

iPhone App Dev

Lesson 3

Practice

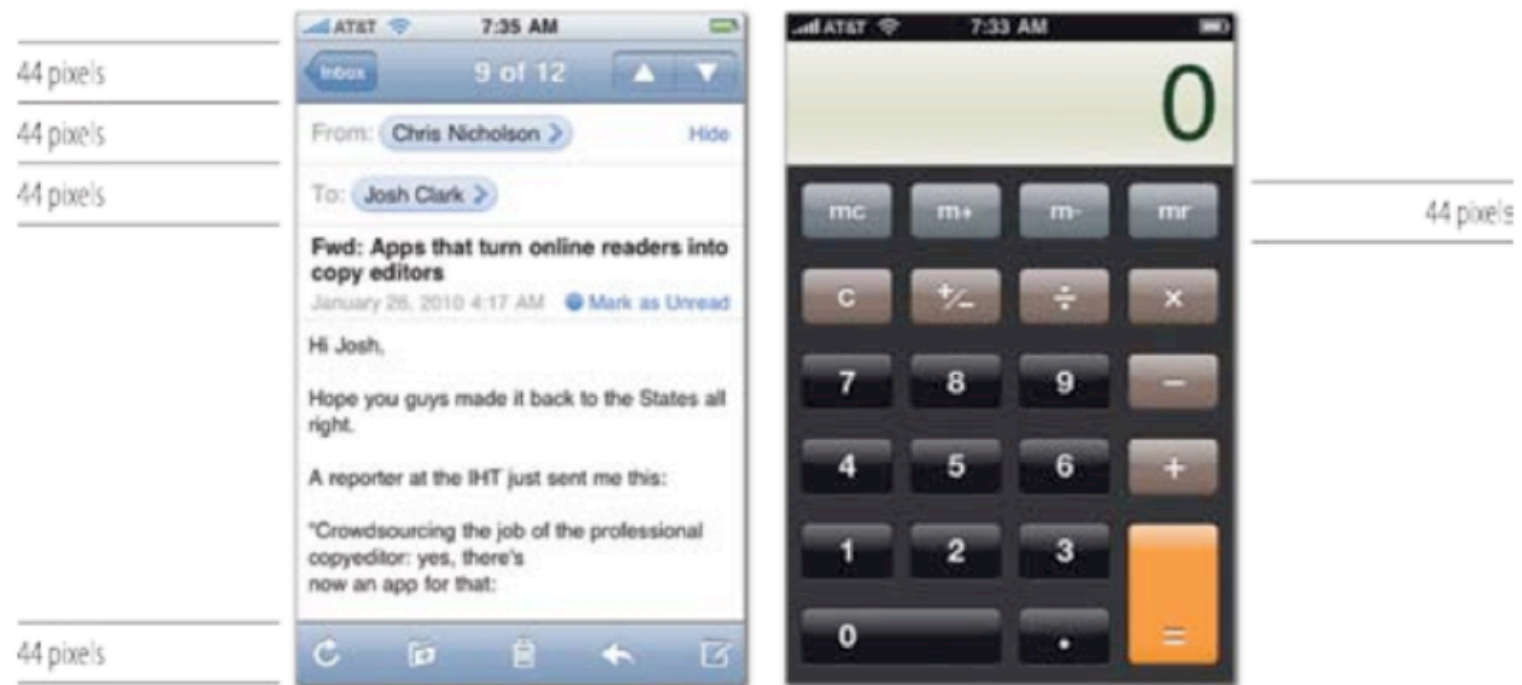
- ✓ Design an utility app.
- ✓ Present your idea to the class.

Summary

- The 44 Pixels
- Detecting Touches
- Multi Touches
- Gesture Recognition
- Object Memory Management
- Instance Property

Button Tap Area

The 44 pixels



Graph from TapWorthy book.

The 44 pixels



Graph from TapWorthy book.

