Stanford CS193p

Developing Applications for iOS
Winter 2013

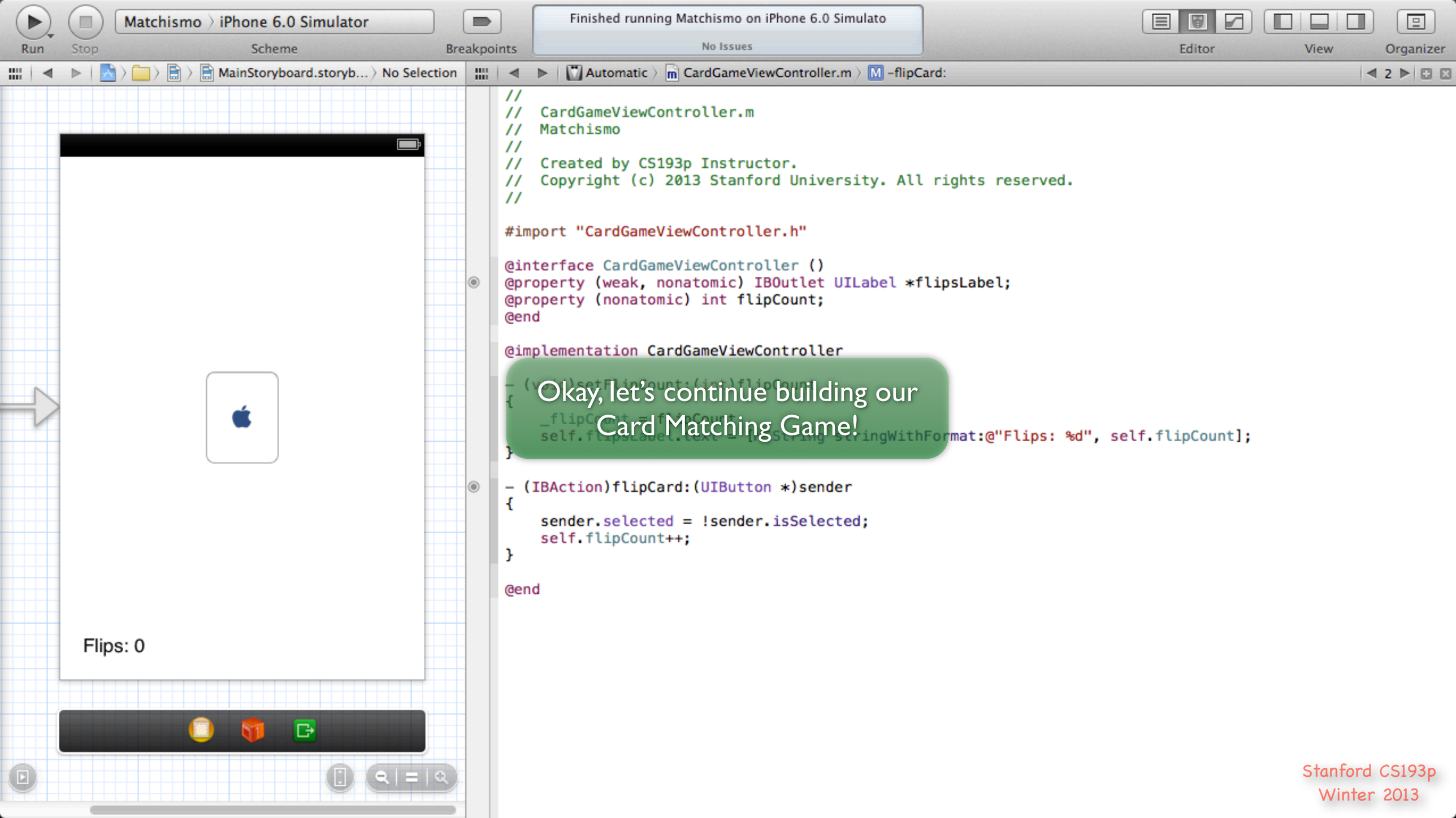
Today

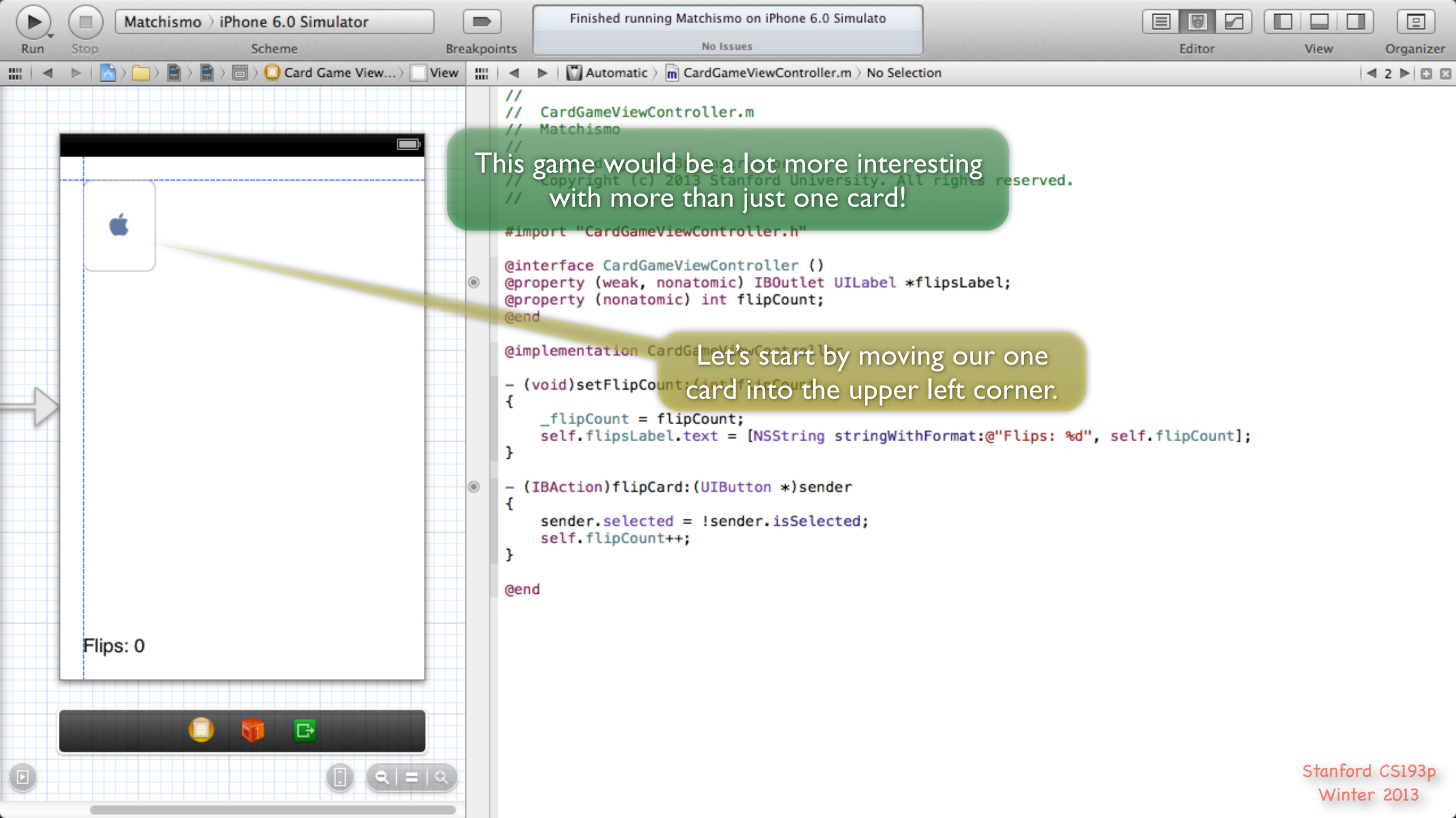
- Card Matching Game

