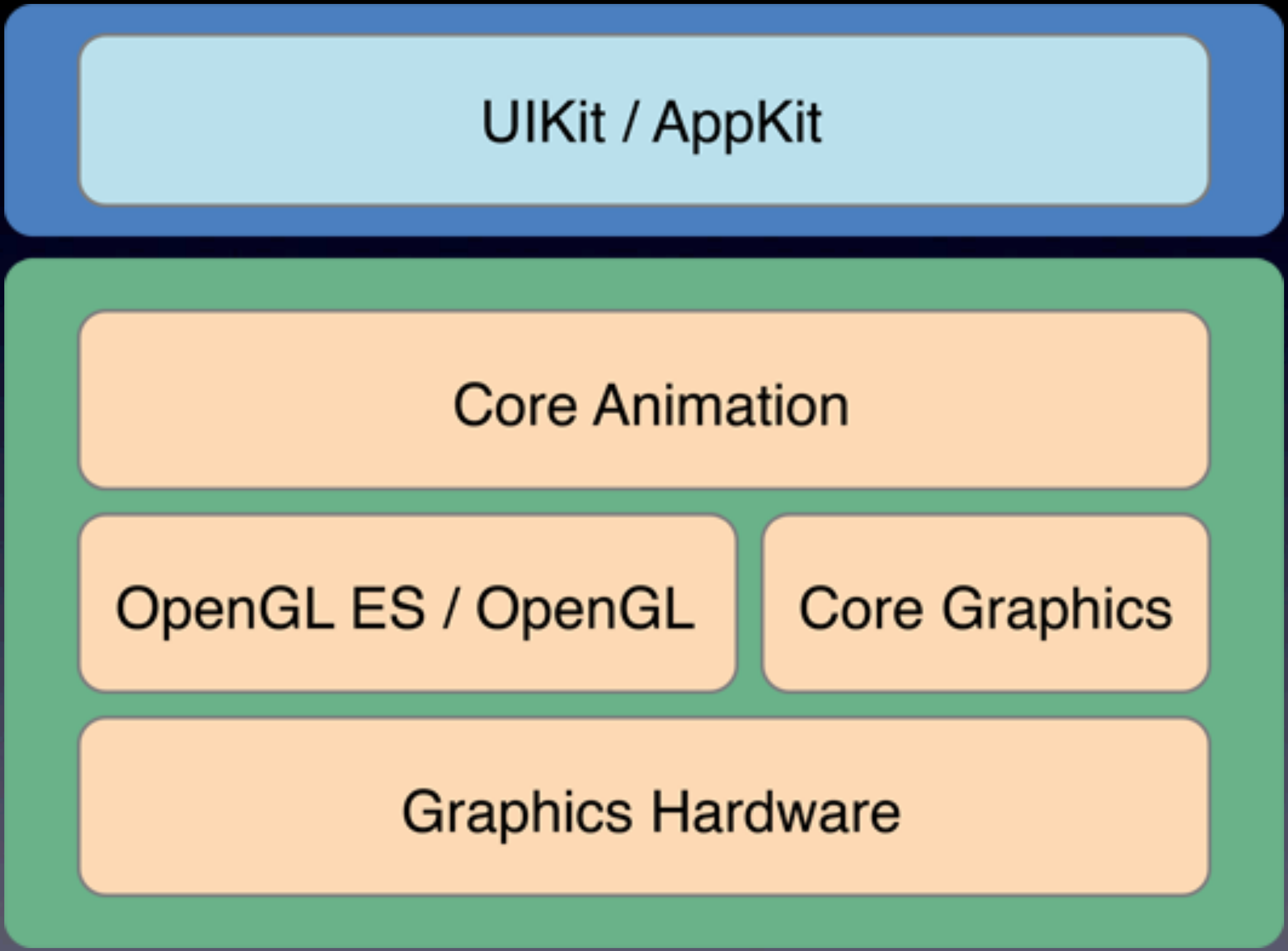
Views and Stuff

The iOS/OSX graphics stack



UIKit / AppKit

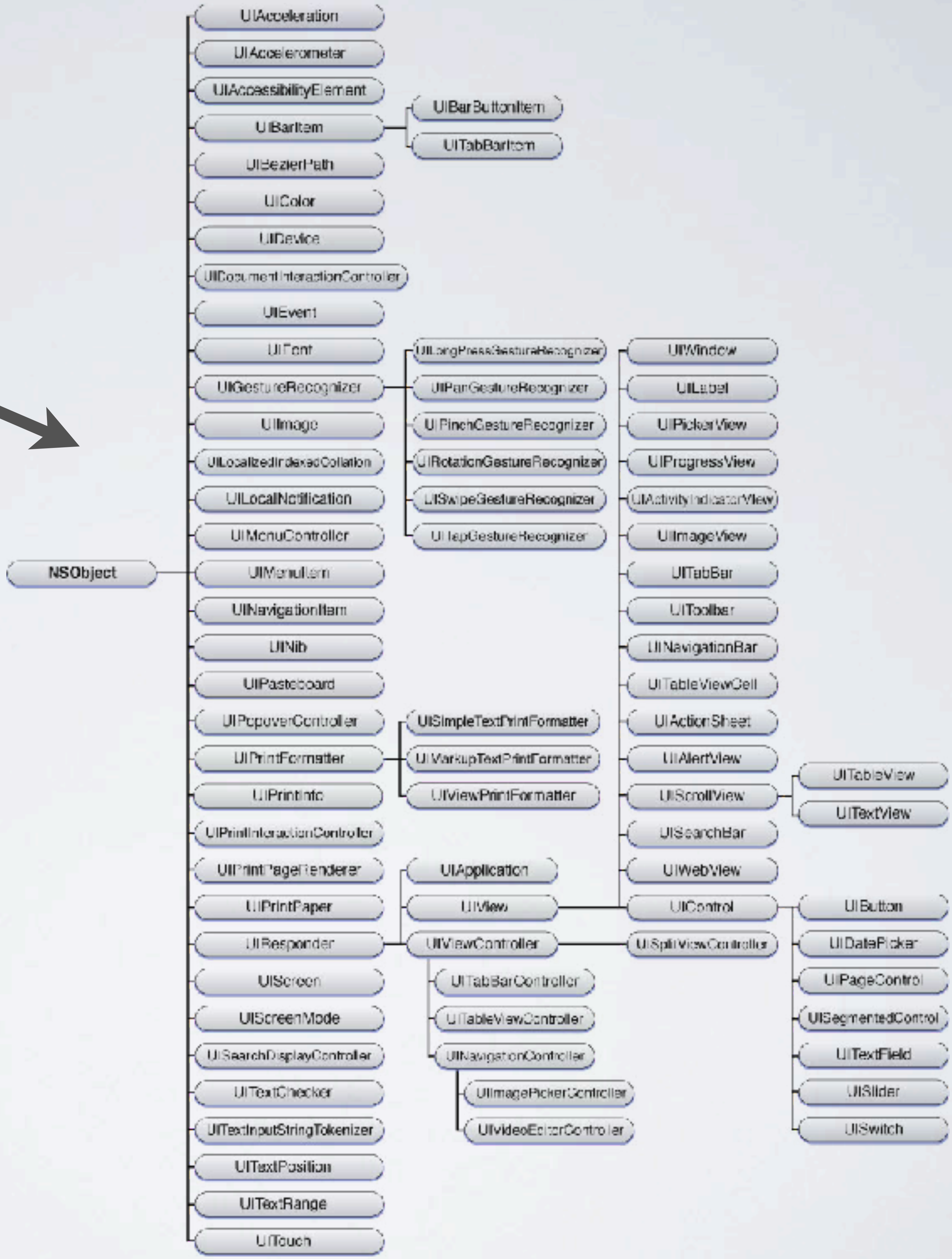
Core Animation

OpenGL ES / OpenGL

Core Graphics

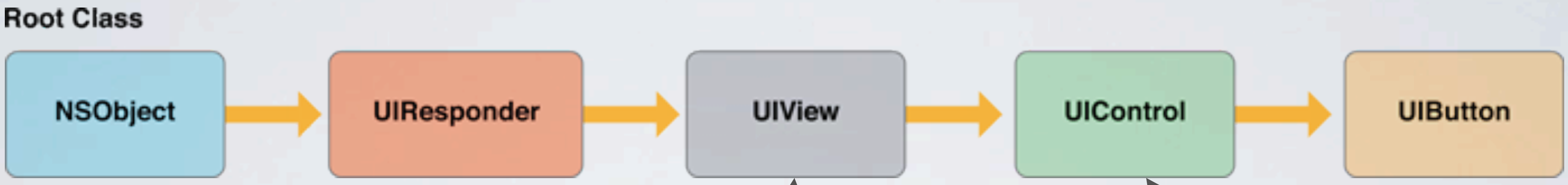
Graphics Hardware

These are the classes in UIKit



The position of a class in the hierarchy tells you about it's functionality

For example: UIButton



responds to and handles events

able to display itself

conveys user intent to the app

View Basics

- Rectangular area on the screen
- Draws itself
- Handles events (Subclasses UIResponder)
- Organized in a hierarchy of superview and subviews
- Superview retains its subviews
- Views can be temporarily hidden

```
[myView setHidden:YES];
```

Handy Data Types

- CGFloat - basically a **`float`**
- CGPoint - structure (X,Y)
- CGSize - structure (width, height)
- CGRect - structure (origin (X,Y),
  size (width , height))

