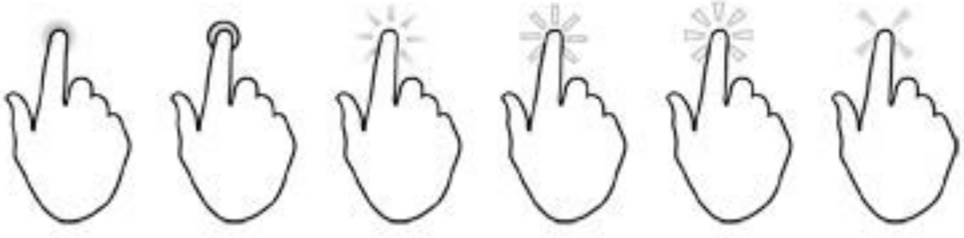
ITP 342
Mobile App Dev

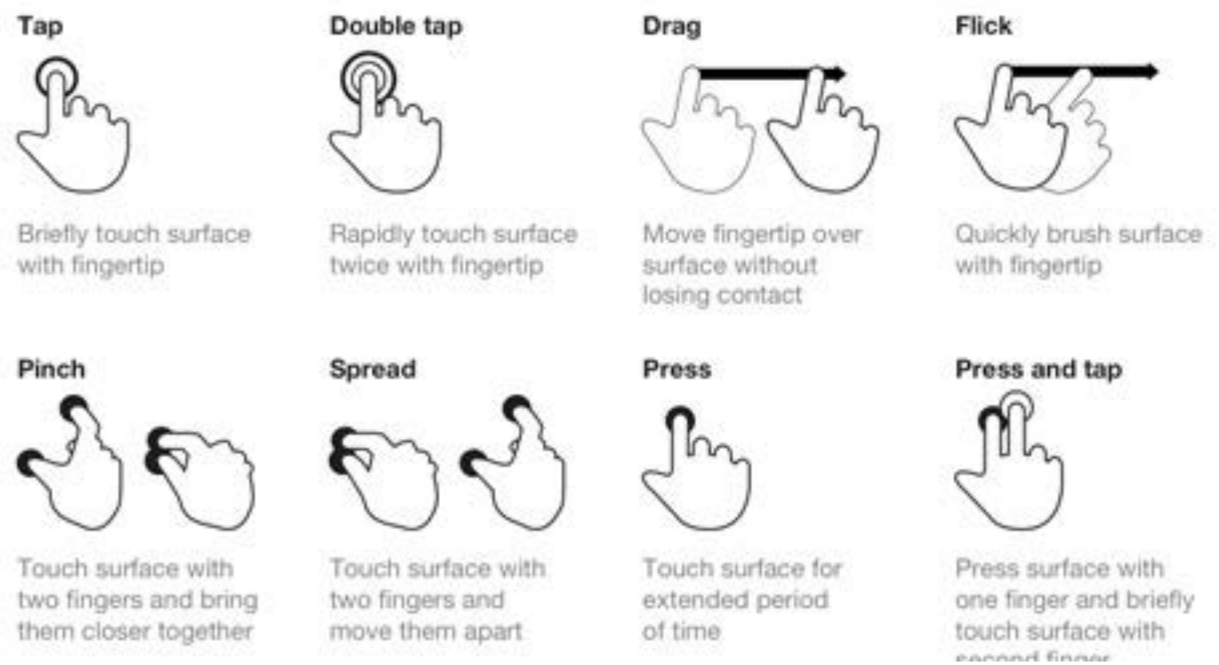


Gestures

USC
School of Engineering

University of Southern California

Gestures



Tap

Briefly touch surface with fingertip

Double tap

Rapidly touch surface twice with fingertip

Drag

Move fingertip over surface without losing contact

Flick

Quickly brush surface with fingertip

Pinch

Touch surface with two fingers and bring them closer together