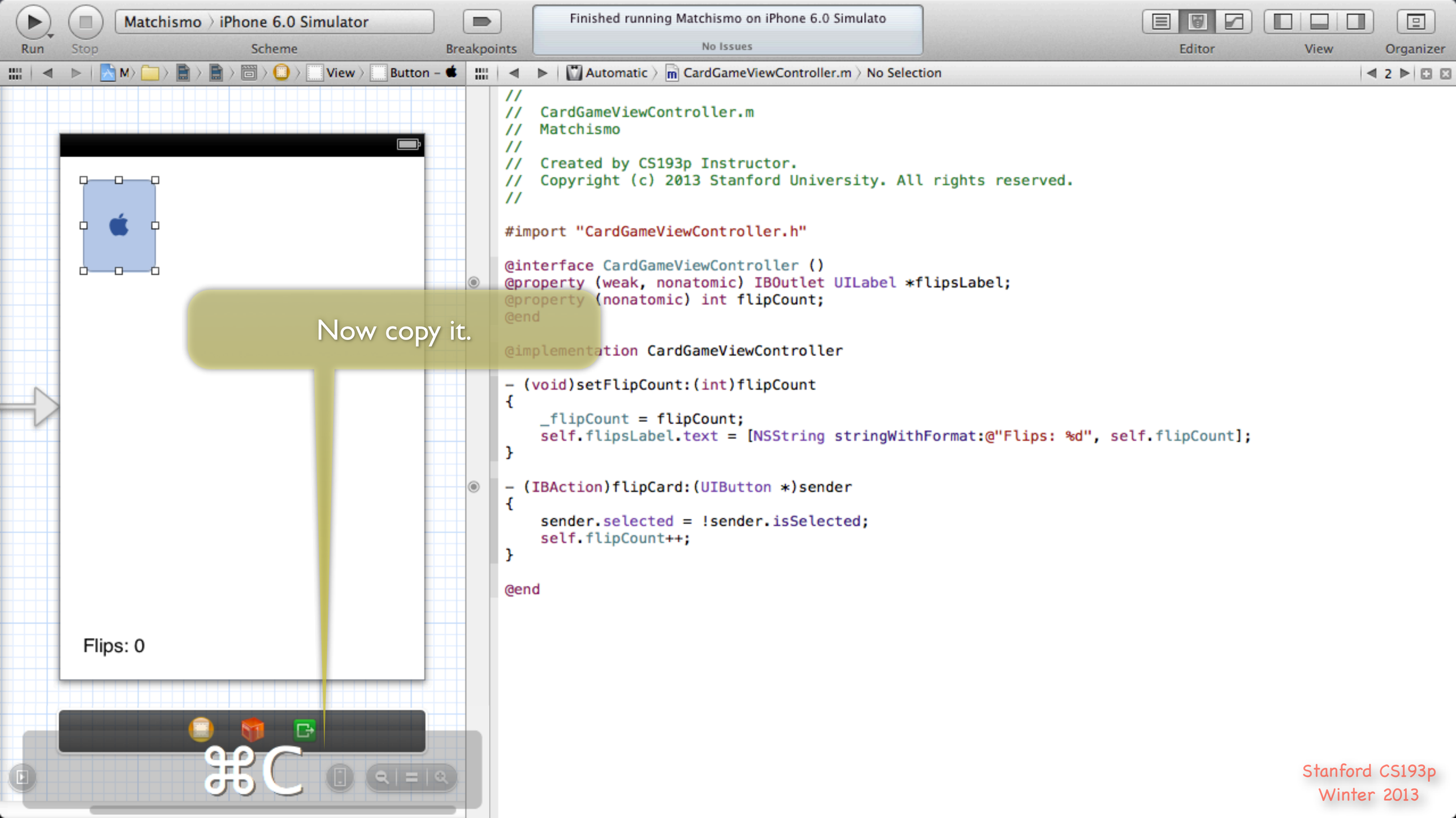
  Today we will create a much more sophisticated Model (than a mere deck of playing cards).
  And we'll enhance our Controller and View to take advantage.