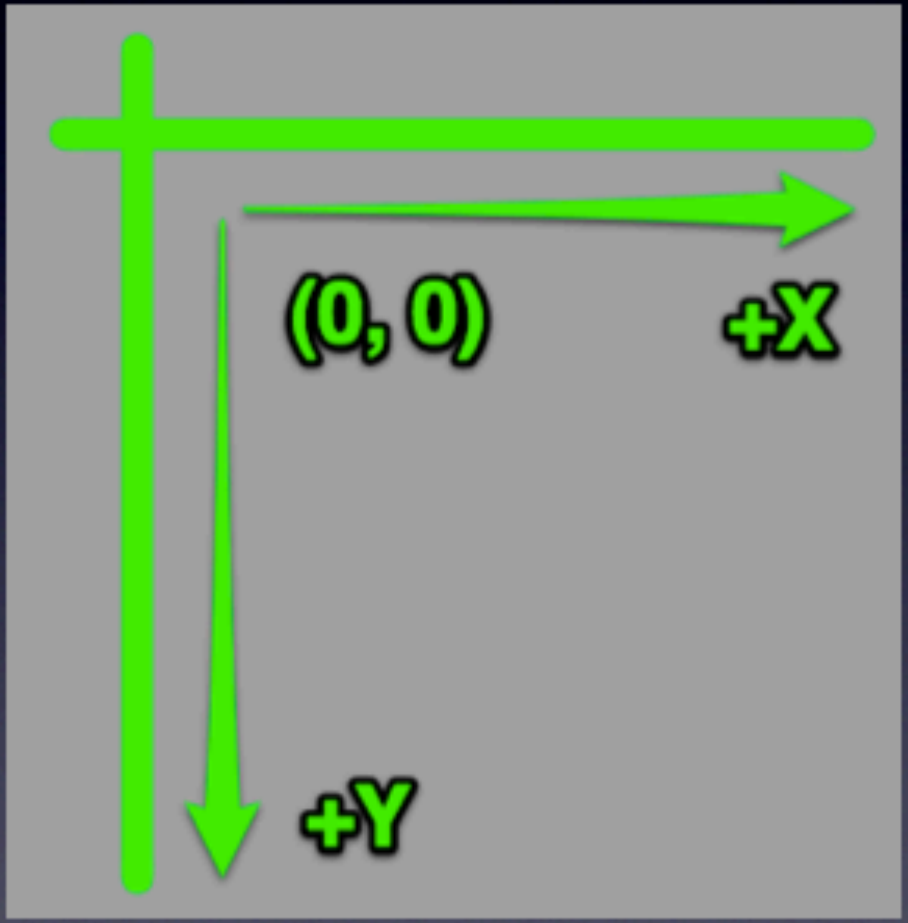
Handy Functions

- CGPoint CGPointMake(CGFloat X, CGFloat Y)
- CGSize CGSizeMake(CGFloat width, CGFloat height)
- CGRect CGRectMake(CGFloat X, CGFloat Y,
  CGFloat width, CGFloat height)
- CGRectContainsPoint, CGDivideRect, ...

See: *CGGeometry Reference* for list of functions

Coordinate System

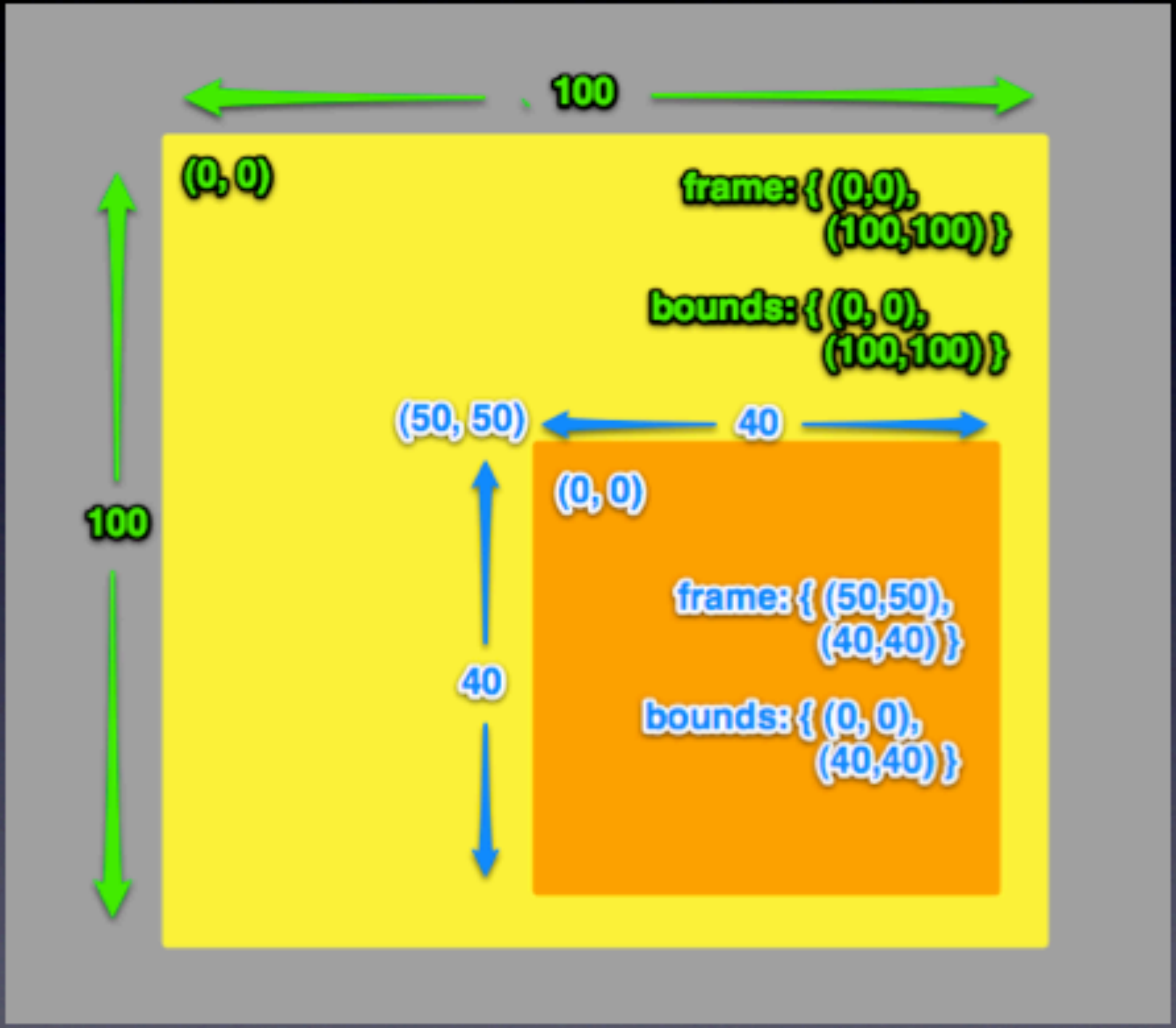
- Origin in upper left
- Values increase to the right and downwards
- Doesn’t change when the device is rotated



