

CS 696 Mobile Phone Application Development
Fall Semester, 2010
Doc 20 2D Graphics & Touch
Nov 9, 2010

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

References

Beginning iPhone 3 Development, Mark & LaMarche, Chapters 12 & 13

Stanford iPhone Course CS193P, Winter 2010, Lecture 5

Graphics

Quartz 2D (Core Graphics)

OpenGL ES (3D graphics)

Core Animation

Views

Draws content

Handles events

Subclass of UIResponder

Views arranged hierarchically

- every view has one superview

- every view has zero or more subviews

UIWindow

Views live inside of a window

UIWindow is a view

One UIWindow for an iPhone app

Structures & Functions

CGPoint

location in space: { x , y }

CGSize

dimensions: { width , height }

CGRect

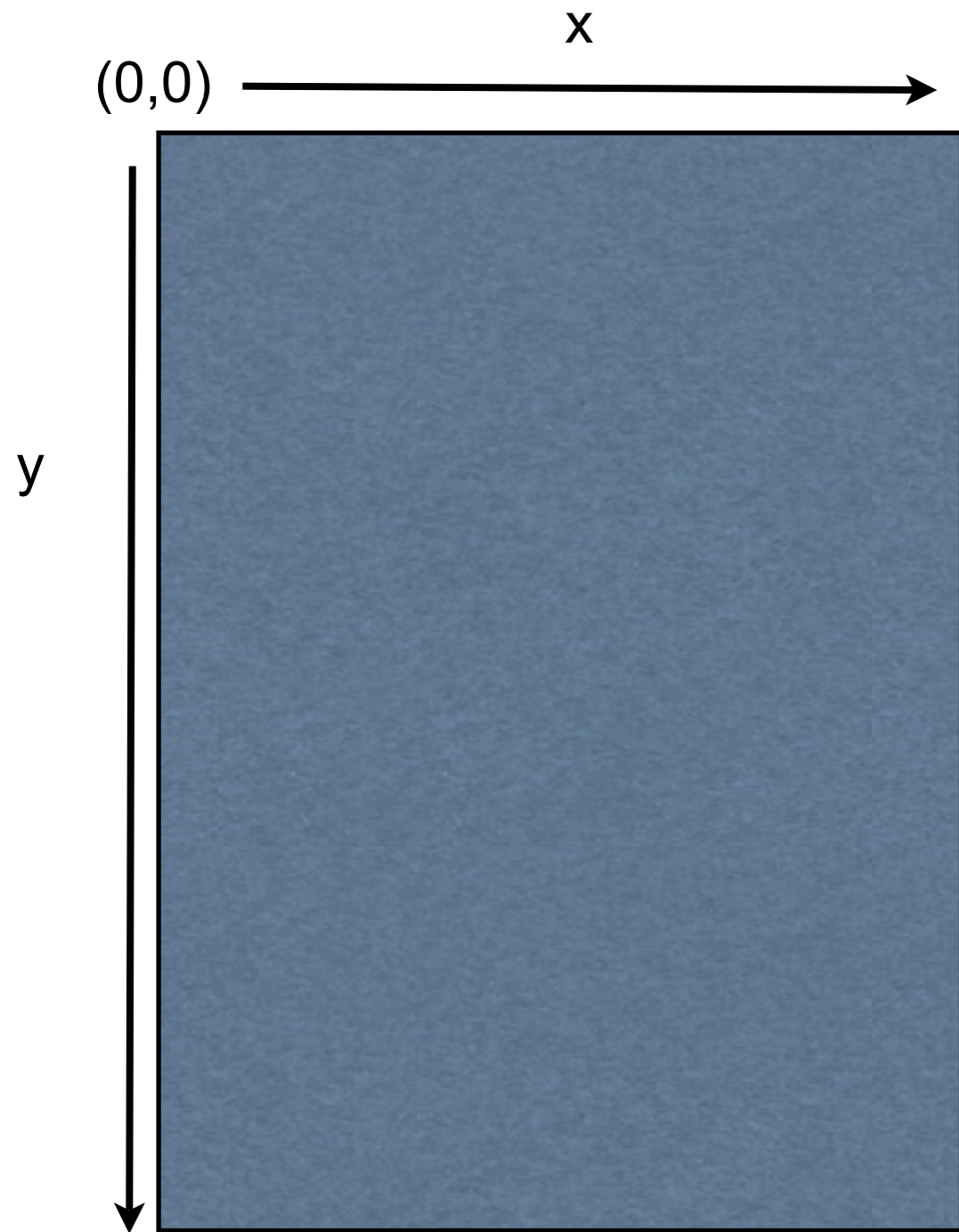
location and dimension: { origin , size }

CGPointMake (x, y)

CGSizeMake (width, height)

CGRectMake (x, y, width, height)