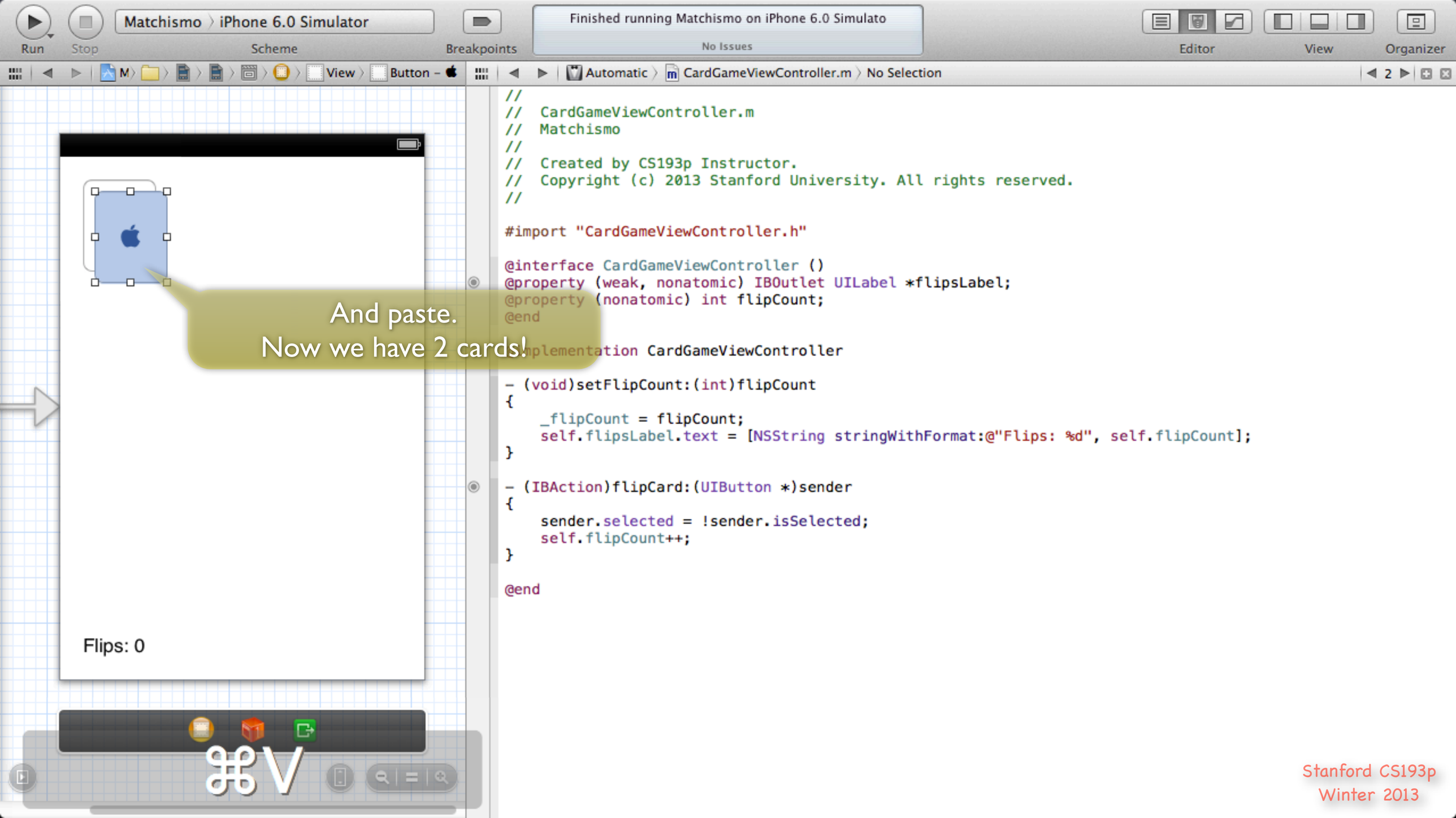
- Quick Review

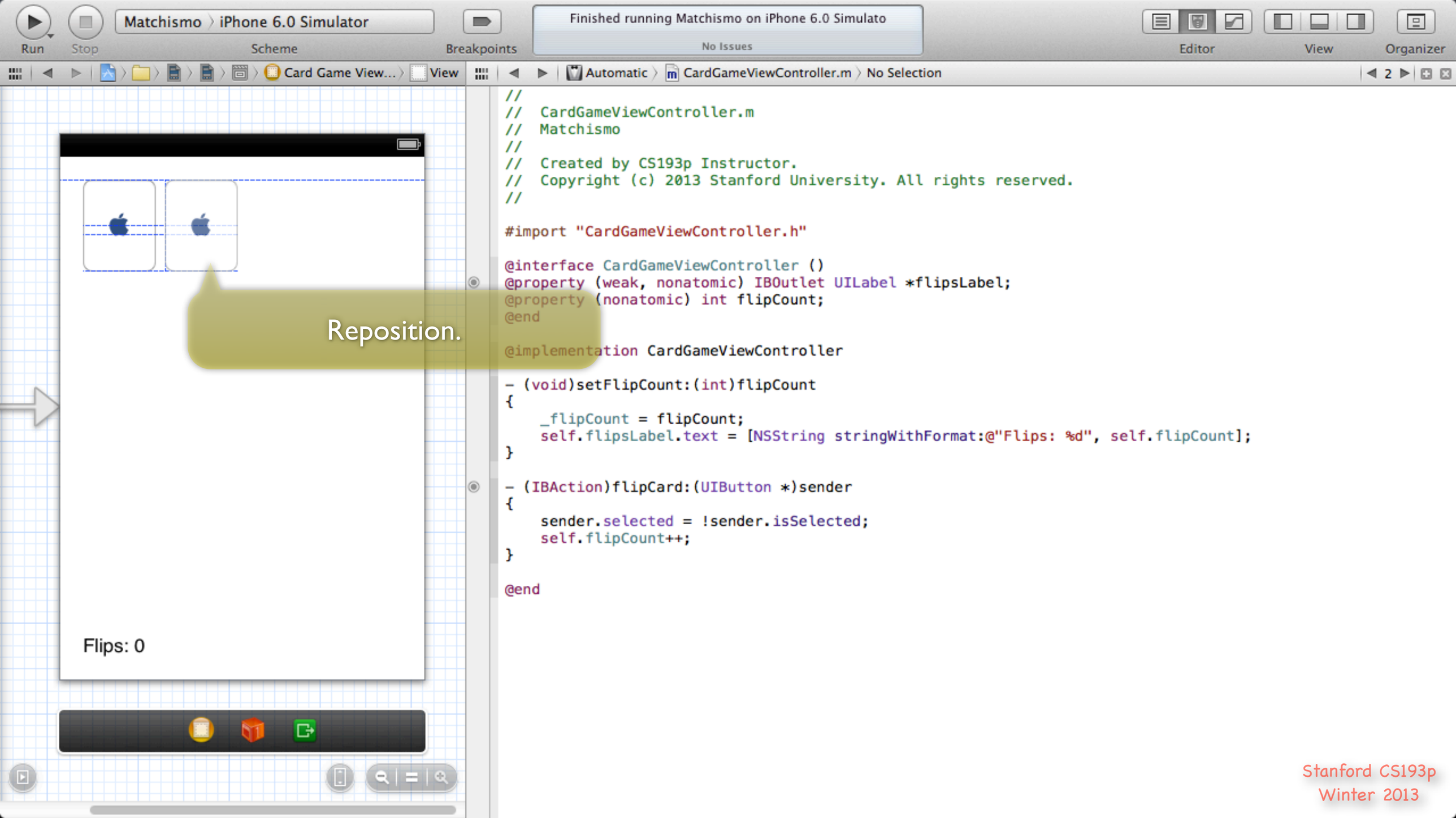
  Time permitting, we'll review what we've learned so far conceptually.


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
MainStoryboard.storyb... 〉 No Selection
Automatic 〉 CardGameViewController.m 〉 -flipCard:
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Okay, let's continue building our Card Matching Game!
Stanford CS193p
Winter 2013


Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
No Selection
2
//
// CardGameViewController.m
// Matchismo
//
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
This game would be a lot more interesting with more than just one card!
Let's start by moving our one card into the upper left corner.