Spread

Touch surface with two fingers and move them apart

Press

Touch surface for extended period of time

Press and tap

Press surface with one finger and briefly touch surface with second finger

Touch

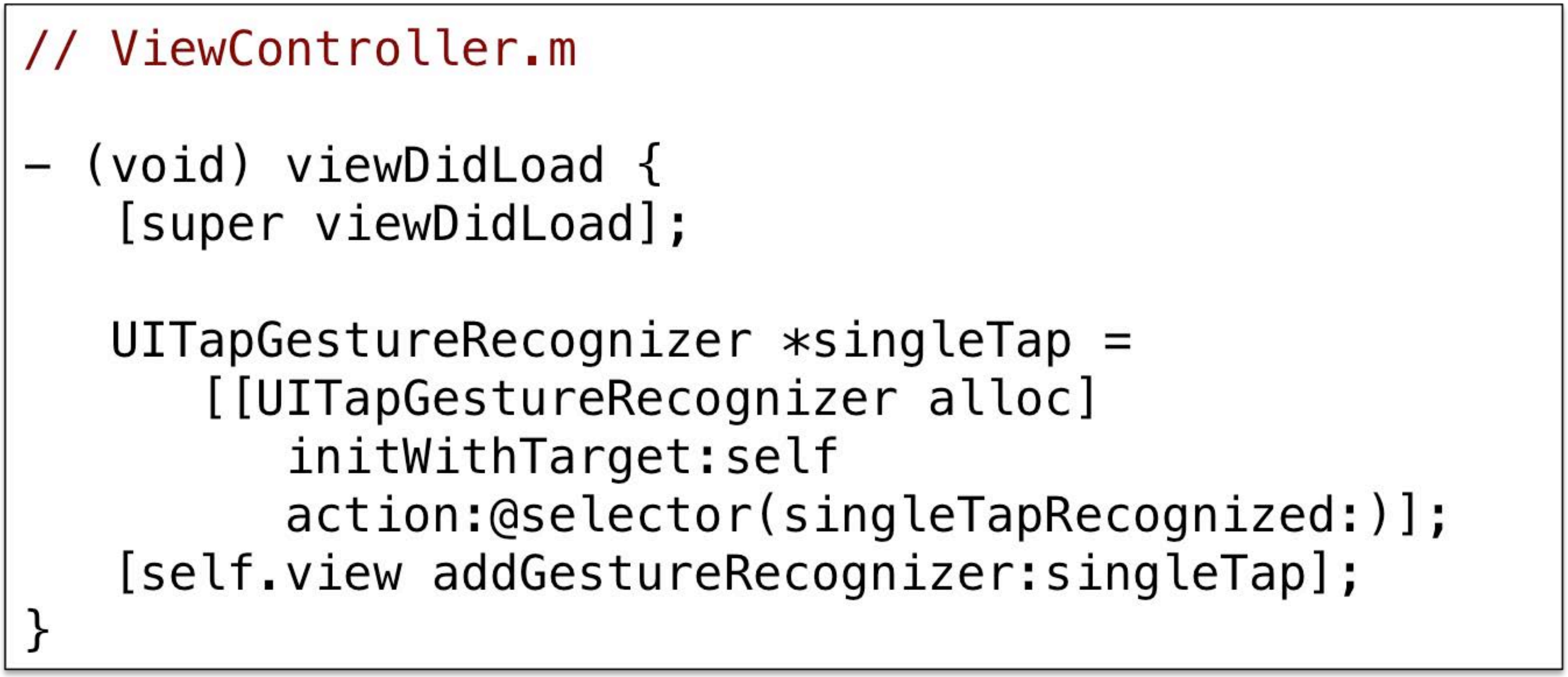
- Improve the UI by adding gesture recognizers
- https://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/GestureRecognizer_basics/GestureRecognizer_basics.html