Quartz 2D Coordinates



Frame & Bounds

Both give location & size of View

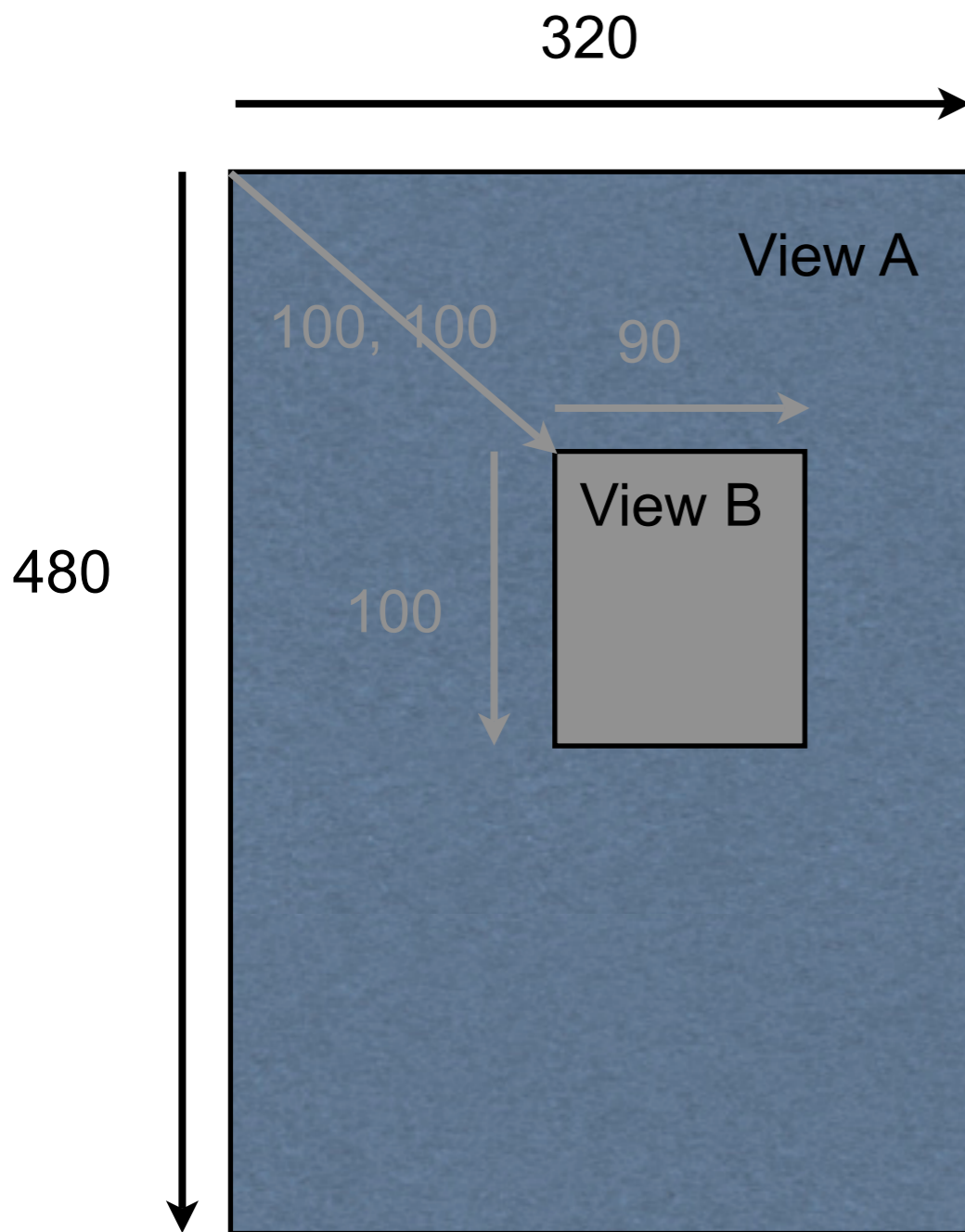
Frame

In superview coordinates

Computed

Bounds

In local coordinates



View A frame