Flips: 0
Stanford CS193p
Winter 2013

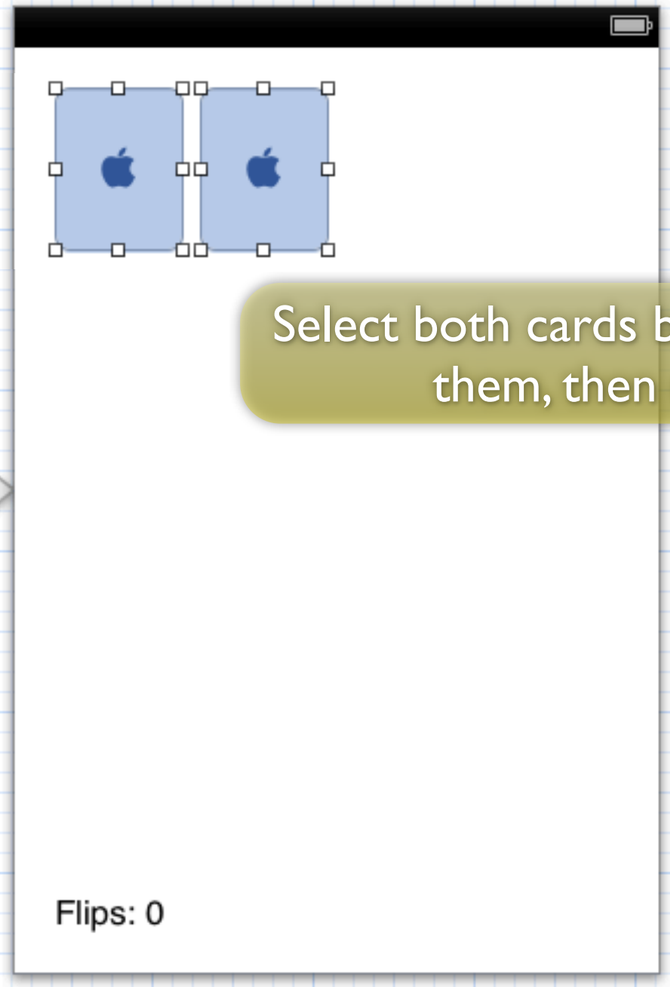

Matchismo 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic 〉 CardGameViewController.m 〉 No Selection
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Now copy it.
Flips: 0
⌘C
Stanford CS193p
Winter 2013


Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic
CardGameViewController.m
No Selection
Button
Flips: 0
And paste.
Now we have 2 cards!
⌘V
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 No Selection
2
Reposition.
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

Run Stop Matchismo › iPhone 6.0 Simulator Scheme Breakpoints Finished running Matchismo on iPhone 6.0 Simulato No Issues Editor View Organizer