Location and Size

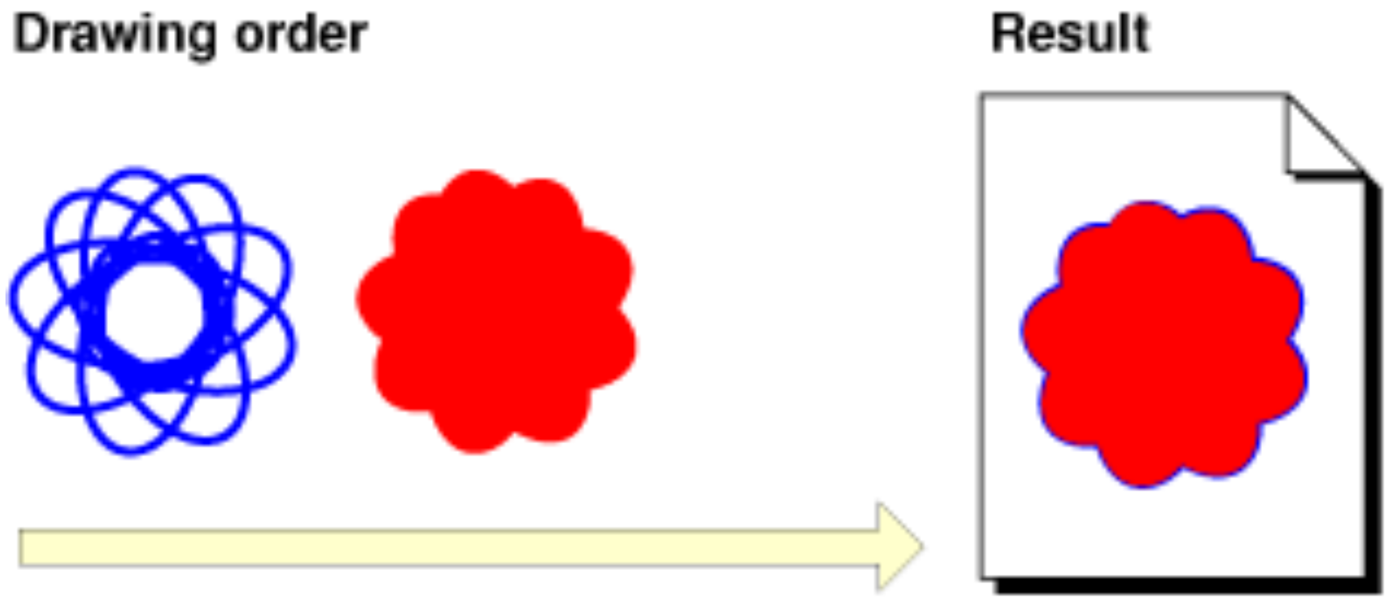
- Frame: location and size in superview's coordinates
- Bounds: location and size in local coordinates
- Place views using origin (set frame property) or set center property directly