size: 320 x 480

origin: 0, 0

View A bounds

size: 320 x 480

origin: 0, 0

View B frame

size: 90 x 100

origin: 100, 100

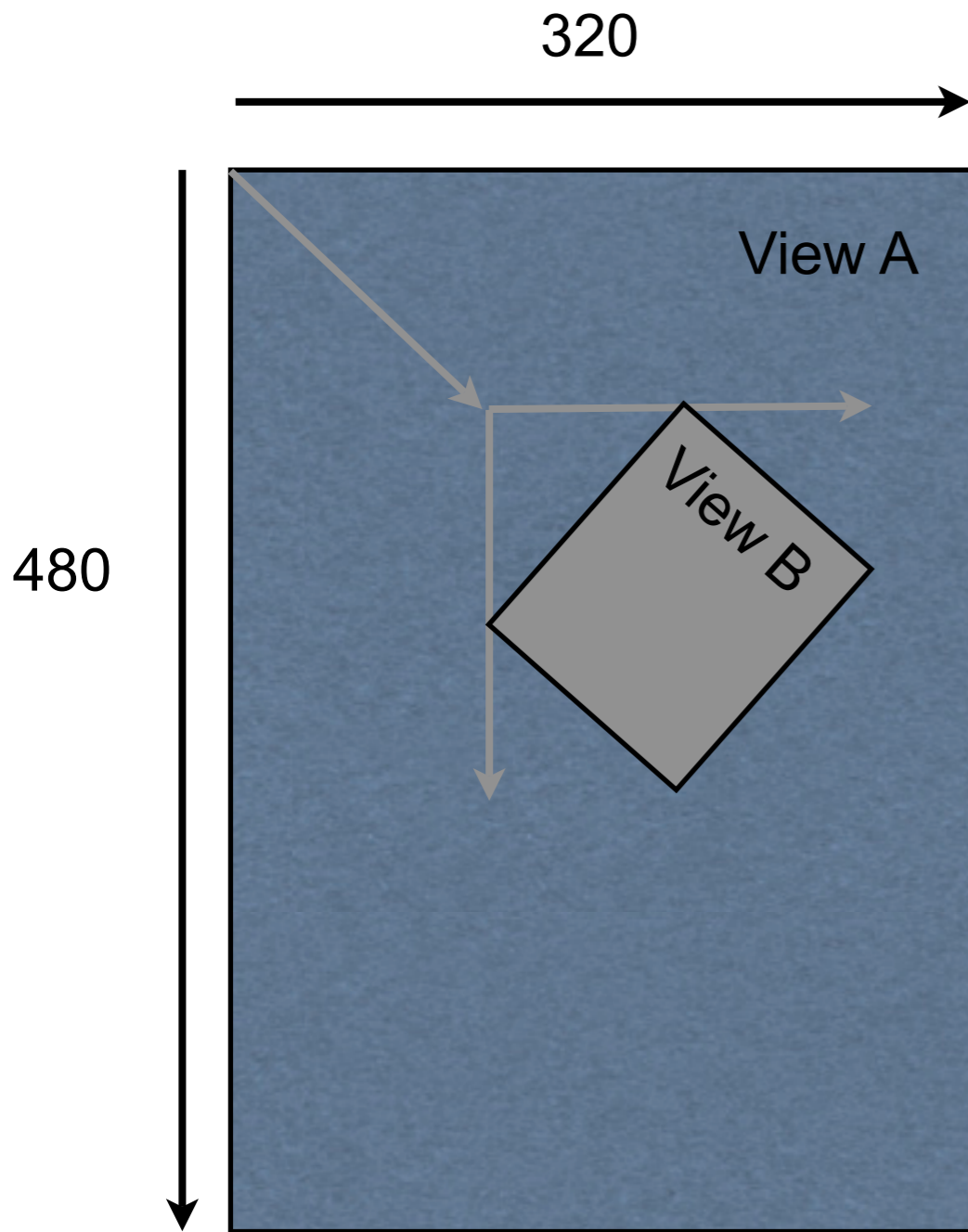
View B bounds

size: 90 x 100

origin: 0, 0