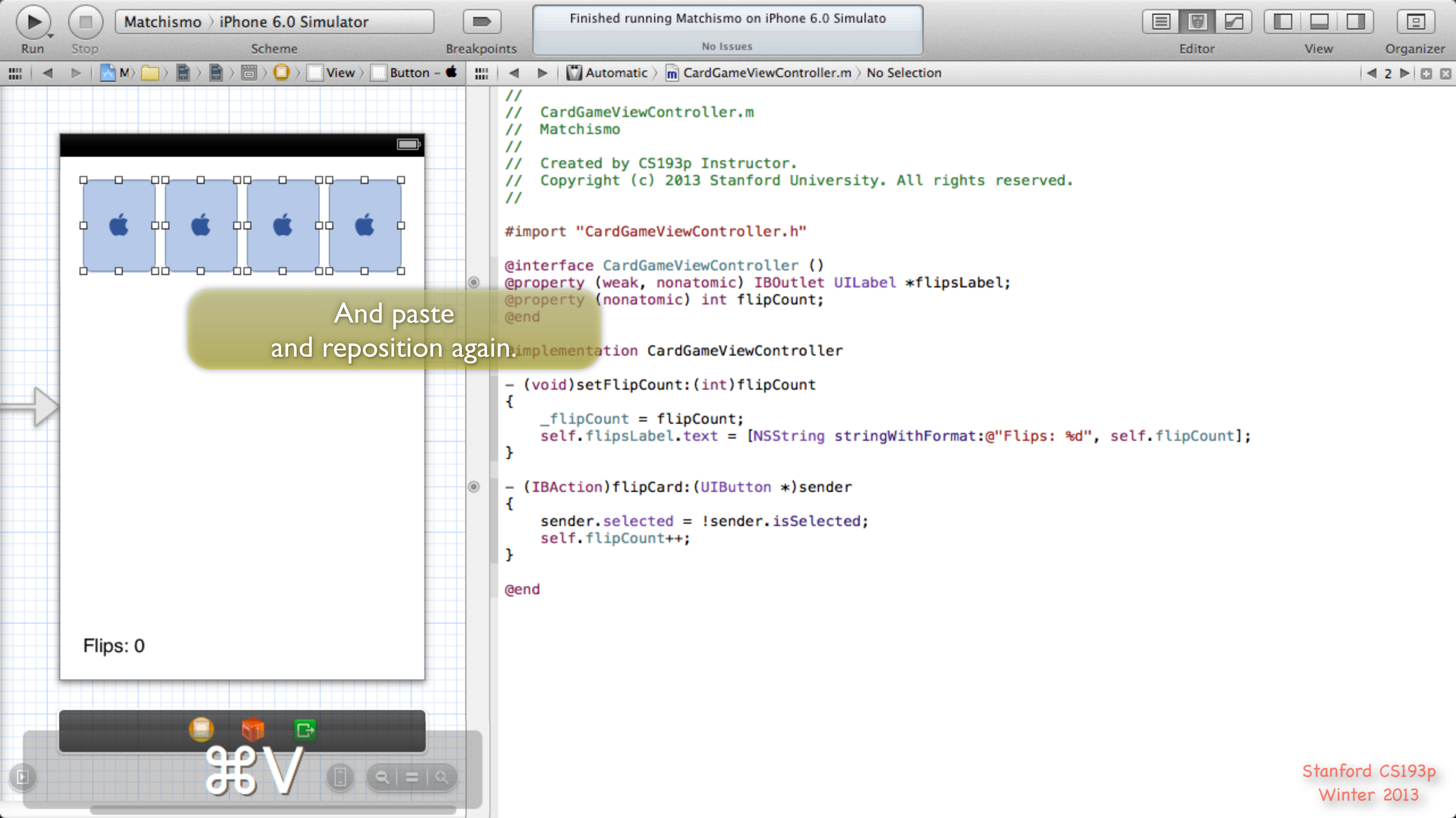
M › View › Button – Automatic › CardGameViewController.m › No Selection 2

