

Workshop 9: Taps and gestures

In this workshop you will learn about the iOS multitouch features.

Task 1: Multiple fingers

Touch screens are not new, but the great Apple innovation was 'multitouch'. This means recognising more than one finger on the screen. For example, the iPad can handle up to 11 fingers on screen at once!

Create a new View-based project called Touches. In the view controller, add the following code:

```
- (void)touchesBegan:(NSSet *)touches withEvent:(UIEvent *)event {
    NSUInteger numTouches = [touches count];
    NSUInteger numTaps = [[touches anyObject] tapCount];
    NSLog(@"touchesBegan... Touches: %d Taps: %d", numTouches, numTaps);
}

- (void)touchesMoved:(NSSet *)touches withEvent:(UIEvent *)event {
    NSLog(@"touchesMoved...");
}

- (void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event {
    NSLog(@"touchesEnded...");
}
```

Run the app and try clicking in the simulator. What is the meaning of these 3 methods? Why is the number of touches always 1?

Task 2: Taps

Definitions:

Touch = a finger on the screen

Tap = a very short touch (touch down then touch up)

Modify your app so that: if you tap on the screen 3 times, then a photo of a monster is shown.

Hint: You will need to add a UIImageView to your view controller.

Task 3: Taps, the easy way

You can program touches and taps using the 3 methods above, but there is an easier way!

Delete the touchesBegan, touchesMoved and touchesEnded methods.

In your viewDidLoad method, set up a UIGestureRecognizer:

```
- (void)viewDidLoad {
    [super viewDidLoad];

    UITapGestureRecognizer *tapRecognizer = [[UITapGestureRecognizer alloc]
initWithTarget:self action:@selector(tap:)];
    tapRecognizer.numberOfTapsRequired = 3;
    [self.view addGestureRecognizer:tapRecognizer];
    [tapRecognizer release];
}
```

There are 3 steps to setting up the UITapGestureRecognizer:

1. Create the UITapGestureRecognizer object and set an action (e.g. @selector(tap:)).
2. Set the numberOfTapsRequired.
3. Add the recognizer object to the view.

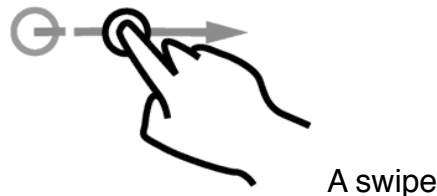
You also need to add a method to be performed when the taps are detected:

```
- (void)tap:(UIGestureRecognizer *)recognizer {
    NSLog(@"3 taps!");
    // Show the monster
}
```

Run the app.

Task 4: Swipe

Gesture recognizers make it easy to detect different types of touches. A swipe is a quick drag in one direction:



Try the following code:

```
- (void)viewDidLoad {
    [super viewDidLoad];

    UISwipeGestureRecognizer *swipeRightRecognizer =
[[UISwipeGestureRecognizer alloc] initWithTarget:self action:@selector
(swipeRight:)];
    swipeRightRecognizer.direction =
UISwipeGestureRecognizerDirectionRight;
    [self.view addGestureRecognizer:swipeRightRecognizer];
    [swipeRightRecognizer release];

    UISwipeGestureRecognizer *swipeLeftRecognizer =
[[UISwipeGestureRecognizer alloc] initWithTarget:self action:@selector
(swipeLeft:)];
    swipeLeftRecognizer.direction = UISwipeGestureRecognizerDirectionLeft;
    [self.view addGestureRecognizer:swipeLeftRecognizer];
    [swipeLeftRecognizer release];
}
```

```

- (void)swipeRight:(UIGestureRecognizer *)recognizer {
    NSLog(@"Right");
}

- (void)swipeLeft:(UIGestureRecognizer *)recognizer {
    NSLog(@"Left");
}

```

Task 5: Swipe to open a picker

Let's use the swipe gesture recognizer to do something useful...

Start by [downloading the code](#).

Run the app. You will see two buttons that show and hide a 'picker'. A UIPickerView is a component for selecting options (it has a data source and delegate like a UITableView).

Your task is to remove the two buttons and modify the app so that a 'swipe up' opens the picker and a 'swipe down' hides the picker.

Task 6: More gestures

Download the SimpleGestureRecognizers sample code from Apple:

<http://developer.apple.com/library/IOS/#samplecode/SimpleGestureRecognizers/Introduction/Intro.html>

Look at the code and run the app. What gestures are used?

Try out some more gesture recognizers:

Gesture	Class
Tapping (any number of taps)	UITapGestureRecognizer
Pinching in and out (for zooming a view)	UIPinchGestureRecognizer
Panning or dragging	UIPanGestureRecognizer
Swiping (in any direction)	UISwipeGestureRecognizer
Rotating (fingers moving in opposite directions)	UIRotationGestureRecognizer
Long press (also known as "touch and hold")	UILongPressGestureRecognizer