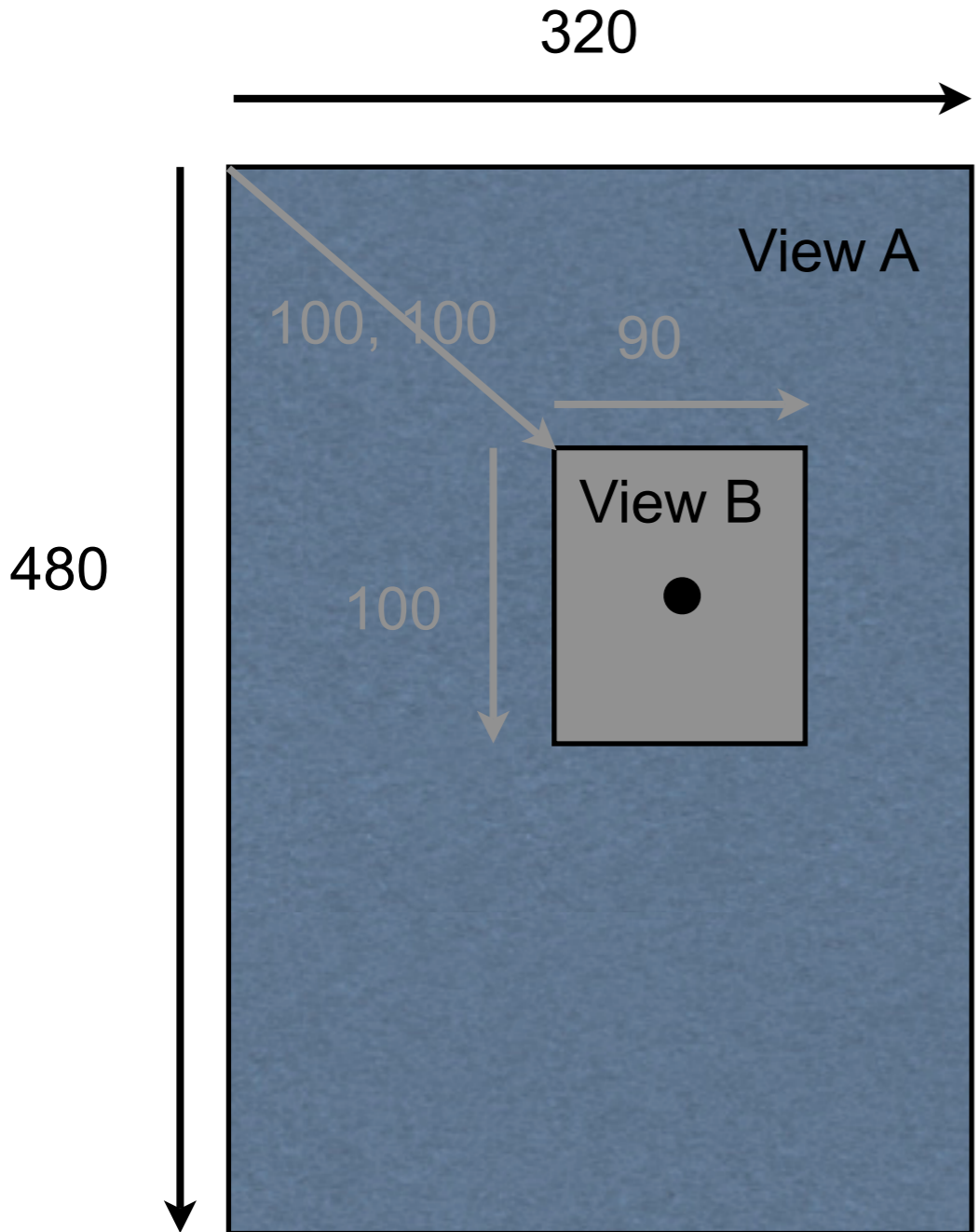
Frame

Smallest rectangle that contains view



Center

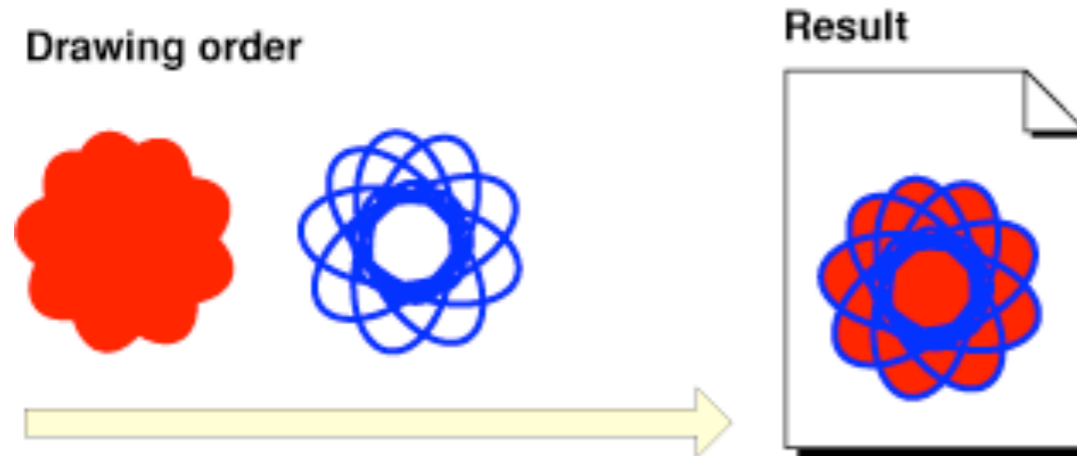
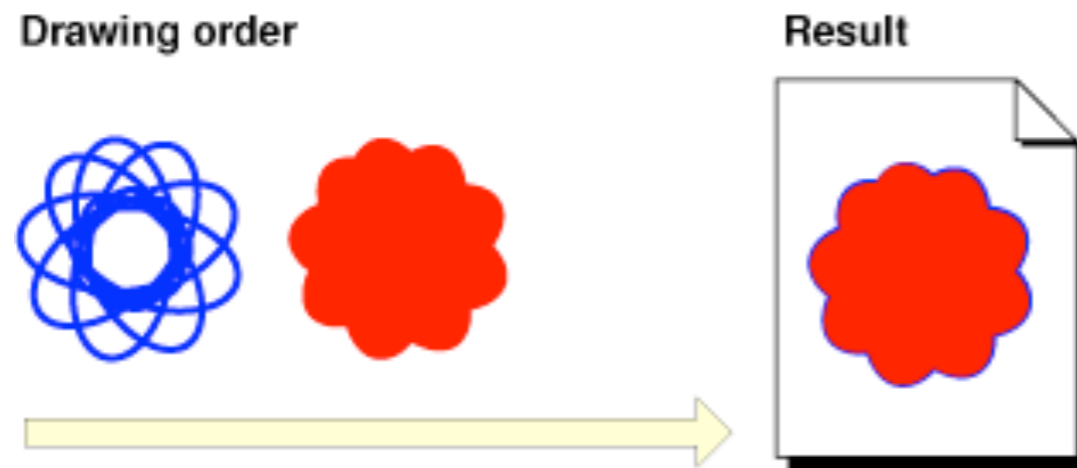
Given in superview's coordinates



