

# iPhone App Dev

Lesson 3

## Source Codes

<https://github.com/makzan/ios-dev-course-example>

## Practice

- Browse the App Store. Find some apps that impress you. Discuss why they are good.
- Do you have any problem that want to solve in a mobile phone? What app do you want to develop?

## Practice

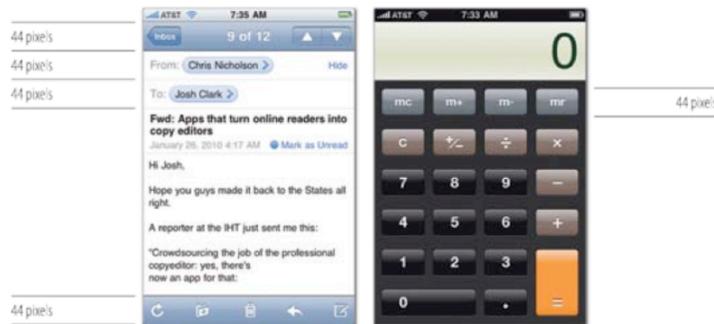
- ✓ Design an utility app.
- ✓ Present it to the class in lesson.

# Summary

- The 44 Pixels
- (Extra) Currency Convertor in RubyMotion
- Detecting Touches
- Multi Touches
- Gesture Recognition
- Object Memory Management
- Instance Property
- Comfortable Thumb Area

# Button Tap Area

## The 44 pixels



Graph from TapWorthy book.

## The 44 pixels



Graph from TapWorthy book.

# Less than 44 pixels

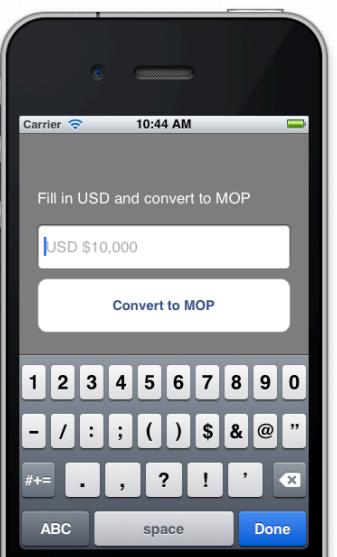


Graph from TapWorthy book.

# RubyMotion

## RubyMotion

```
10:53:50 molcan: MOP : /Users/molcan/.rvm/gems/ruby-1.9.3-p125 : v--v : master$ rake
    Build .build/iPhoneSimulator-5.0-Development
  Simulate .build/iPhoneSimulator-5.0-Development/MOP.app
(main)>> 2012-05-11 10:53:53.788 MOP[54667:8f83] Ruby Rocks
(main)>>
```



## RubyMotion

- Compiling Ruby code directly into iOS byte code.
- Share all the iOS API
- Mixing Objective-C library
- Interactive console
- Apple Store compatible

# RubyMotion

- Creating app project

```
1 $ motion create MOP
2   Create MOP
3   Create MOP/.gitignore
4   Create MOP/Rakefile
5   Create MOP/app
6   Create MOP/app/app_delegate.rb
7   Create MOP/resources
8   Create MOP/spec
9   Create MOP/spec/main_spec.rb
```