Less than 44 pixels



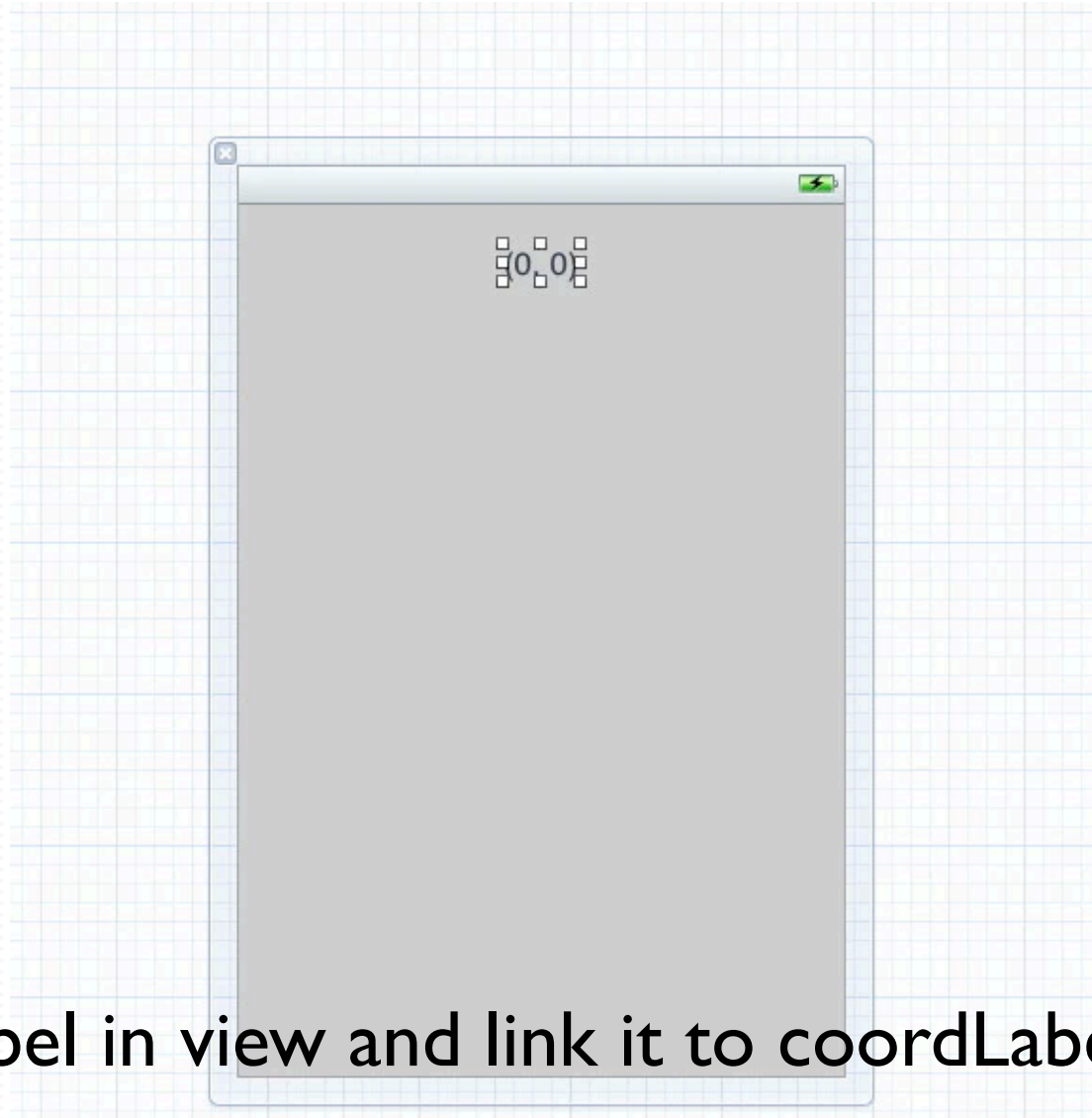
Graph from TapWorthy book.

Detecting Touches

Detect Touches

```
1 - (void) touchesBegan: (NSSet *) touches withEvent: (UIEvent *) event;  
2 - (void) touchesMoved: (NSSet *) touches withEvent: (UIEvent *) event;  
3 - (void) touchesEnded: (NSSet *) touches withEvent: (UIEvent *) event;
```

Detect Touches



Put a label in view and link it to coordLabel outlet.