Tap Gestures

- A gesture recognizer can be added to any UIView
- When a gesture is recognized, the recognizer sends an action message to the target object
- This decouples the logic for recognizing a gesture from the logic that handles the gesture

Tap Gestures

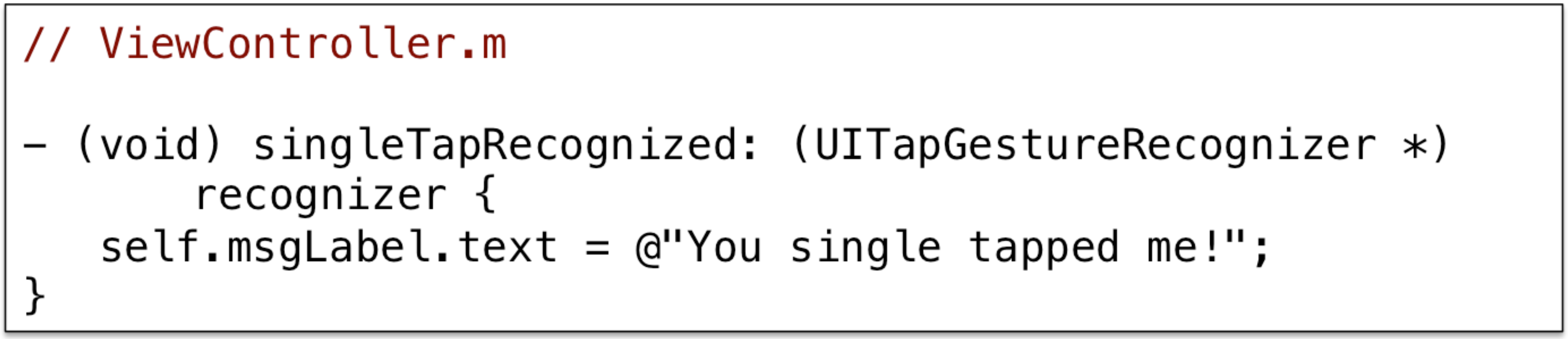
- Add gesture recognizer in `viewDidLoad` method



```
// ViewController.m

- (void) viewDidLoad {
   [super viewDidLoad];

   UITapGestureRecognizer *singleTap =
      [[UITapGestureRecognizer alloc]
         initWithTarget:self
         action:@selector(singleTapRecognized:)];
   [self.view addGestureRecognizer:singleTap];
}
```

Selectors

- A selector is just a method identifier
- It is not an explicit method call
- Keep your promises!
  - Make sure the methods you refer to actually exist



```
// ViewController.m

- (void) singleTapRecognized: (UITapGestureRecognizer *)
        recognizer {
    self.msgLabel.text = @"You single tapped me!";
}
```