View B Center
145, 150

Painter's Model

Drawing Order matters



CGContext

All drawing on CGContext

RGBA color

Paths, lines, ellipse, rectangle

Patterns

Shadows

Images

Gradients

Layers

Example Project

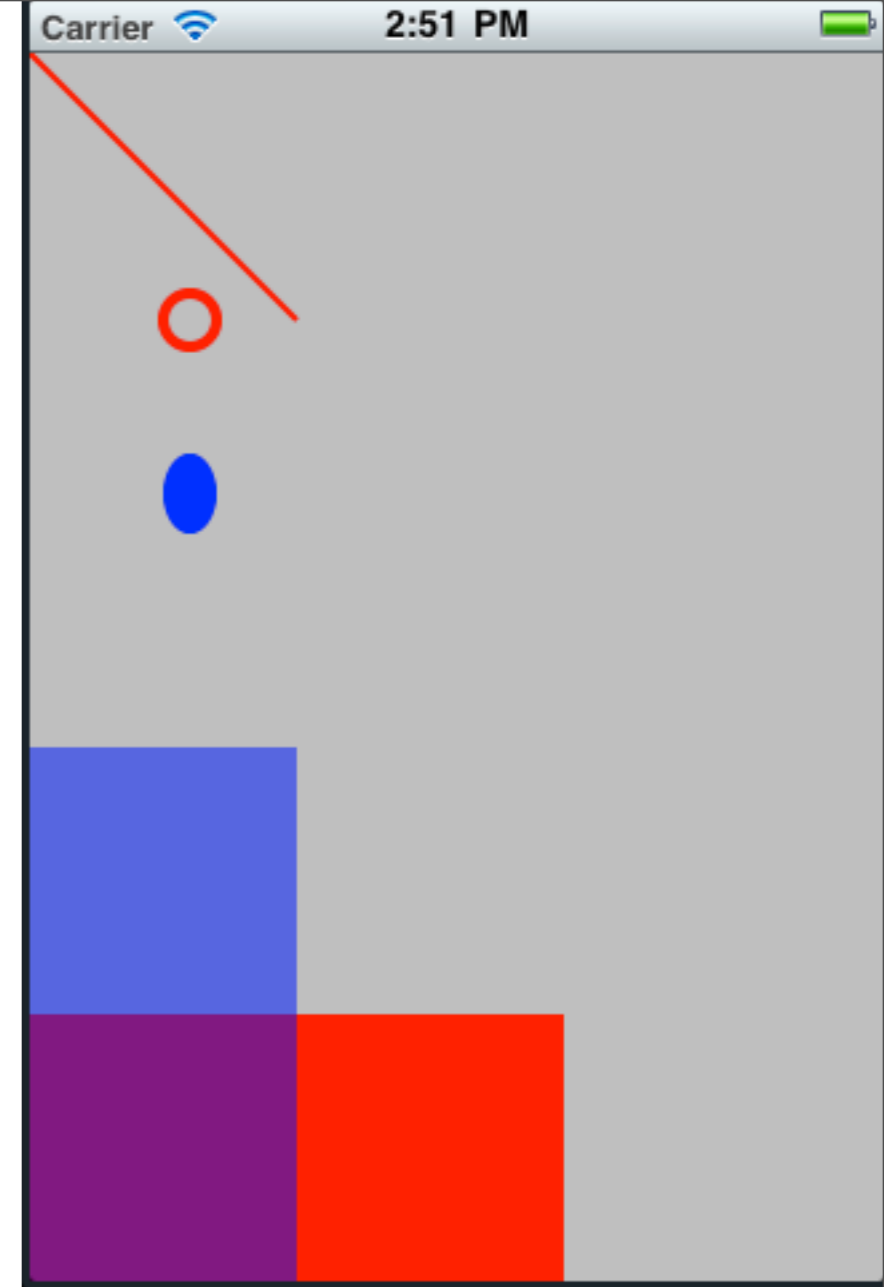
Add UIView in UIBuilder

Create subclass of UIBuilder

Set class of UIView in View to your subclass

In subclass override

- (void)drawRect:(CGRect)rect



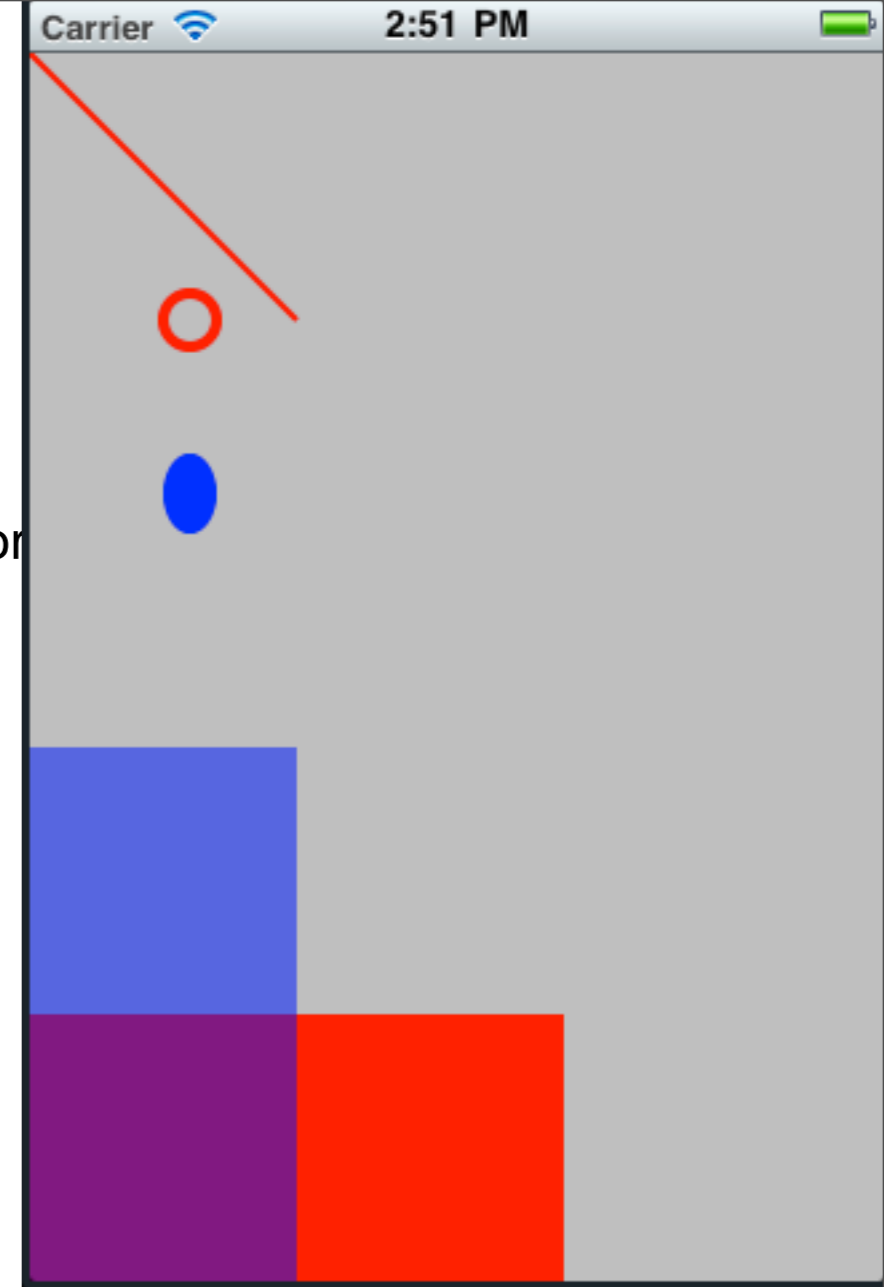
Drawing

```
- (void)drawRect:(CGRect)rect {
    CGContextRef context = UIGraphicsGetCurrentContext();
    CGContextSetLineWidth(context, 2.0);
    CGContextSetStrokeColorWithColor(context, [UIColor redColor].CGColor);
    CGContextMoveToPoint(context, 0.0f, 0.0f);
    CGContextAddLineToPoint(context, 100.0f, 100.0f);
    CGContextStrokePath(context);

    CGRect circleBoundry = CGRectMake(50, 90, 20,20);
    CGContextSetLineWidth(context, 4);
    CGContextAddEllipseInRect(context, circleBoundry);
    CGContextDrawPath(context, kCGPathStroke);

    CGRect ellipseBoundry = CGRectMake(50, 150, 20,30);
    CGContextAddEllipseInRect(context, ellipseBoundry);
    CGContextSetFillColorWithColor(context, [UIColor blueColor].CGColor);
    CGContextDrawPath(context, kCGPathFill);

    CGContextSetRGBFillColor (context, 1, 0, 0, 1);
    CGContextFillRect (context, CGRectMake (0, 360, 200, 100 ));
    CGContextSetRGBFillColor (context, 0, 0, 1, .5);
    CGContextFillRect (context, CGRectMake (0, 260, 100, 200));
}
```

