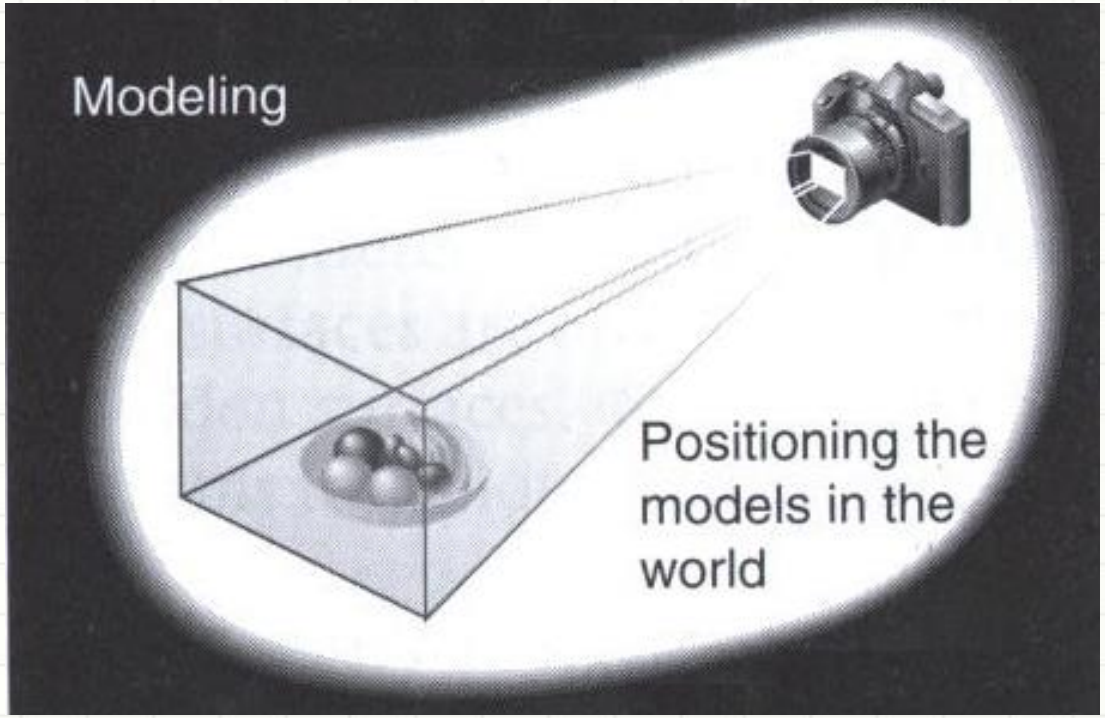




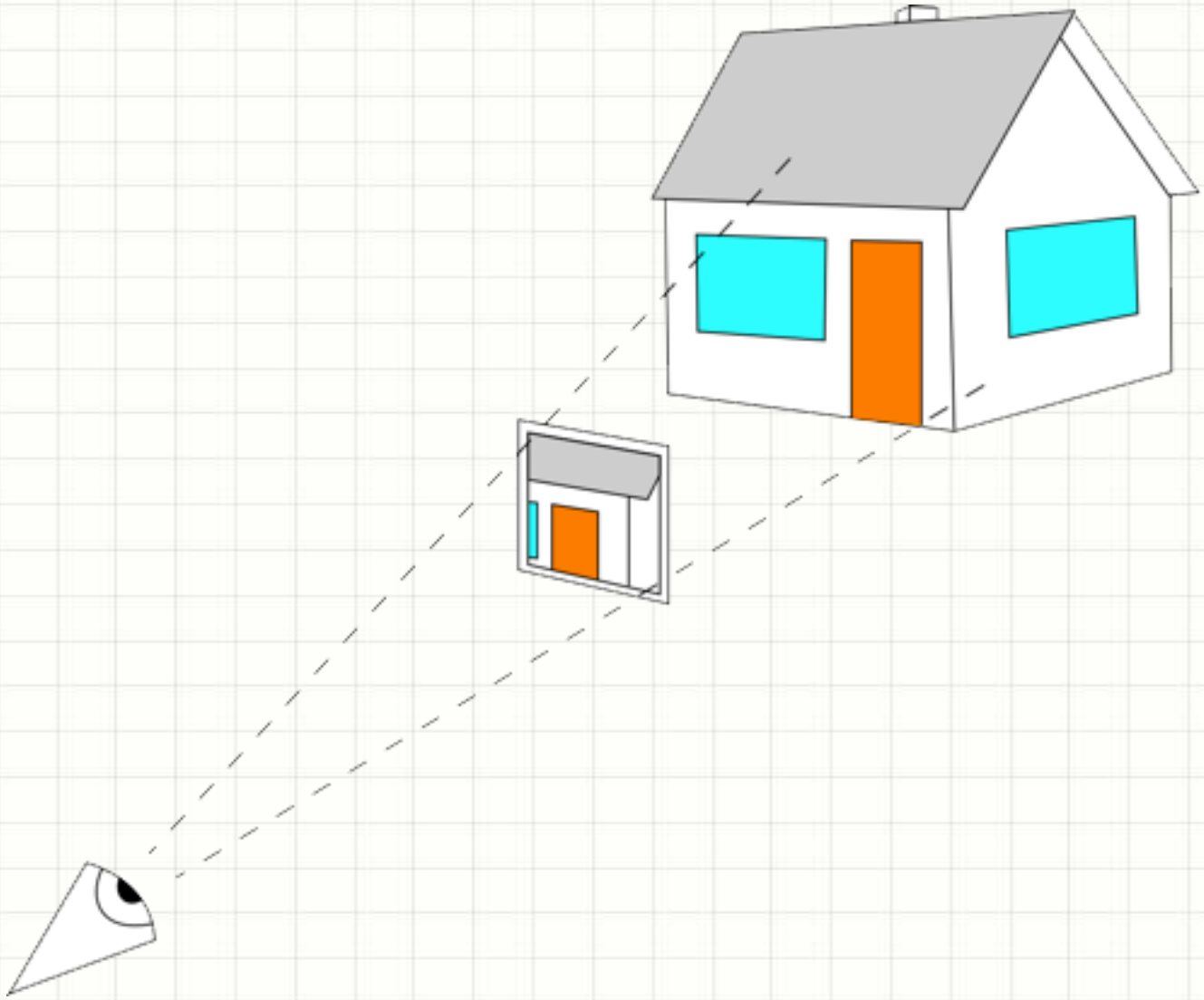
Modeling

Arranging the scene





2D representation of your world/scene



Literature

- Foley, J. D., Van Dam, A., Feiner, S.K., Hughes, J. F., & Phillips R. L. (1996). *Introduction to Computer Graphics*.
- Watt A. H. (1990). *Fundamentals of three-dimensional computer graphics*. Addison-Wesley.
- D.H. Eberly, *3D game engine design, a practical approach to real-time computer graphics, Academic Press, Morgan Kaufmann, 2001*
- Hughes, J. F., Van Dam, A., Foley, J. D., & Feiner, S. K. (2013). *Computer graphics: principles and practice.* Pearson Education.
- *Dunn, F., & Parberry, I. (2011). 3D math primer for graphics and game development. CRC Press.*
- *ARB, Dave Shreiner, editor,*
  - *OpenGL programming guide (RED)*