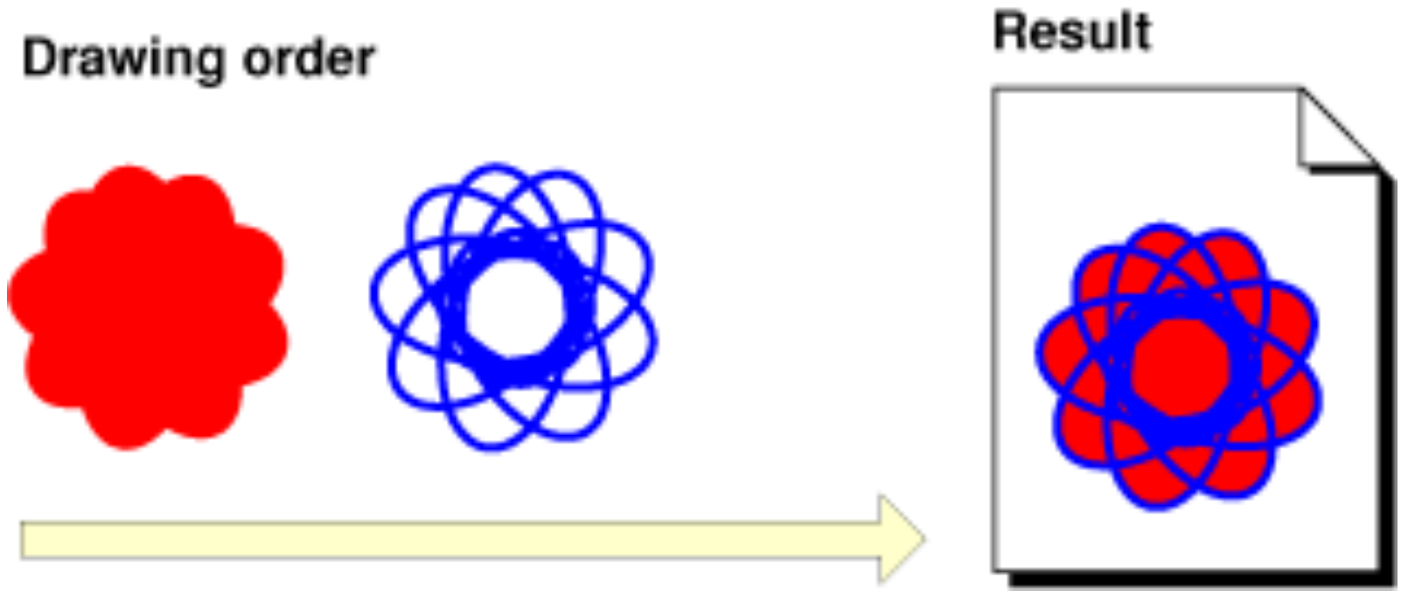
Frame and Bounds



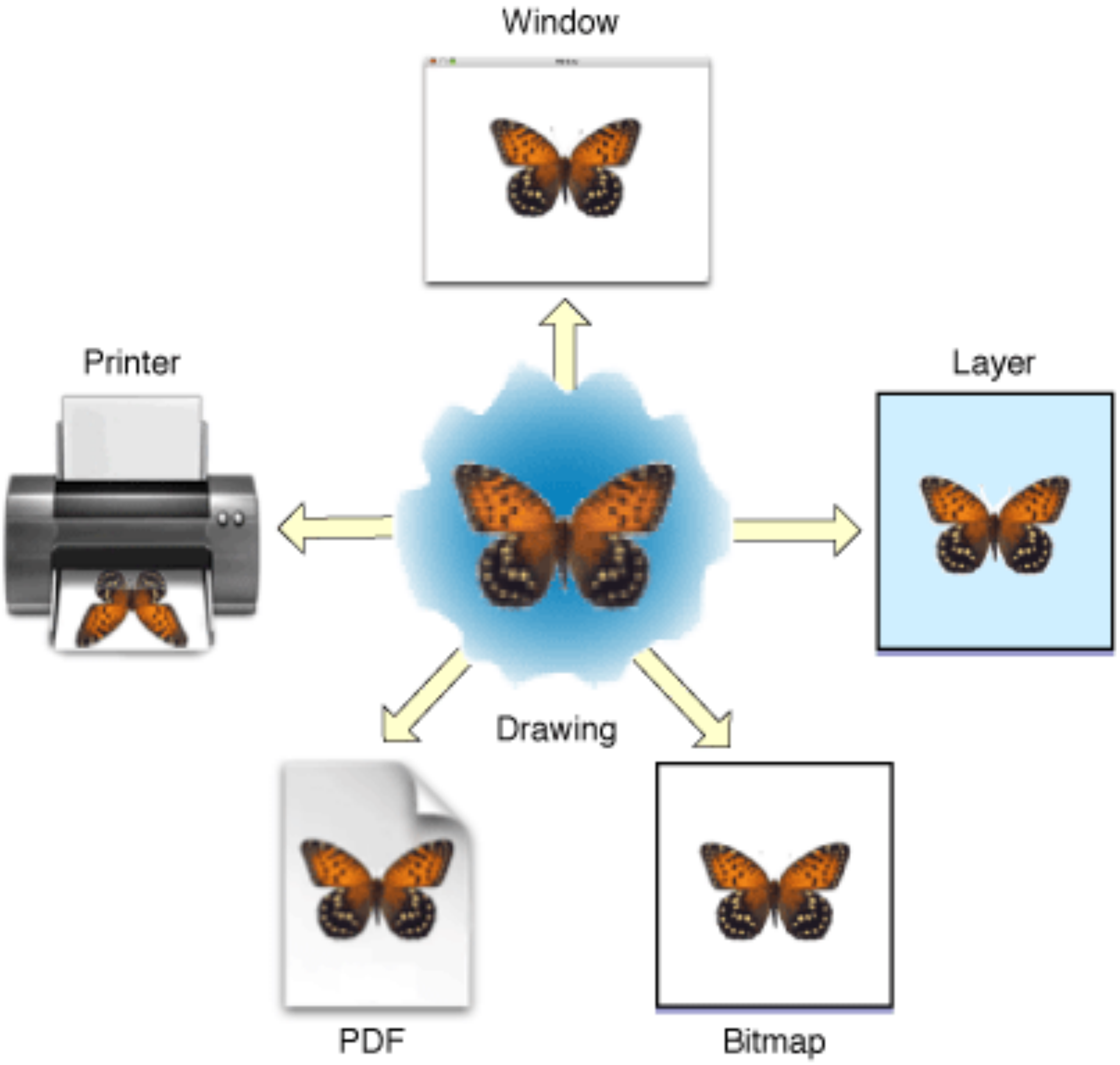
Quartz 2D

Quartz2D drawing uses a “painter’s model”