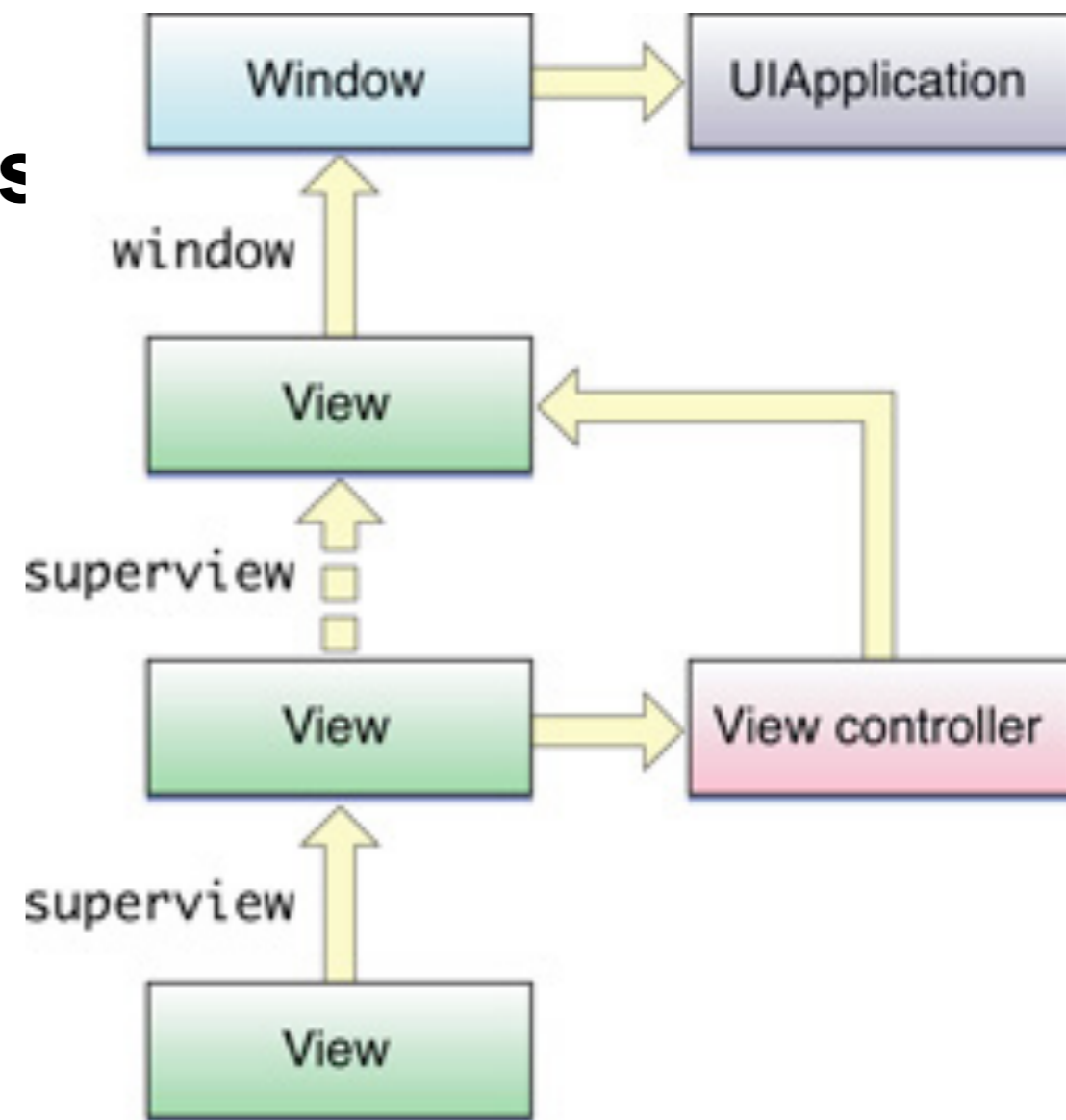


Touch Events

Responders

Event is sent to view it occurs in

If it does not handle event it is passed on to super view (or controller)



Responder methods

- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
- (void)touchesCancelled:(NSSet *)touches withEvent:(UIEvent *)event

touches contains on UITouch object for each finger on screen

UITouch

Getting the Location of Touches

- locationInView:
- previousLocationInView:
 - view (property)
 - window (property)

Getting Touch Attributes

- tapCount (property)
- timestamp (property)
- phase (property)

Getting a Touch Object's Gesture Recognizers

- gestureRecognizers (property)

Phases

- UITouchPhaseBegan
- UITouchPhaseMoved
- UITouchPhaseStationary
- UITouchPhaseEnded
- UITouchPhaseCancelled

UIEvent

Types

UIEventTypeTouches

UIEventTypeMotion

UIEventTypeRemoteControl

Getting the Touches for an Event

- allTouches
- touchesForView:
- touchesForWindow:

Subtypes

UIEventSubtypeNone

UIEventSubtypeMotionShake

UIEventSubtypeRemoteControlPlay

UIEventSubtypeRemoteControlPause

etc.