Detect Touches

```
1 - (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
2 {
3     UITouch *touch = [touches anyObject];
4
5     CGPoint point = [touch locationInView:self.view];
6
7     NSLog(@"Get a touch began at (%.2f, %.2f)", point.x, point.y);
8
9     self.coordLabel.text = [NSString stringWithFormat:@"(%.2f, %.2f)",
point.x, point.y];
10 }
```

Detect Touches

```
1 - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
2 {
3     UITouch *touch = [touches anyObject];
4
5     CGPoint point = [touch locationInView:self.view];
6
7     NSLog(@"Get a touch began at (%.2f, %.2f)", point.x, point.y);
8
9     self.coordLabel.text = [NSString stringWithFormat:@"(%.2f, %.2f)",
point.x, point.y];
10 }
```

Detect Touches

```
1 - (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
2 {
3     self.coordLabel.text = @"Not Touching";
4 }
```

NSSet

- Major methods

```
1 - (NSArray *)allObjects;  
2 - (id)anyObject;
```

NSSet

- Major methods

```
1 @interface NSMutableSet : NSSet
2
3 - (void) addObject: (id) object;
4 - (void) removeObject: (id) object;
5
6 @end
```

NSSet

- Major methods

```
1 @interface NSMutableSet (NSExtendedMutableSet)
2
3 - (void) addObjectsFromArray: (NSArray *) array;
4 - (void) intersectSet: (NSSet *) otherSet;
5 - (void) minusSet: (NSSet *) otherSet;
6 - (void) removeAllObjects;
7 - (void) unionSet: (NSSet *) otherSet;
8
9 - (void) setSet: (NSSet *) otherSet;
10
11 @end
```


CGGeometry

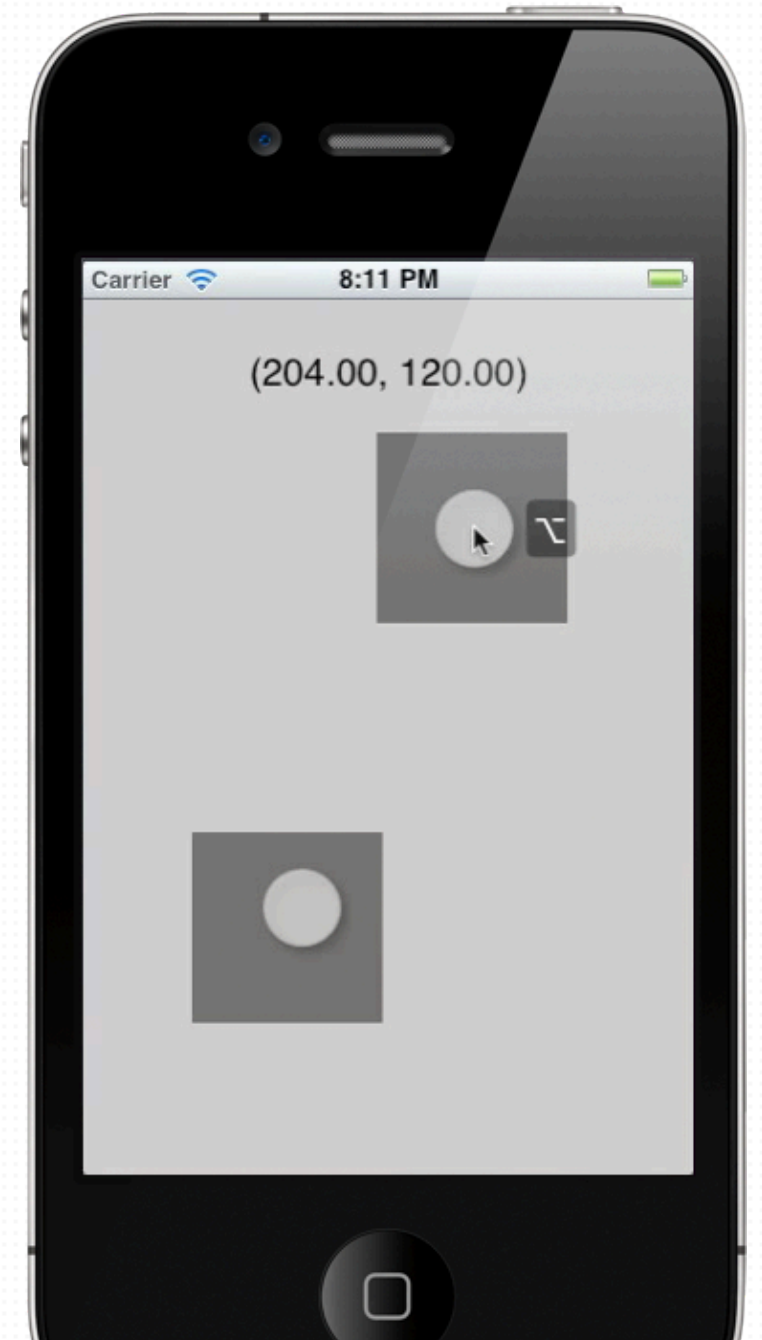
- Primitives and Functions for geometric.
- CGPoint
- CGRect
- CGSize