# RubyMotion

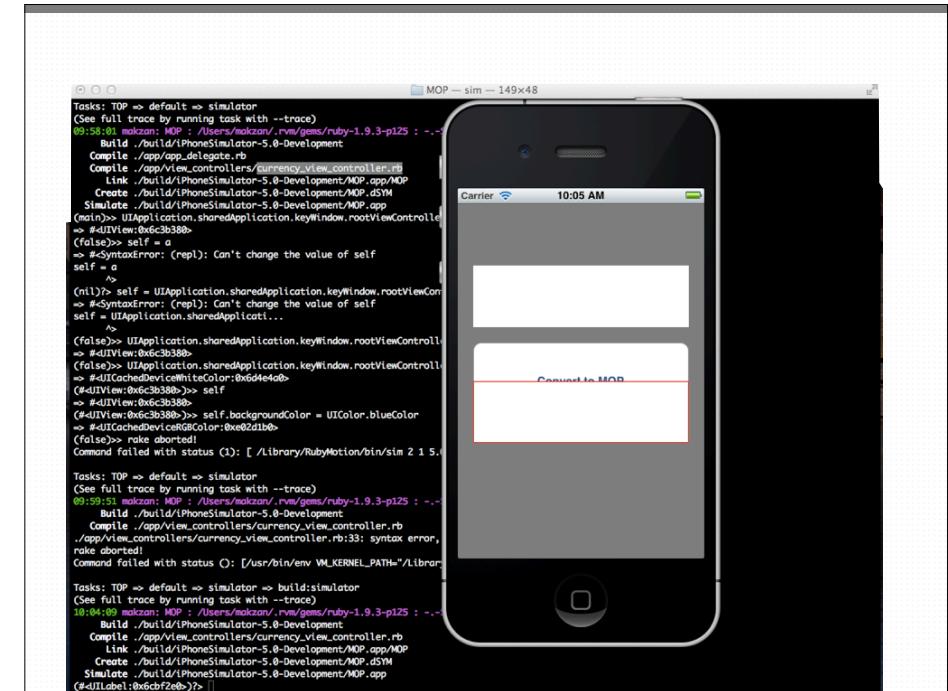
- Interaction Console

```
1 $ rake
2   Build ./build/iPhoneSimulator-5.0-Development
3   Compile ./app/app_delegate.rb
4   Compile ./app/view_controllers/currency_view_controller.rb
5     Link ./build/iPhoneSimulator-5.0-Development/MOP.app/MOP
6   Create ./build/iPhoneSimulator-5.0-Development/MOP.dSYM
7   Simulate ./build/iPhoneSimulator-5.0-Development/MOP.app
8
9 (#<UIView:0x6c3b380>) >> self
10 => #<UIView:0x6c3b380>
11 (#<UIView:0x6c3b380>) >> self.backgroundColor = UIColor.blueColor
12 => #<UICachedDeviceRGBColor:0xe02d1b0>
13 (nil) ?>
```

# RubyMotion

- Code in Ruby, with iOS API

```
1 class CurrencyViewController < UIViewController
2   def loadView
3     self.view = UIView.alloc.init
4   end
5
6   def viewWillAppears(animated)
7     super
8   end
9 end
```



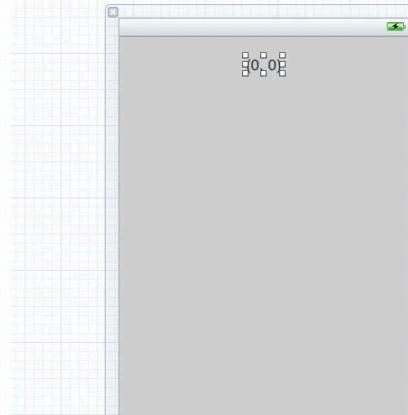
# RubyMotion Demo

# 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     coordLabel.text = [NSString stringWithFormat:@"(% .2f, %.2f)",
10     point.x, point.y];
11 }
```

# 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     coordLabel.text = [NSString stringWithFormat:@"(% .2f, %.2f)",
10     point.x, point.y];
11 }
```

# Detect Touches

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

# 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
```

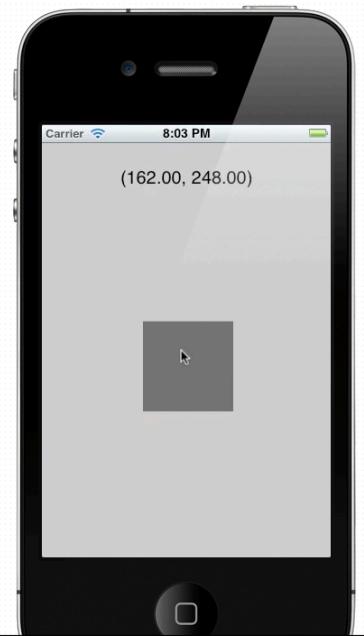
# CGGeometry

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

# CGGeometry

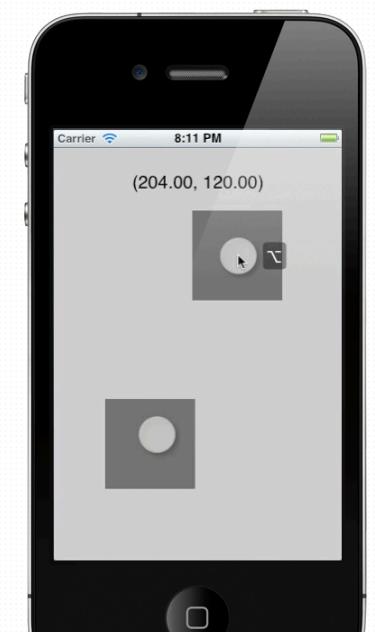
- Primitives and Functions for geometric.
- CGPointMake
- CGRectMake
- CGSizeMake

## Drag and Drop Demo



## Multi Touches

## Multi Touches Drag-n-Drop



# Multi Touches Dragging

- 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 }
```

