

# 2D Coordinate Systems

- ## 2D Geometric Objects
- A 2D geometric object is a set of points in the plane
  - Almost always an infinite set of points
    - Therefore, can't simply list all of those points
  - Usually, equation used to represent object, defining relation between x and y coordinates

### Example: Line

$$Ax + By + C = 0$$

### Example: Ellipse

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

- ## Parametric Equations
- For some geometric objects, it's more convenient to use parametric representation
  - For lines, easy enough to find coordinates of points on line:  $y = m x + b$
  - For more intricate curves, not so easy
  - Need to solve equation for y in terms of x
  - More complicated for 3D and higher dimensional spaces
  - Use parametric equations instead

## Parametric Equation

Introduce an independent variable t

$$x = f(t)$$

$$y = g(t)$$

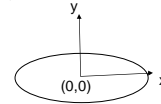
Parametric equation of a line	Parametric equation of an ellipse
$x = x_0 + at$	$x = x_0 + a \cos t$
$y = y_0 + bt$	$y = y_0 + b \sin t$

## Space

- A space is a collection of all points or coordinates
- Types of space in a graphics system:
  - Object Space
  - World Space
  - Device Space
- Each space has own coordinate system
- Objects can be mapped between coordinate systems through transformations

## Types of Coordinate Space

- Object Coordinate System
  - Local or modeling coordinate system
  - Typically choose a convenient coordinate system for the individual object
    - E.g., Center an ellipse at (0,0)
  - Object later placed in world space through transformations (e.g., translations, scaling, etc)

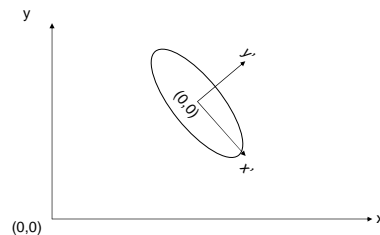


## Types of Coordinate Space

- World Coordinate System
  - Common reference space for all objects in model
  - Shared virtual world for modeling and rendering subsystems
  - Object transformations used to place objects in world space

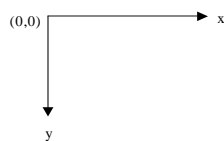


## An object in world space

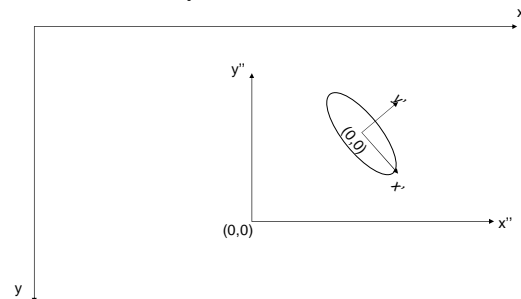


## Types of Coordinate Space

- Device Coordinate System
  - Display of an output device (e.g., screen or printer)



## An object in world space, rendered in device space



## Rendering Pipeline

1. Construct the 2D objects.
2. Apply transformations to the objects.
3. Apply color and other rendering properties.
4. Render the scene on a graphics device.