CGGeometry

- Primitives and Functions for geometric.
- CGPointMake
- CGRectMake
- CGSizeMake

Multi Touches

Multi Touches Drag-n-Drop



Multi Touches Dragging

- Two variables to remember the touches pointer

```
1 @interface ViewController()  
2     @property (nonatomic, strong) UITouch *touchForBox1;  
3     @property (nonatomic, strong) UITouch *touchForBox2;  
4 @end
```

Multi Touches Dragging

- Determine which box we are touching

```
1 - (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event
2 {
3     for(UITouch *touch in touches)
4     {
5         CGPoint point = [touch locationInView:self.view];
6         self.coordLabel.text = [NSString stringWithFormat:@"(%.2f, %.2f)",
point.x, point.y];
7         if (CGRectContainsPoint(box.frame, point))
8         {
9             self.box.center = point;
10            self.touchForBox1 = touch;
11        } else if (CGRectContainsPoint(box2.frame, point))
12        {
13            self.box2.center = point;
14            self.touchForBox2 = touch;
15        }
16    }
17 }
```

Multi Touches Dragging

- Move the box if the touch pointer matches the touchBegan one.

```
1 - (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event
2 {
3     for(UITouch *touch in touches)
4     {
5         CGPoint point = [touch locationInView:self.view];
6         self.coordLabel.text = [NSString stringWithFormat:@"%(.2f, %.2f)",
point.x, point.y];
7
8         if (self.touchForBox1 == touch)
9         {
10             self.box.center = point;
11         } else if (self.touchForBox2 == touch)
12         {
13             self.box2.center = point;
14         }
15     }
16 }
```

Multi Touches Dragging

- Reset the pointer after touch ended

```
1 - (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
2 {
3     self.coordLabel.text = @"Not Touching";
4
5     for (UITouch *touch in touches)
6     {
7         CGPoint point = [touch locationInView:self.view];
8
9         self.coordLabel.text = [NSString stringWithFormat:@"(%.2f, %.2f)",
point.x, point.y];
10
11         if (self.touchForBox1 == touch)
12         {
13             self.touchForBox1 = nil;
14         }
15
16         if (self.touchForBox2 == touch)
17         {
18             self.touchForBox2 = nil;
19         }
22     }
23 }
```


Gesture Recognition

Gesture Recognizer

- Tap

```
1 UITapGestureRecognizer *tapGesture = [[UITapGestureRecognizer alloc]
initWithTarget:self action:@selector(tapGestureHandler)];
2 tapGesture.numberOfTapsRequired = 2;
3 [button addGestureRecognizer:tapGesture];
```

```
1 - (void)tapGestureHandler:(UIGestureRecognizer*)gestureRecognizer
2 {
3     NSLog(@"Tap Gesture Triggered. %d fingers tapped.",
gestureRecognizer.numberOfTouches);
4 }
```

@selector

@selector(helloWorld)

- (void)helloWorld

@selector(helloWorld:)

- (void)helloWorld:(id)param

@selector

@selector(sendMessage:to:)

- (void)sendMessage:(id)msg to:(id)somebody

Gesture Recognizer

- Long Press

```
1 UILongPressGestureRecognizer *longPressGesture =  
[[UILongPressGestureRecognizer alloc] initWithTarget:self  
action:@selector(longPressHandler:)];  
2 [button addGestureRecognizer:longPressGesture];
```

Gesture Recognizer

- Swipe

```
1 UISwipeGestureRecognizer *swipeGesture = [[UISwipeGestureRecognizer alloc]
initWithTarget:self action:@selector(swipeGestureHandler)];
2 swipeGesture.direction = UISwipeGestureRecognizerDirectionLeft;
3 [self.view addGestureRecognizer:swipeGesture];
```