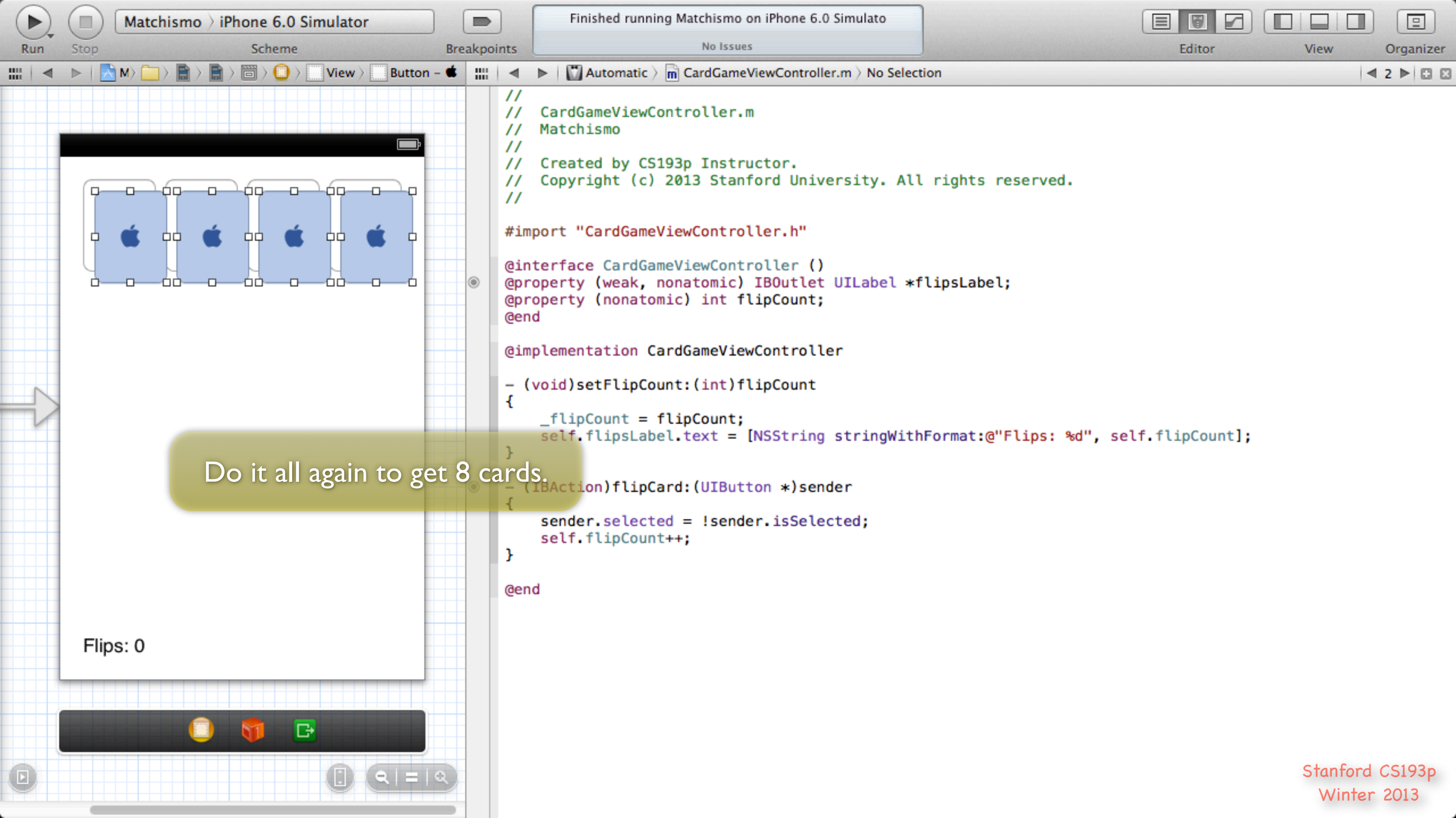


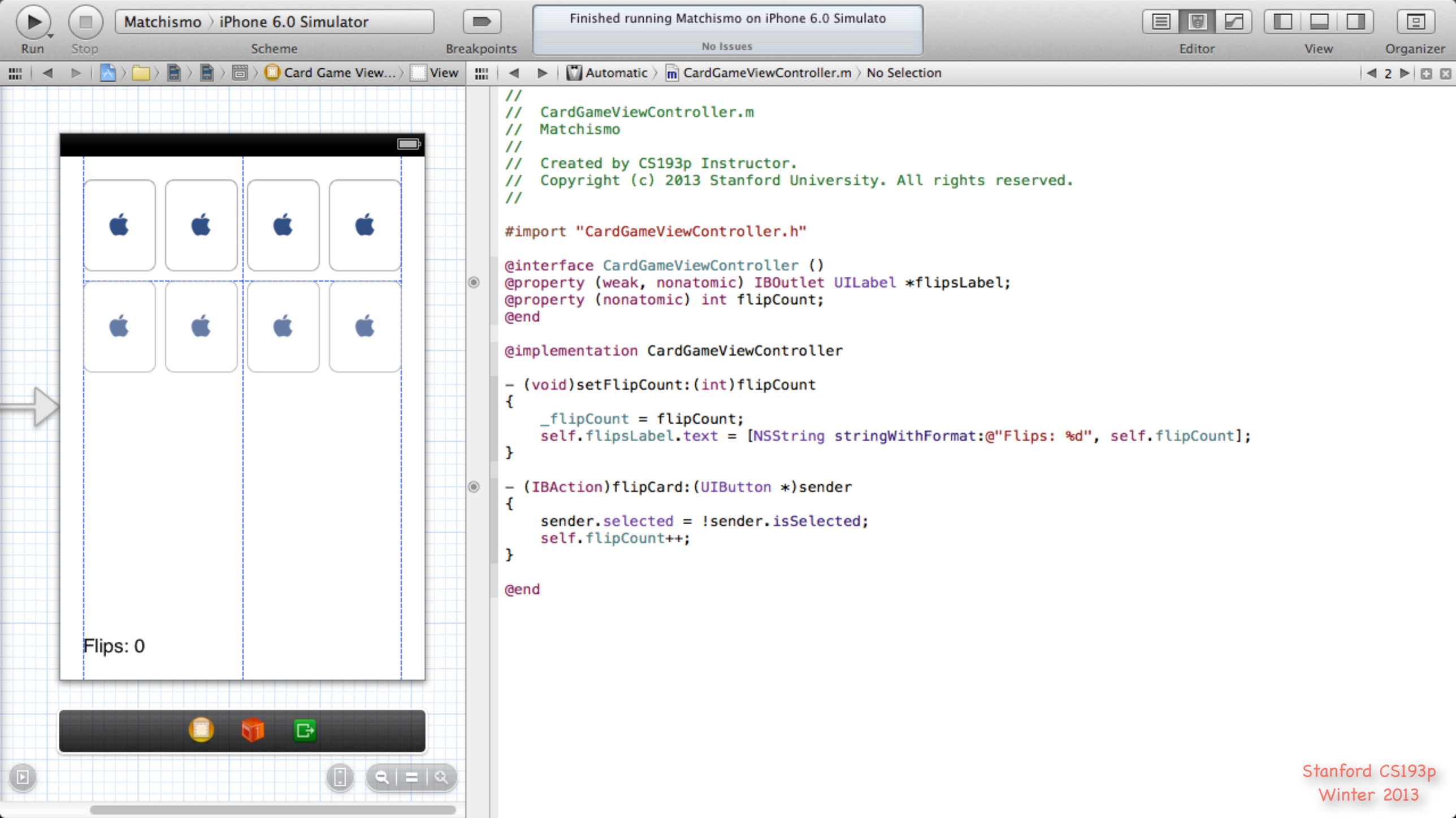
Select both cards by clicking on them, then copy.