Undeclared Selectors

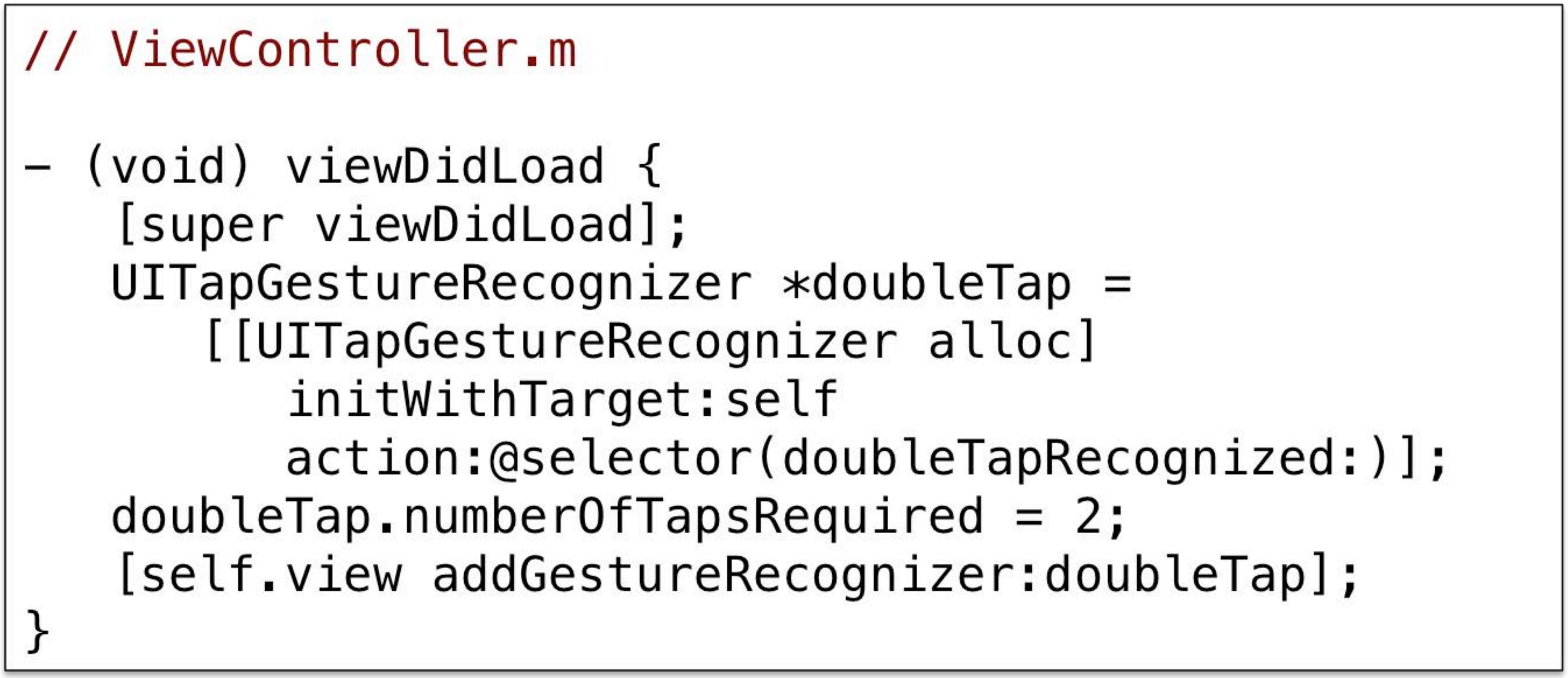
- Selectors are a promise
  - Keep that promise or

```
Terminating app due to uncaught exception
```

- You must implement selectors that you pass
- Xcode will not warn you of missing selectors unless
  - You tell it to in your target's Build Settings
    - Set Undeclared Selector to Yes