Gesture Recognizer

- Rotation

```
1 UIRotationGestureRecognizer *rotationGesture =  
[[UIRotationGestureRecognizer alloc] initWithTarget:self  
action:@selector(rotationGestureHandler)];  
2 [self.view addGestureRecognizer:rotationGesture];
```

```
1 - (void)rotationGestureHandler:  
(UIRotationGestureRecognizer*)gestureRecognizer  
2 {  
3     float degree = gestureRecognizer.rotation * 180 / M_PI;  
4     NSLog(@"Rotating: %fdeg", degree);  
5 }
```

Gesture Recognizer

- Pan

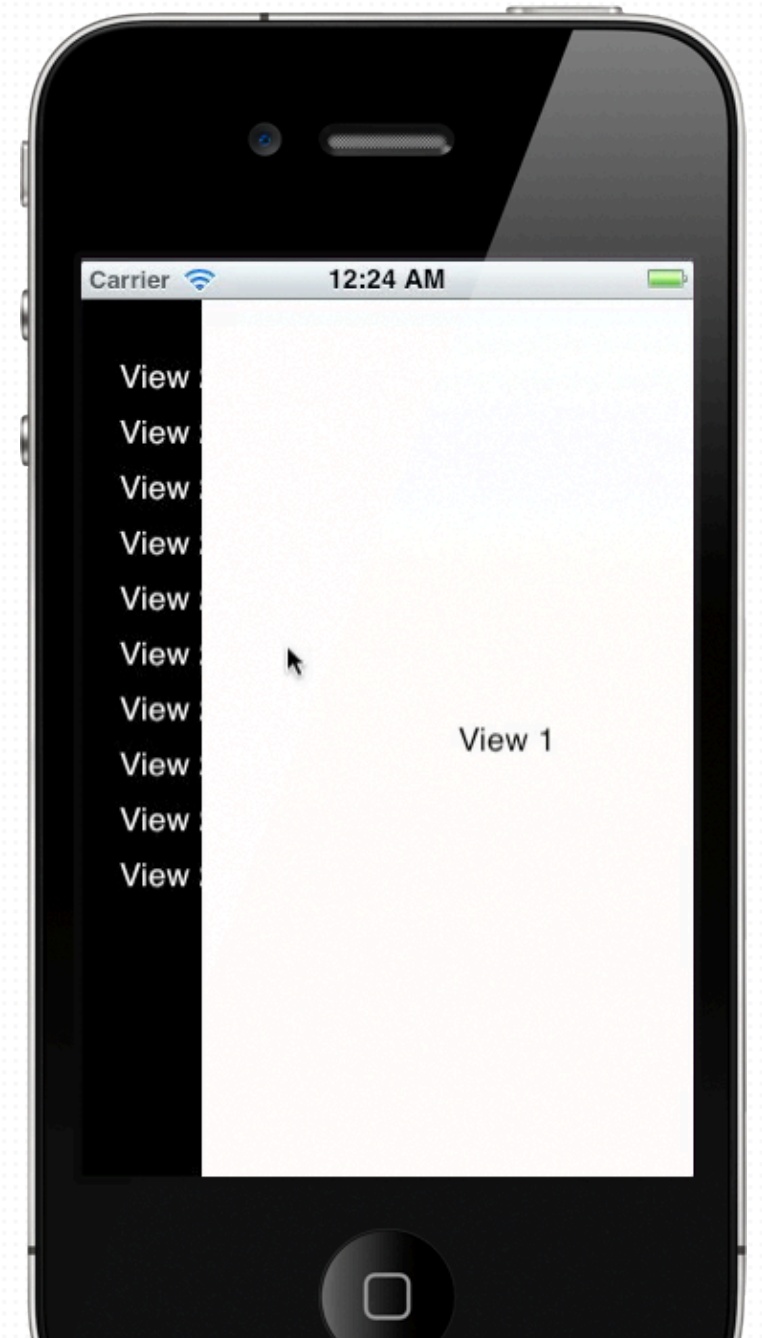
```
1 UIPanGestureRecognizer *panGesture = [[UIPanGestureRecognizer alloc]
initWithTarget:self action:@selector(panGestureHandler)];
2 [self.view addGestureRecognizer:panGesture];
```

```
1 - (void)panGestureHandler:(UIPanGestureRecognizer*)gestureRecognizer
2 {
3     NSString *translation = NSStringFromCGPoint([gestureRecognizer
translationInView:self.view]);
4     NSString *velocity = NSStringFromCGPoint([gestureRecognizer
velocityInView:self.view]);
5     NSLog(@"translation: %@, velocity: %@", translation, velocity);
6 }
```