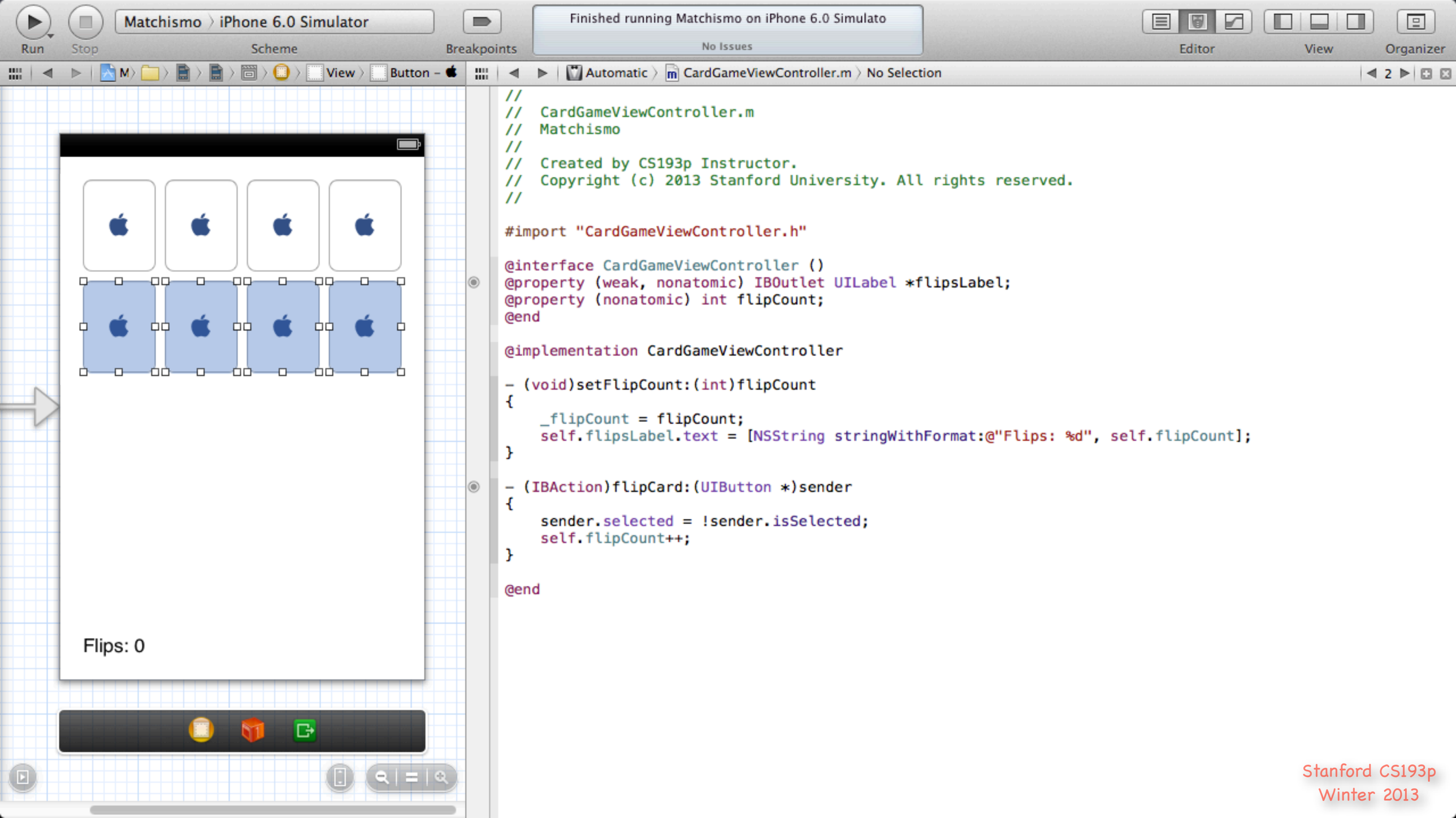

```
//
//  CardGameViewController.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University. All rights reserved.
//

#import "CardGameViewController.h"

@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;

@implementation CardGameViewController

- (void)setFlipCount:(int)flipCount
{
    _flipCount = flipCount;
    self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}

- (IBAction)flipCard:(UIButton *)sender
{
    sender.selected = !sender.isSelected;
    self.flipCount++;
}

@end
```