Double Taps

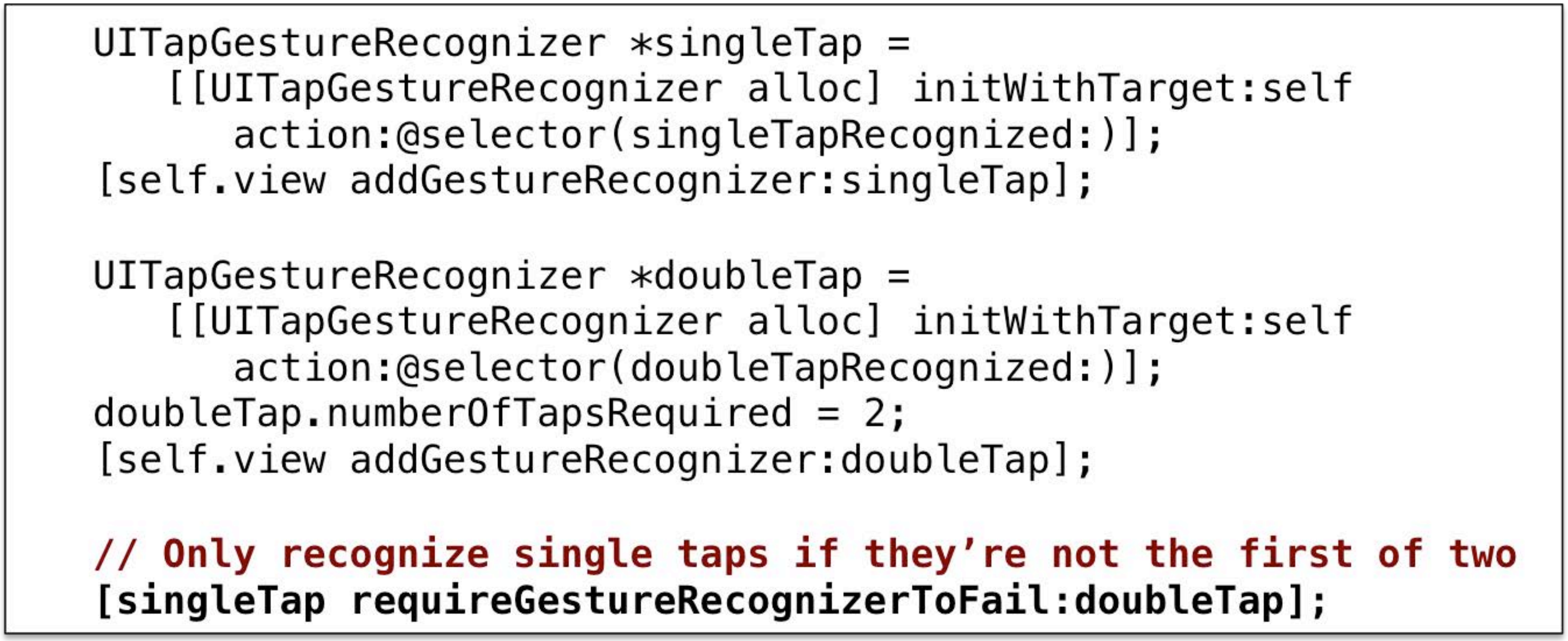
- UITapGestureRecognizer responds to each tap
- You can configure it to require a double tap instead



```
// ViewController.m

- (void) viewDidLoad {
   [super viewDidLoad];
   UITapGestureRecognizer *doubleTap =
      [[UITapGestureRecognizer alloc]
         initWithTarget:self
         action:@selector(doubleTapRecognized:)];
   doubleTap.numberOfTapsRequired = 2;
   [self.view addGestureRecognizer:doubleTap];
}
```

Single & Double Taps

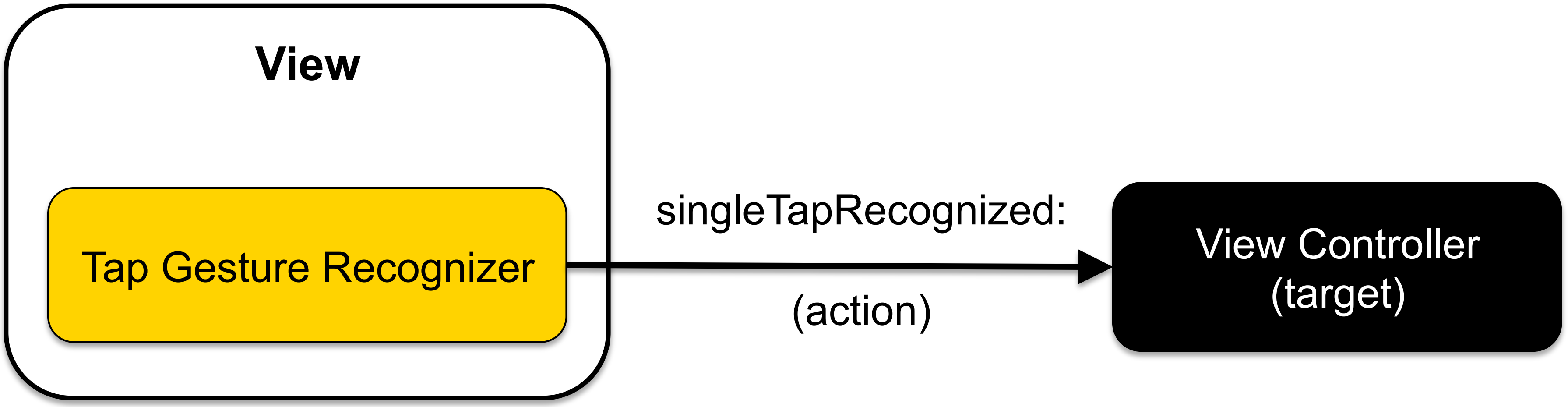
- Only recognize single taps if they are not the first of two taps