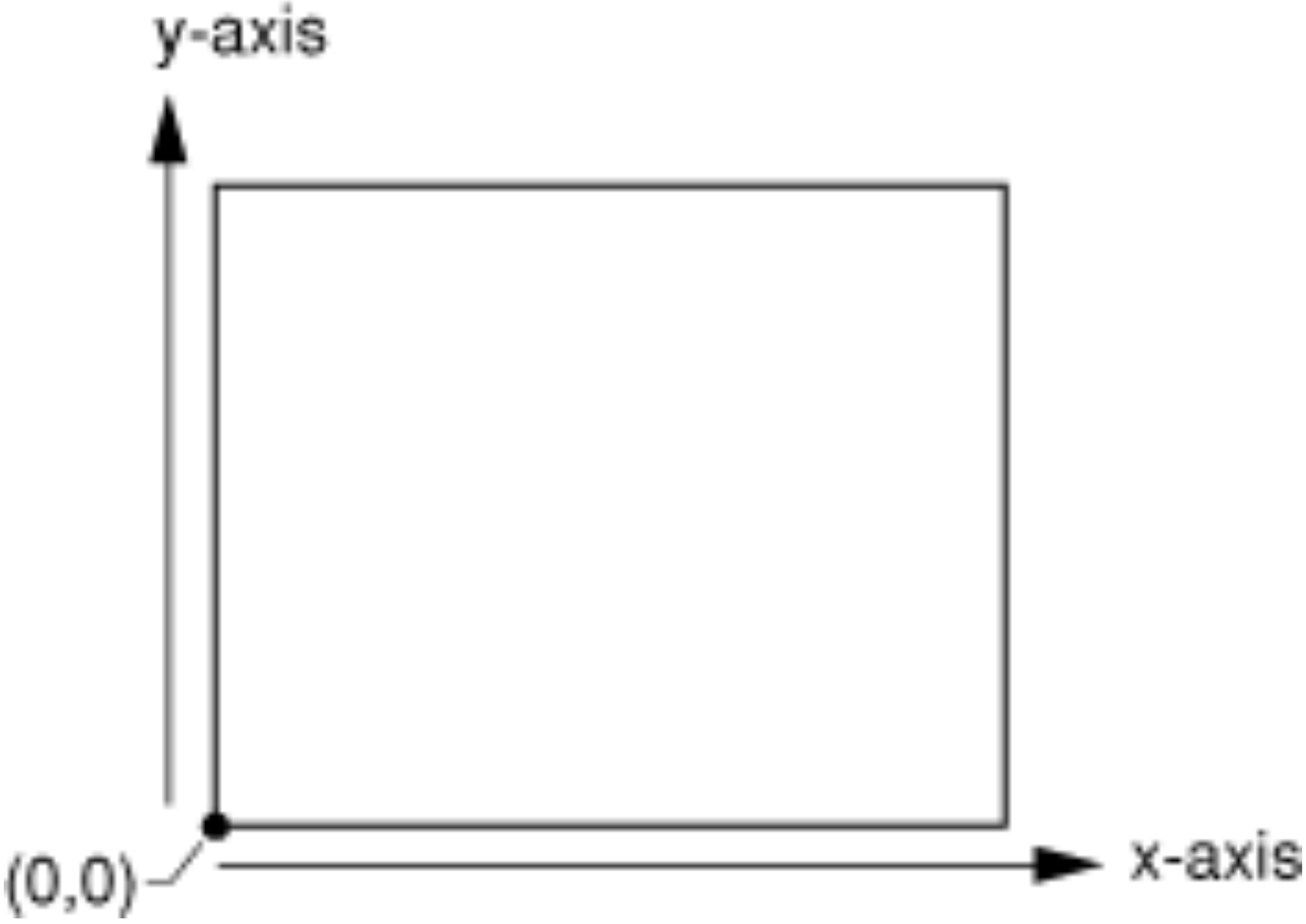




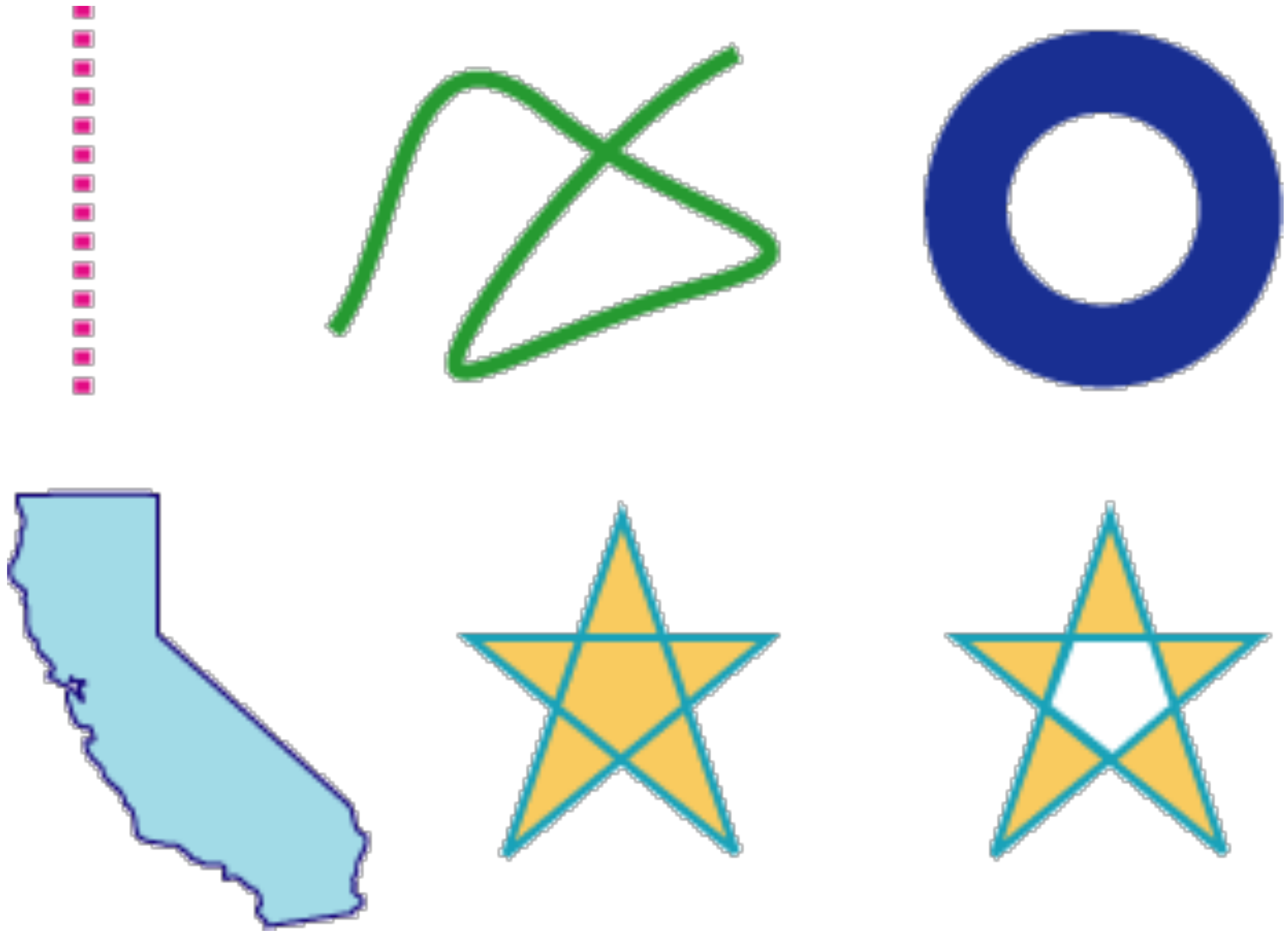
A Quartz drawing context can be many things