# 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         coordLabel.text = [NSString stringWithFormat:@"(% .2f, %.2f)",
point.x, point.y];
7         if (CGRectContainsPoint(box.frame, point))
8         {
9             box.center = point;
10            touchForBox1 = touch;
11        } else if (CGRectContainsPoint(box2.frame, point))
12        {
13            box2.center = point;
14            touchForBox2 = touch;
15        }
16    }
17 }
```

# Multi Touches Dragging

- Two variables to remember the touches pointer

```
1 @implementation ViewController {
2     UITouch *touchForBox1;
3     UITouch *touchForBox2;
4 }
```

# 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         coordLabel.text = [NSString stringWithFormat:@"(% .2f, %.2f)",
point.x, point.y];
7         if (touchForBox1 == touch)
8         {
9             box.center = point;
10        } else if (touchForBox2 == touch)
11        {
12            box2.center = point;
13        }
14    }
15 }
```

# Multi Touches Dragging

- Reset the pointer after touch ended

```
1 - (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event
2 {
3     coordLabel.text = @"Not Touching";
4
5     for(UITouch *touch in touches)
6     {
7         CGPoint point = [touch locationInView:self.view];
8
9         coordLabel.text = [NSString stringWithFormat:@"(% .2f, % .2f)",
point.x, point.y];
10
11        if (touchForBox1 == touch)
12        {
13            touchForBox1 = nil;
14        }
15
16        if (touchForBox2 == touch)
17        {
18            touchForBox2 = nil;
19        }
20
21    }
22}
```

# 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 }
```

# Gesture Recognition

## @selector

@selector(helloWorld)

- (void)helloWorld

@selector(helloWorld:)

- (void)helloWorld:(id)param

# @selector

```
@selector(sendMessage:to:  
- (void)sendMessage:(id)msg to:(id)somebody
```

# Gesture Recognizer

- Swipe

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

# Gesture Recognizer

- Long Press

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

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

## View Panning

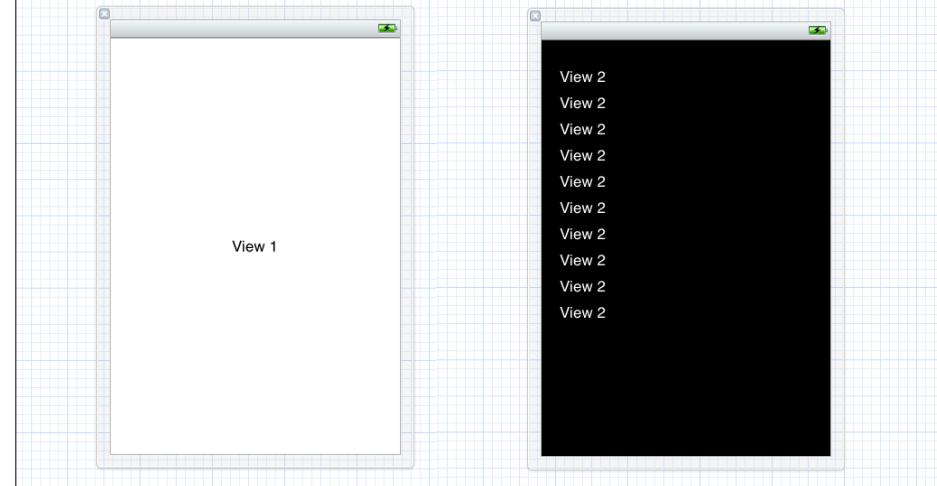


# NSStringFromCGRect

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

- NSStringFromCGRect
- NSStringFromCGPoint
- NSStringFromCGSize

## View Panning



# View Panning

- ViewDidLoad