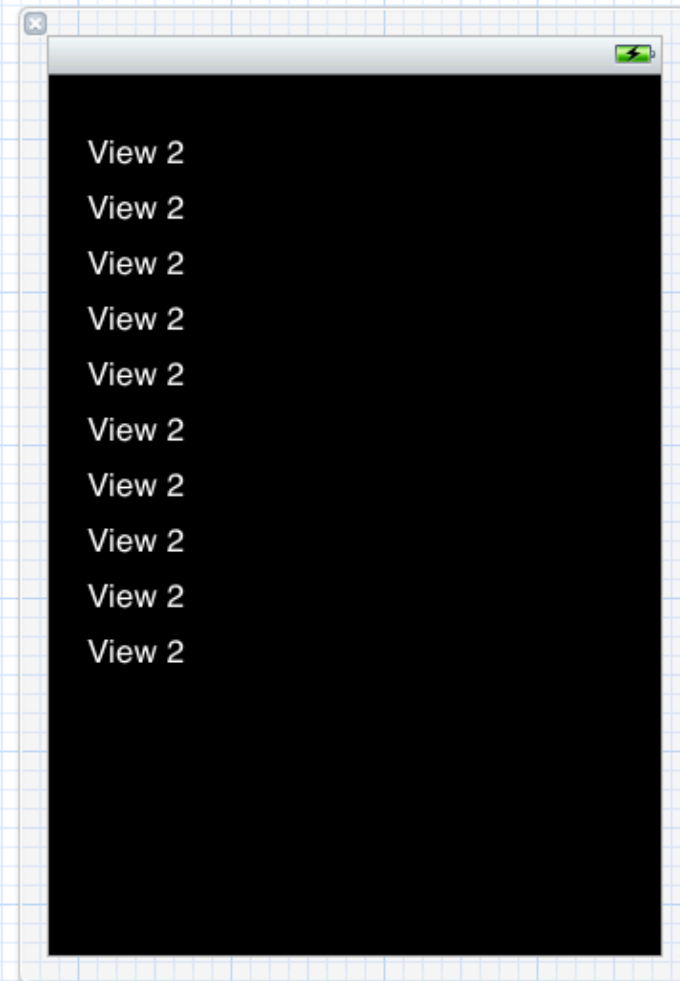
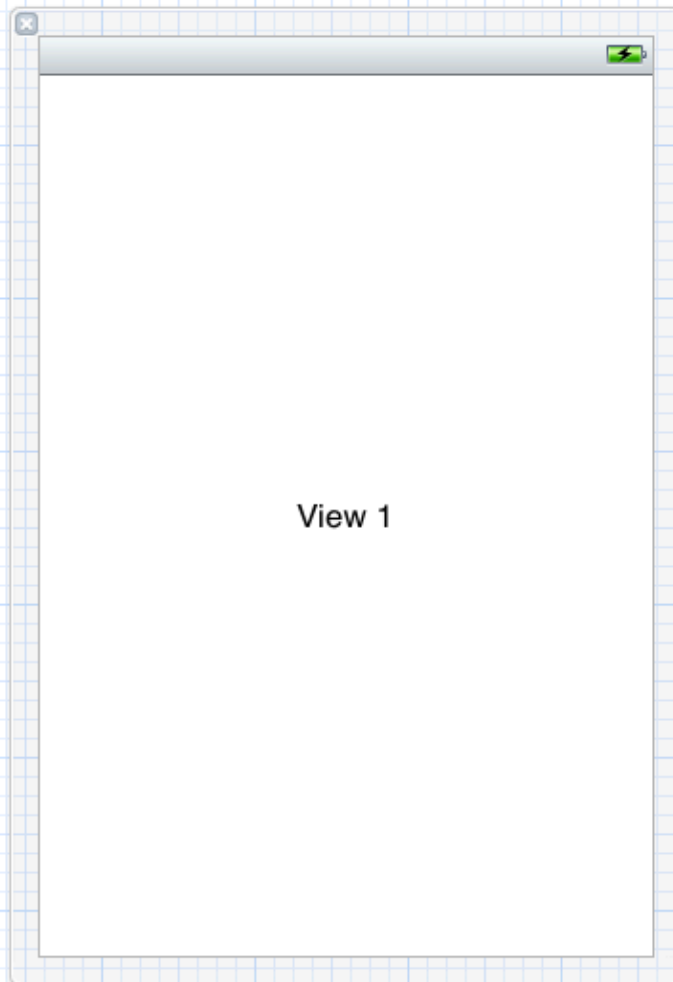