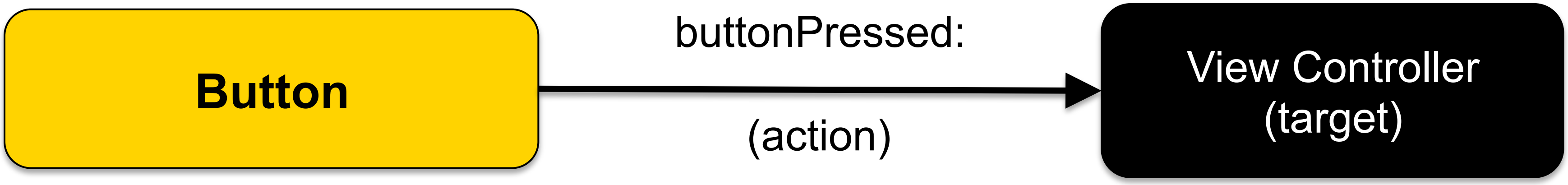
```
UITapGestureRecognizer *singleTap =
    [[UITapGestureRecognizer alloc] initWithTarget:self
       action:@selector(singleTapRecognized:)];
[self.view addGestureRecognizer:singleTap];

UITapGestureRecognizer *doubleTap =
    [[UITapGestureRecognizer alloc] initWithTarget:self
       action:@selector(doubleTapRecognized:)];
doubleTap.numberOfTapsRequired = 2;
[self.view addGestureRecognizer:doubleTap];

// Only recognize single taps if they’re not the first of two
[singleTap requireGestureRecognizerToFail:doubleTap];
```

The Target-Action Design Pattern

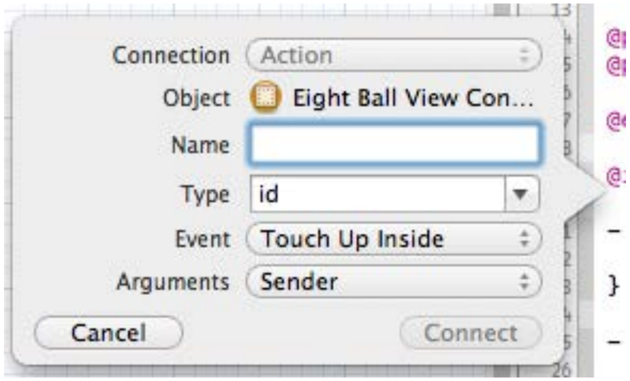
- Gesture recognizers are added to a view
- They take a **target** object and **action** selector
- The action is sent to the target when the specified gesture is recognized on the view