```
1 - (void)viewDidLoad
2 {
3     [super viewDidLoad];
4
5     UIPanGestureRecognizer *panGesture = [[UIPanGestureRecognizer alloc]
initWithTarget:self action:@selector(panGestureHandler:)];
6     [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 = fgView.frame;
7
8     // gesture ended.
9     if (gesture.state == UIGestureRecognizerStateChanged)
10    {
11        frame.origin.x = 0;
12    }else {
13        frame.origin.x = frame.origin.x + translation.x;
14    }
15
16    // transform the frame.
17    fgView.frame = frame;
18
19    [gesture setTranslation:CGPointZero inView:self.view];
20 }
```

# 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 = fgView.frame;
7
8     // gesture ended.
9     if (gesture.state == UIGestureRecognizerStateChanged)
10    {
11        frame.origin.x = 0;
12    }else {
13        frame.origin.x = frame.origin.x + translation.x;
14    }
15
16    // transform the frame.
17    fgView.frame = frame;
18
19    [gesture setTranslation:CGPointZero inView:self.view];
20 }
```

# Object Memory Management

# 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
- retain 3
- release 2
- release 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 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

- Any More?

## **Auto Reference Counting (ARC)**

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

# Object Instance Life Cycle

- What's More?

## **autorelease**

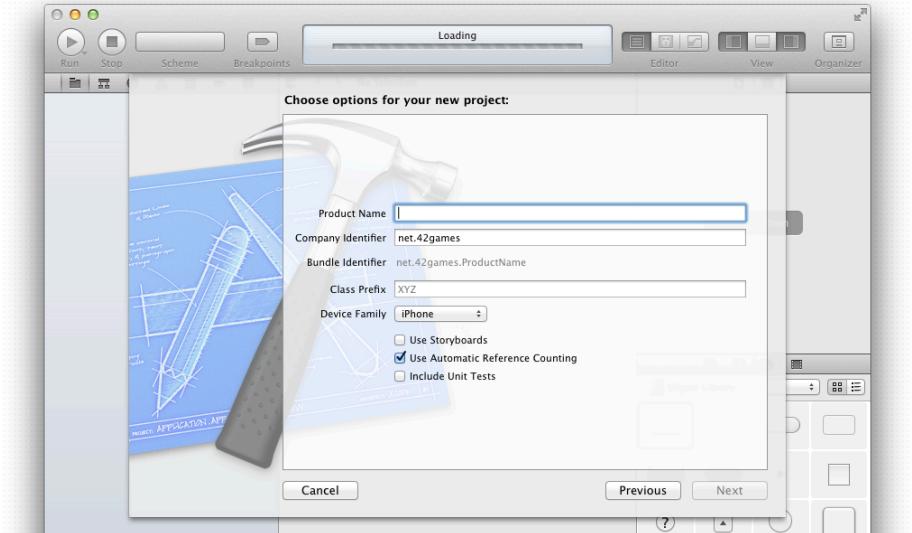
```
[[[UIView alloc] init] autorelease];
```

# Auto Reference Counting

## **Retain Count**

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

# Auto Reference Counting



# Instance Property

# 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

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

# Instance Property

@synthesize something;

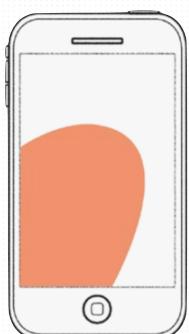
equals to

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

# Comfortable Tapping Area

## Comfortable Thumb Area

- Put usual buttons at bottom
- Put critical button at top



Graph from TapWorthy book.

## Comfortable Thumb Area

- Put usual buttons at bottom
- Put critical button at top



Graph from TapWorthy book.

# Comfortable Thumb Area

- Put usual buttons at bottom
- Put critical button at top



Graph from TapWorthy book.

# 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.