NSStringFromCGRect

- There are some helpers function to represent CGPoint, CGRect and CGSize in NSString
- NSStringFromCGRect
- NSStringFromCGPoint
- NSStringFromCGSize

View Panning



View Panning



View Panning

- ViewDidLoad

```
1 - (void) viewDidLoad
2 {
3     [super viewDidLoad];
4
5     UIPanGestureRecognizer *panGesture = [[UIPanGestureRecognizer alloc]
initWithTarget:self action:@selector(panGestureHandler:)];
6     [self.fgView addGestureRecognizer:panGesture];
7 }
```

View Panning

- Handling panning gesture

```
1 - (void)panGestureHandler:(UIPanGestureRecognizer*) gesture
2 {
3     CGPoint translation = [gesture translationInView:self.view];
4     NSLog(@"%@", NSStringFromCGPoint(translation));
5
6     CGRect frame = self.fgView.frame;
7
8     // gesture ended.
9     if (gesture.state == UIGestureRecognizerStateEnded)
10    {
11        frame.origin.x = 0;
12    } else {
13        frame.origin.x = frame.origin.x + translation.x;
14    }
15
16    // transform the frame.
17    self.fgView.frame = frame;
18
19    [gesture setTranslation:CGPointZero inView:self.view];
20 }
```

Object Memory Management

(required only if Auto Reference Counting disabled)

Object Instance Life Cycle

- alloc memory
- init
- release memory

Object Instance Life Cycle

- alloc memory
- init
- retain
- release
- retain
- release
- release memory

Object Instance Life Cycle

- alloc memory
- init
- retain
- retain
- release
- release
- release memory

Object Instance Life Cycle

- Retain Count is respond to count how many other instances are using this object instance.
- Retain Count increase (+1) when called **init** or **retain**
- Retain Count decrease (-1) when called **release**

Object Instance Life Cycle

Retain Count

- alloc memory 0
- init 1
- release memory 0

Object Instance Life Cycle

Retain Count

- | | |
|------------------|---|
| • alloc memory | 0 |
| • init | 1 |
| • retain | 2 |
| • release | 1 |
| • retain | 2 |
| • release | 1 |
| • release memory | 0 |

Object Instance Life Cycle

Retain Count

- | | |
|------------------|---|
| • alloc memory | 0 |
| • init | 1 |
| • retain | 2 |
| • retain | 3 |
| • release | 2 |
| • release | 1 |
| • release memory | 0 |

Object Instance Life Cycle

Retain Count

- alloc memory 0
- init 1
- retain 2
- retain 3
- release 2
- release memory 1

Leaked

Object Instance Life Cycle

- Getting the retain count

retainCount

```
1 UIView *view = [[UIView alloc] init];  
2 int retainCount = [view retainCount];  
3 NSLog(@"Retain Count: %d", retainCount);
```

Object Instance Life Cycle

- What's More?

autorelease

```
[[[UIView alloc] init] autorelease];
```


Object Instance Life Cycle

- Any More?

Auto Reference Counting (ARC)