Target-Action in Buttons



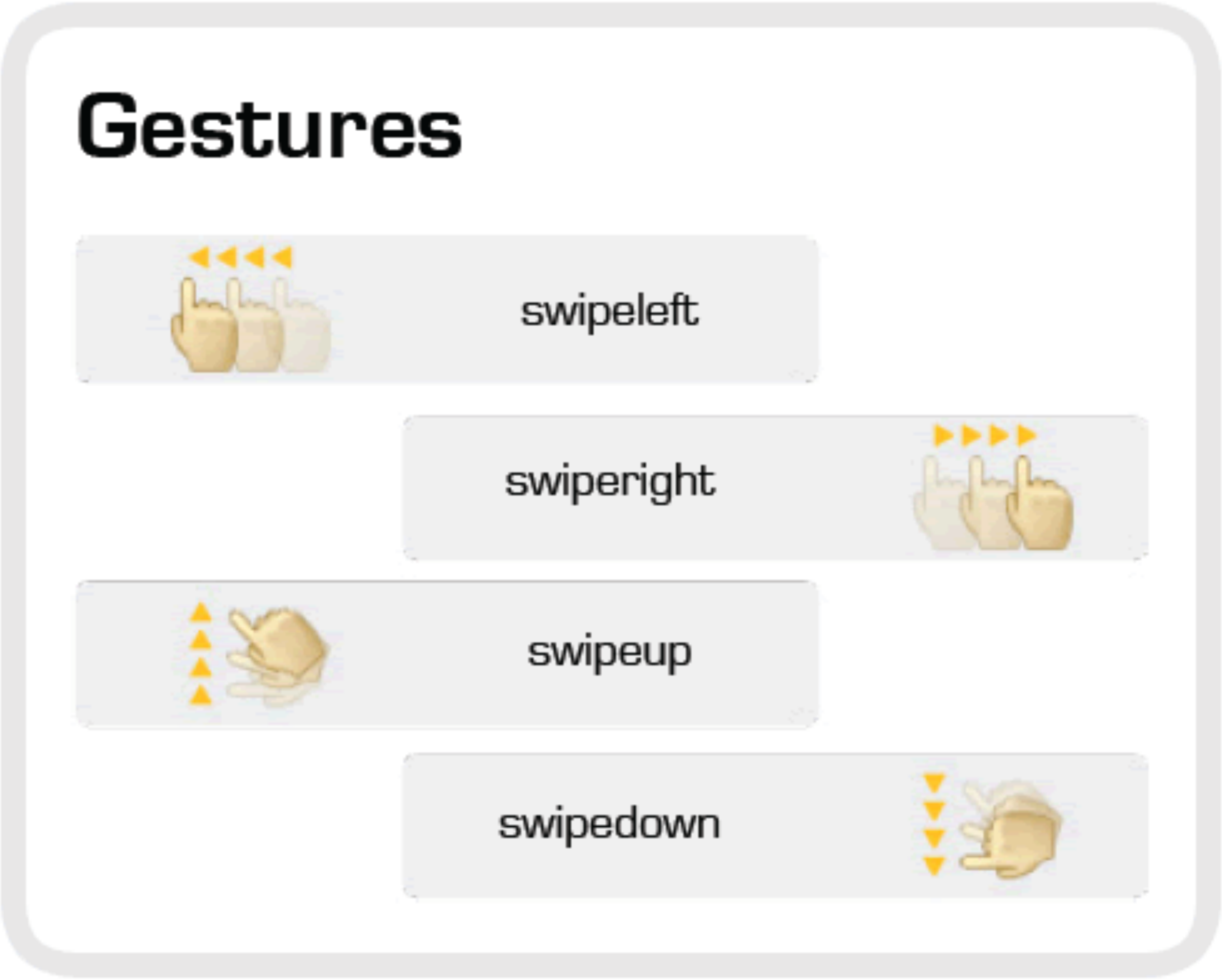
- From Interface Builder
  - IBAction


Connection: Action
Object: Eight Ball View Con...
Name:
Type: id
Event: Touch Up Inside
Arguments: Sender

- From Code

```
[answerButton addTarget: self
   action: @selector( buttonPressed: )
   forControlEvents: UIControlEventTouchUpInside ];
```