Getting Event Attributes

timestamp (property)

Getting the Event Type

type (property)

subtype (property)

Getting the Touches for a Gesture Recognizer

- touchesForGestureRecognizer:

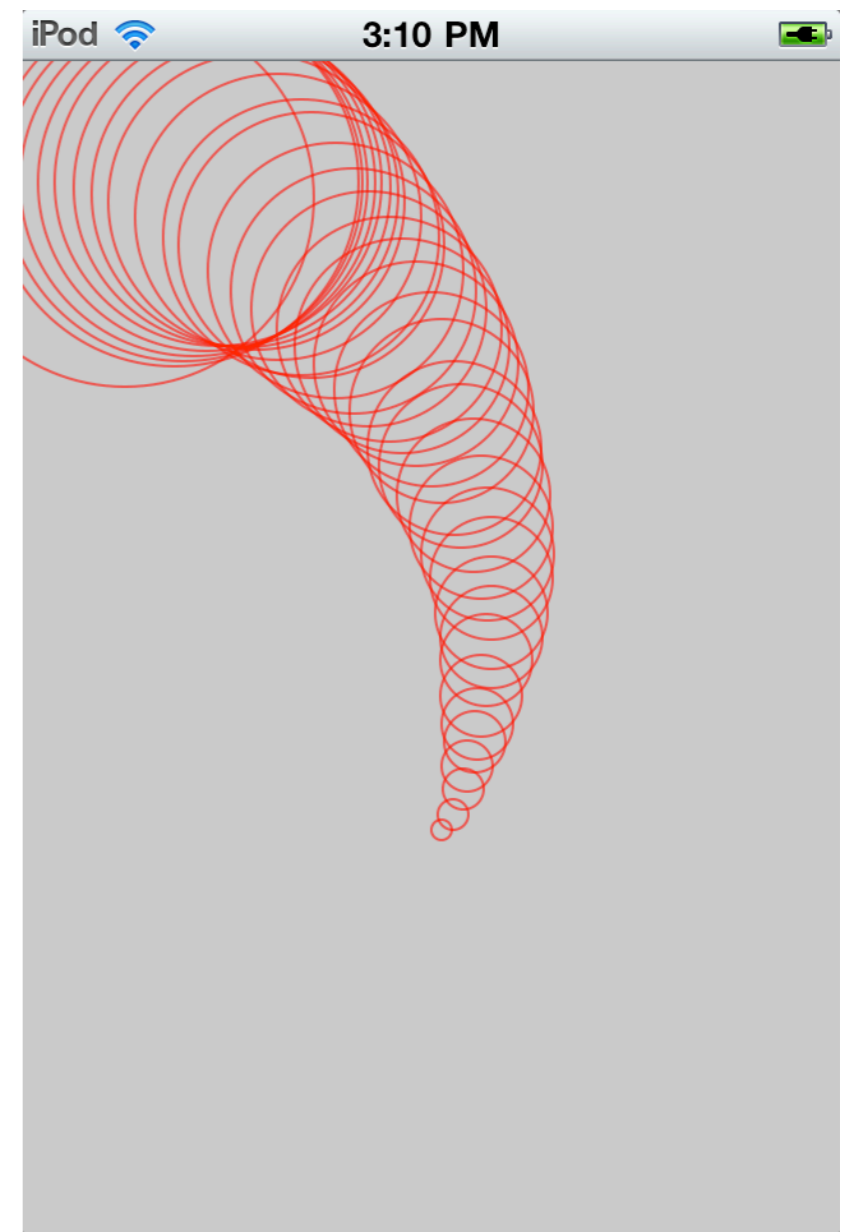
Example

Track user's finger on screen

Draw circles on touch events

Circles get bigger the get older

When user stops touching circles fade



Project Setup

One UIView in view

Its class is Touch

```
#import <Foundation/Foundation.h>
```

```
@interface TouchView : UIView {  
    CGPoint points[100];  
    int numberOfPoints;  
    float alpha;  
}
```

```
- (void) fade;  
@end
```

Drawing Circles

@implementation TouchView

- (void)drawRect:(CGRect)rect {

CGContextRef context = UIGraphicsGetCurrentContext();

CGContextSetStrokeColorWithColor(context,

[UIColor colorWithRed: 1.0 green: 0 blue: 0.0 alpha:alpha].CGColor);

CGContextSetLineWidth(context, 1);

for (int k=0;k < numberOfPoints;k++) {

int size = 4*(numberOfPoints - k + 1);

CGRect circleBoundry = CGRectMake(points[k].x - size/2,points[k].y - size/2,
size,size);

CGContextAddEllipseInRect(context, circleBoundry);

CGContextDrawPath(context, kCGPathStroke);

}

}

First Touch

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {  
    numberOfPoints = 0;  
    alpha = 1.0;  
    CGPoint location = [[touches anyObject] locationInView: self];  
    points[numberOfPoints++] = location;  
}
```

Moving

```
- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event {  
    if (numberOfPoints > 99) {  
        return;  
    }  
    CGPoint location = [[touches anyObject] locationInView: self];  
    points[numberOfPoints++] = location;  
    [self setNeedsDisplay];  
}
```


Touch Ends

```
- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event {  
    [self fade];  
}
```

fading

```
- (void) fade {  
    if (alpha <= 0.01) {  
        alpha = 0.0;  
        [self setNeedsDisplay];  
        return;  
    };  
    alpha = alpha * 0.92;  
    [self setNeedsDisplay];  
    [self performSelector:@selector(fade) withObject:nil afterDelay:0.2];  
}
```