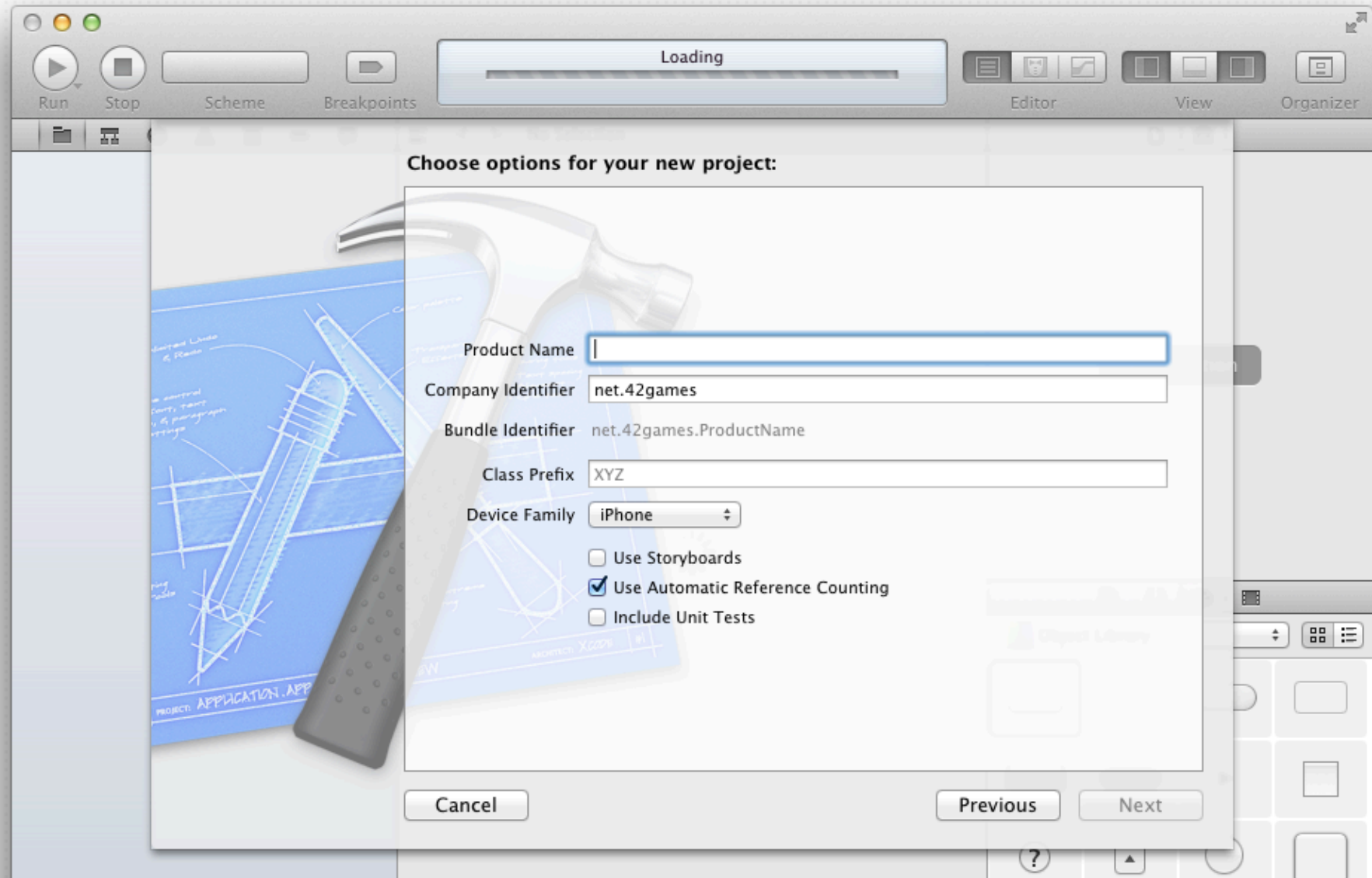
No more **release** and **retain**

Auto Reference Counting

Retain Count

- alloc memory 0
- init 1
- ~~retain~~ 2
- ~~release~~ 1
- ~~retain~~ 2
- ~~release~~ 1
- ~~release~~ memory 0

Auto Reference Counting



Auto Reference Counting

- Do not compatible with non-ARC code.
- You may have problem when using 3rd party code.

(There are workarounds to get non-ARC code work within ARC environment)

Instance Property

Instance Property

```
@property (nonatomic, retain) NSString *something;
```

```
@synthesize something; (Required in old XCode 4)
```

Instance Property

`@property (nonatomic, retain) NSString *something;`

`@synthesize something; (Optional now)`

Instance Property

```
@property (nonatomic, retain) NSString *something;
```

```
@synthesize something=localName;
```


Instance Property

@synthesize something;

equals to

```
1 - (NSString *)something
2 {
3     return something;
4 }
5 - (void)setSomething:(NSString*)newSomething
6 {
7     [something release];
8     something = [newSomething retain];
9 }
```

Exercise

- ✓ Think about how to make use of gestures.
- ✓ Design an app with gesture features.
- ✓ Or discover an app on app store with gesture features.
- ✓ Present it to the class in next lesson.