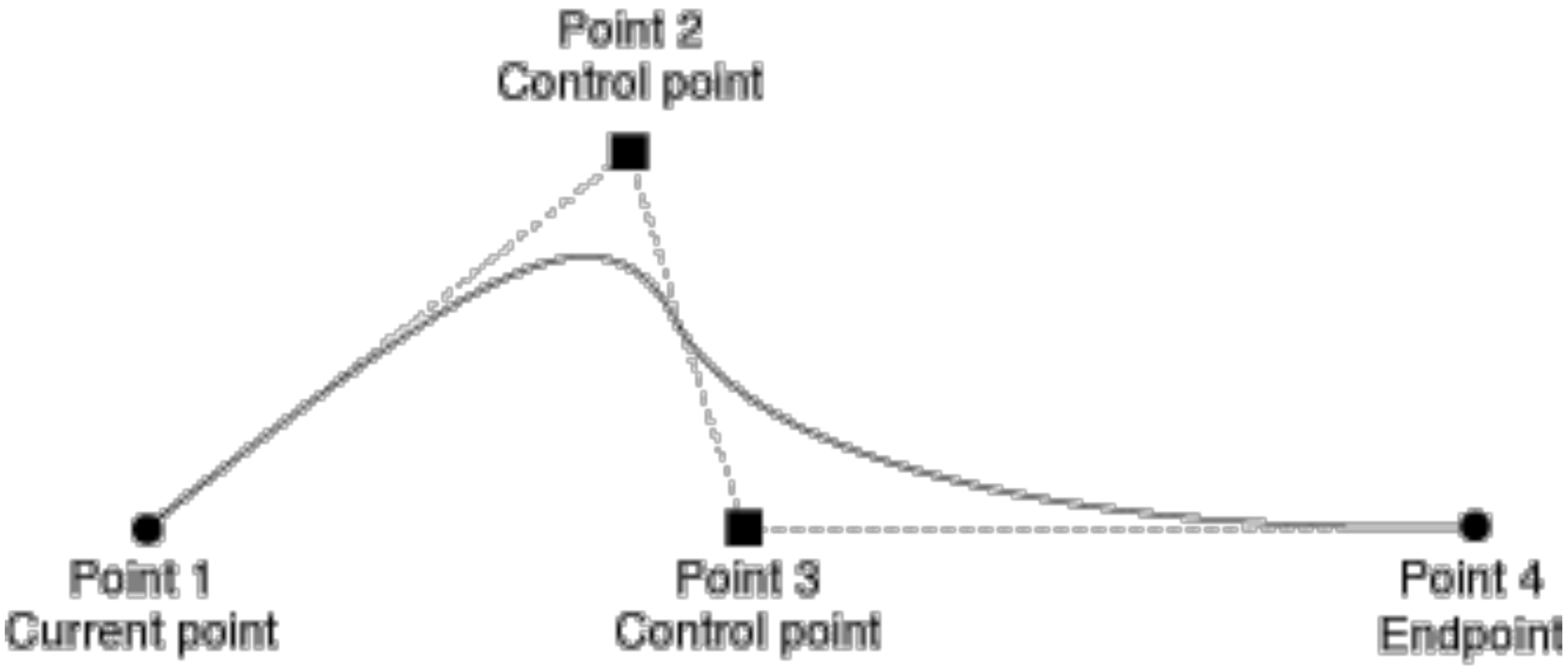
Quartz coordinates are the same as MacOS, but opposite from iOS



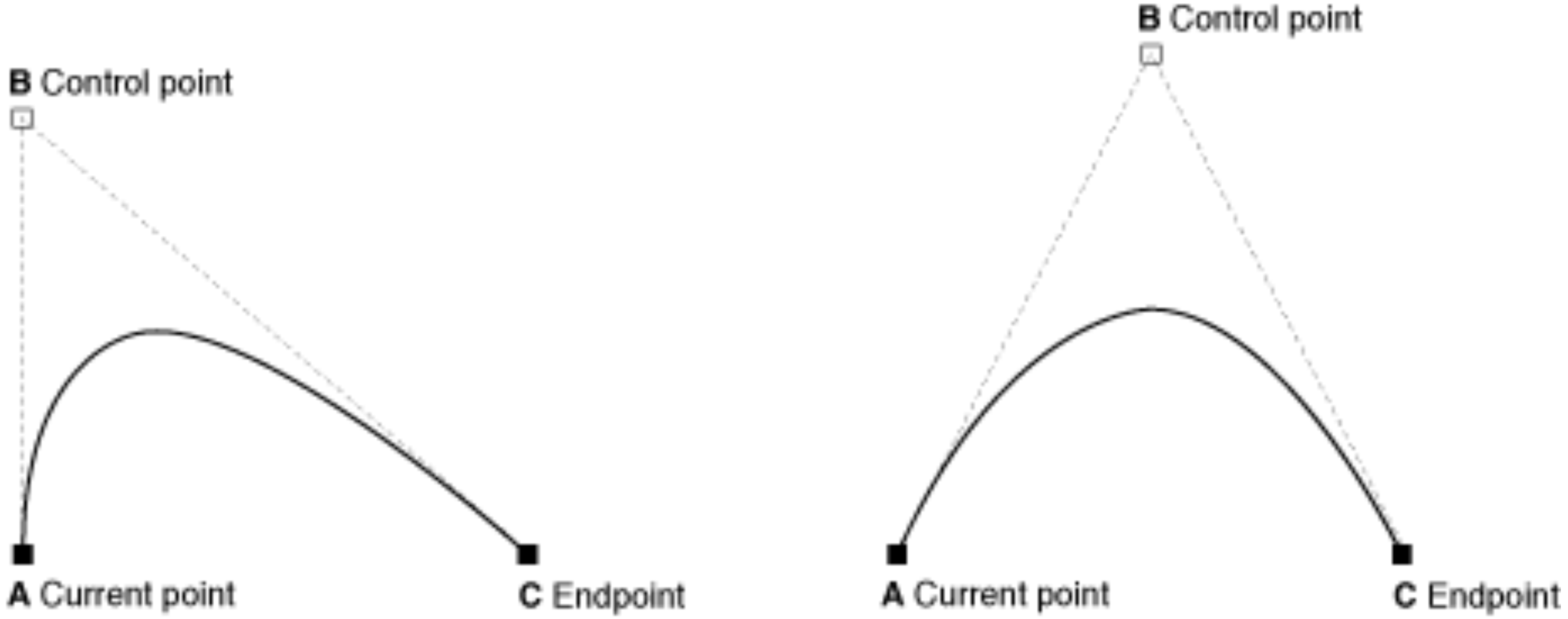
Path-based drawing in Quartz



Quartz Bezier curves



cubic



quadratic