Swipe Gestures



Gestures

swipeleft

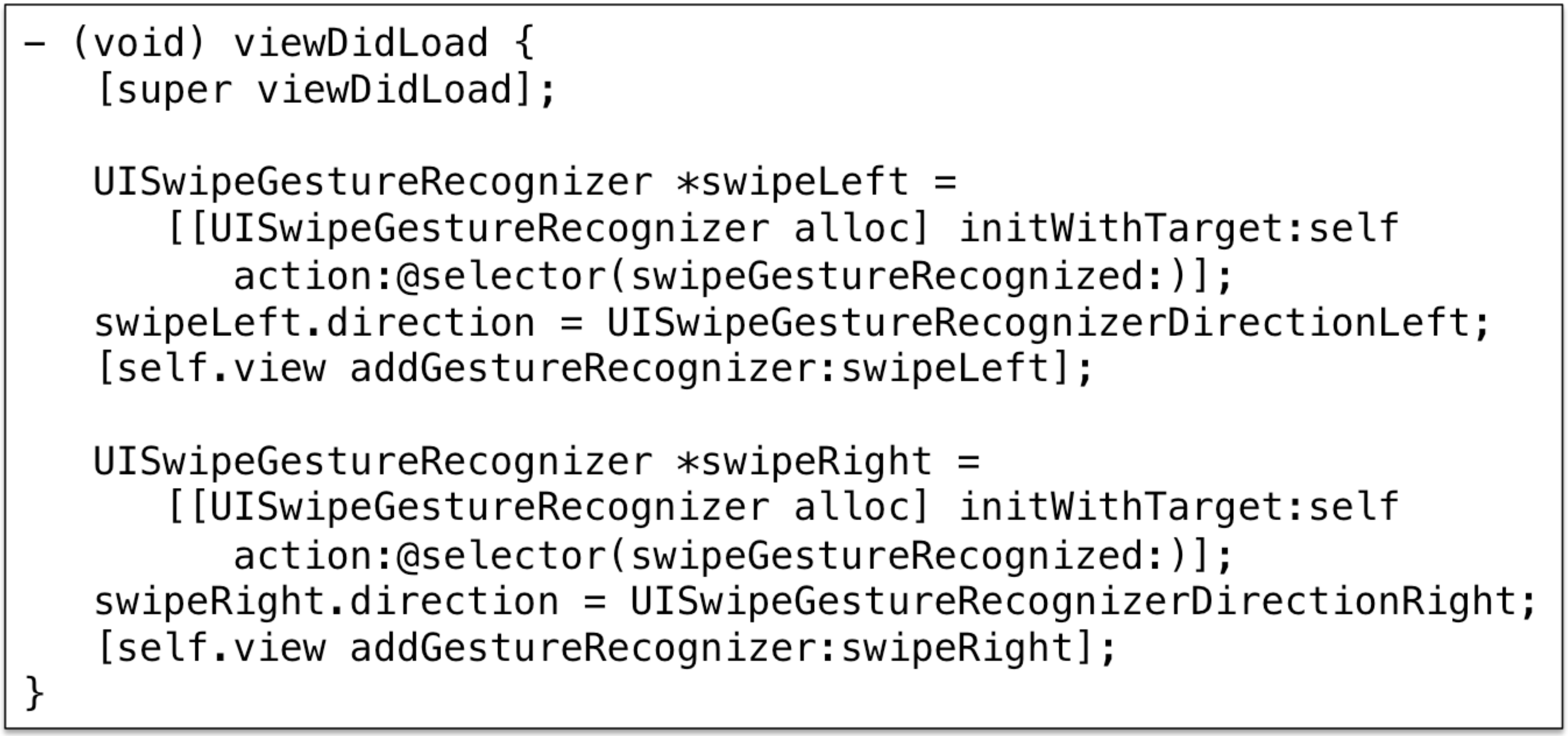
swiperight

swipeup

swipedown

Swipe Gestures

- Need a separate gesture recognizer for every swipe direction we want to support



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

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

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