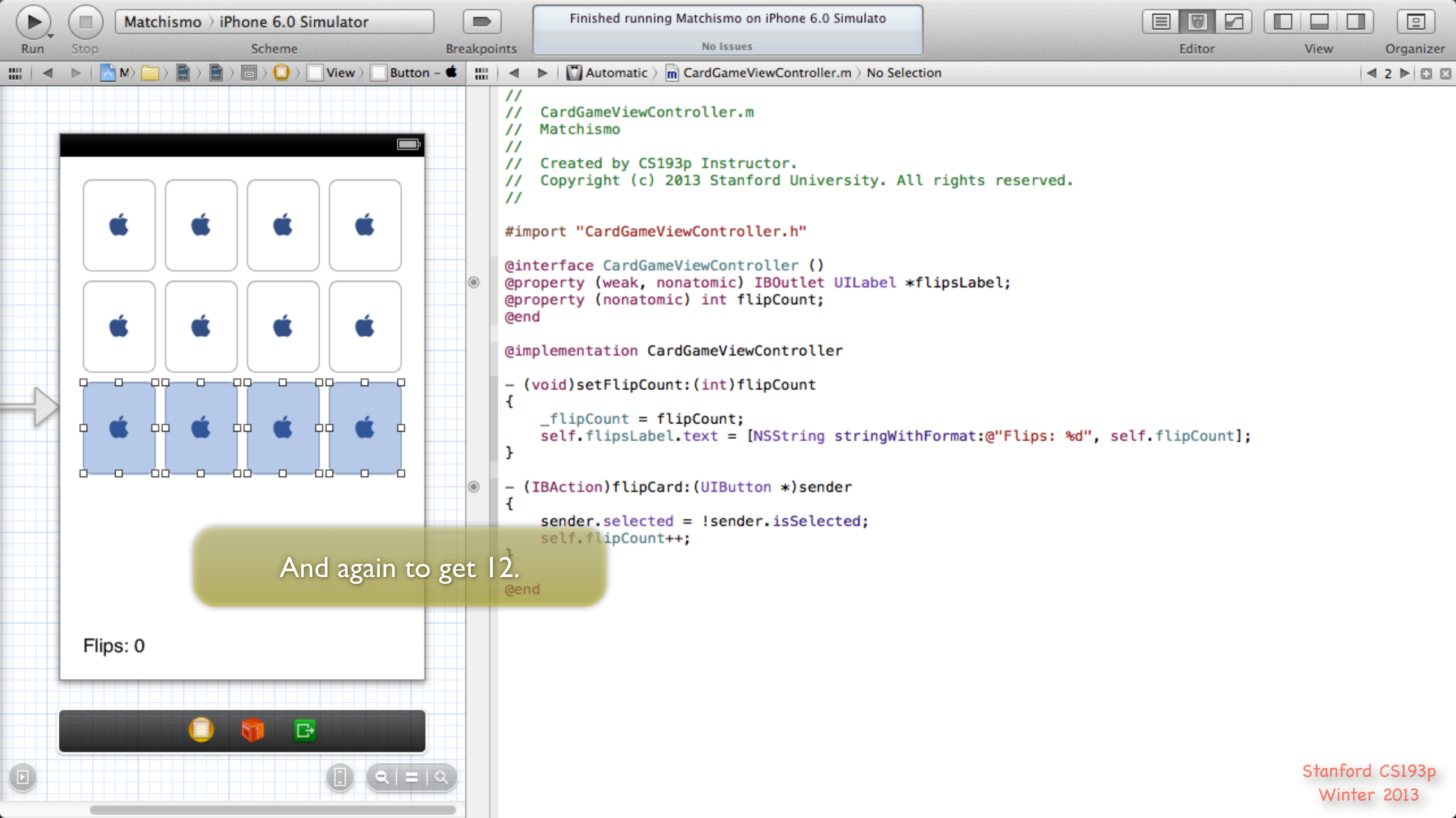
Stanford CS193p
Winter 2013

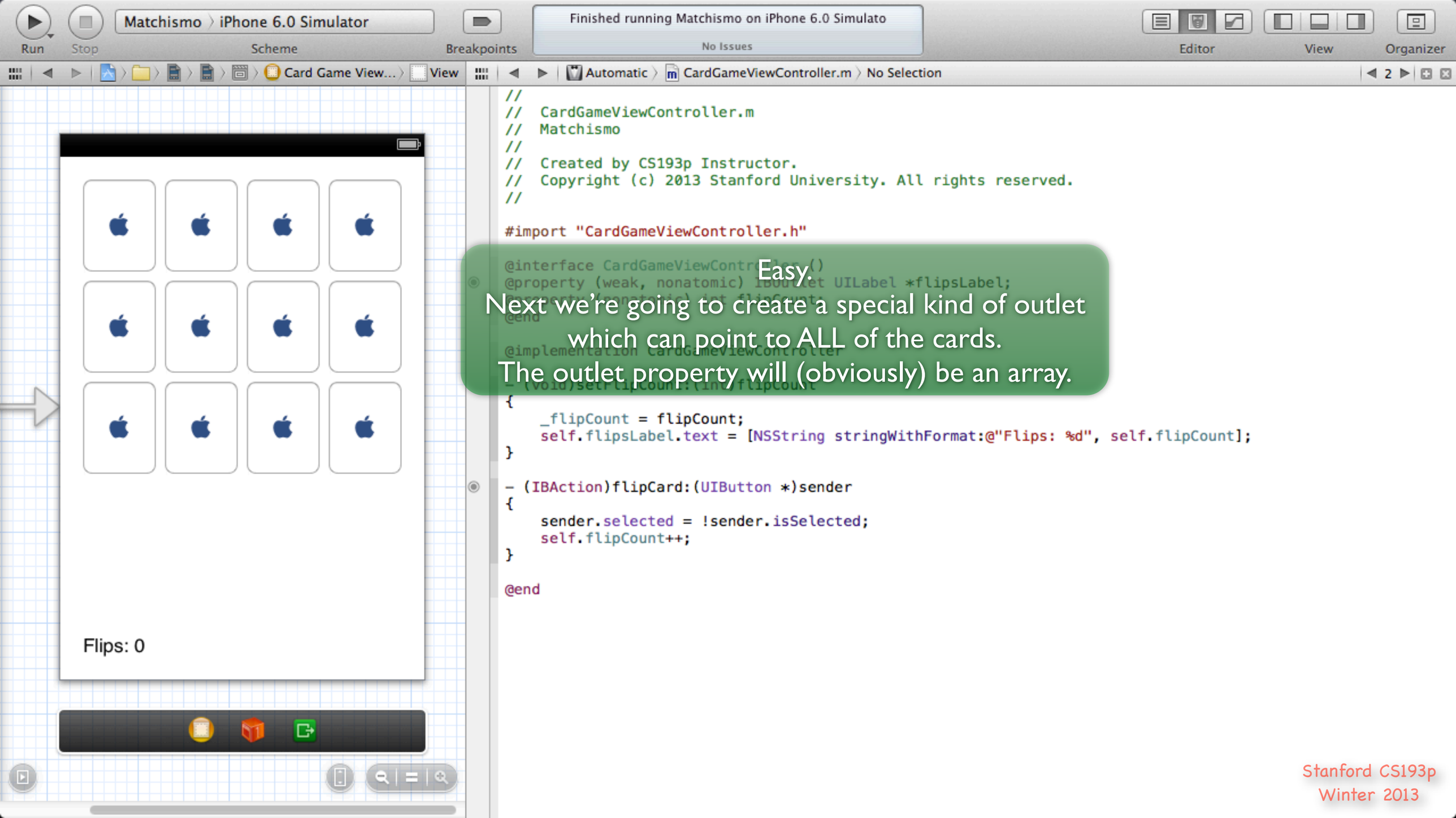

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic
CardGameViewController.m
No Selection
And paste
and reposition again.
Flips: 0
⌘V
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

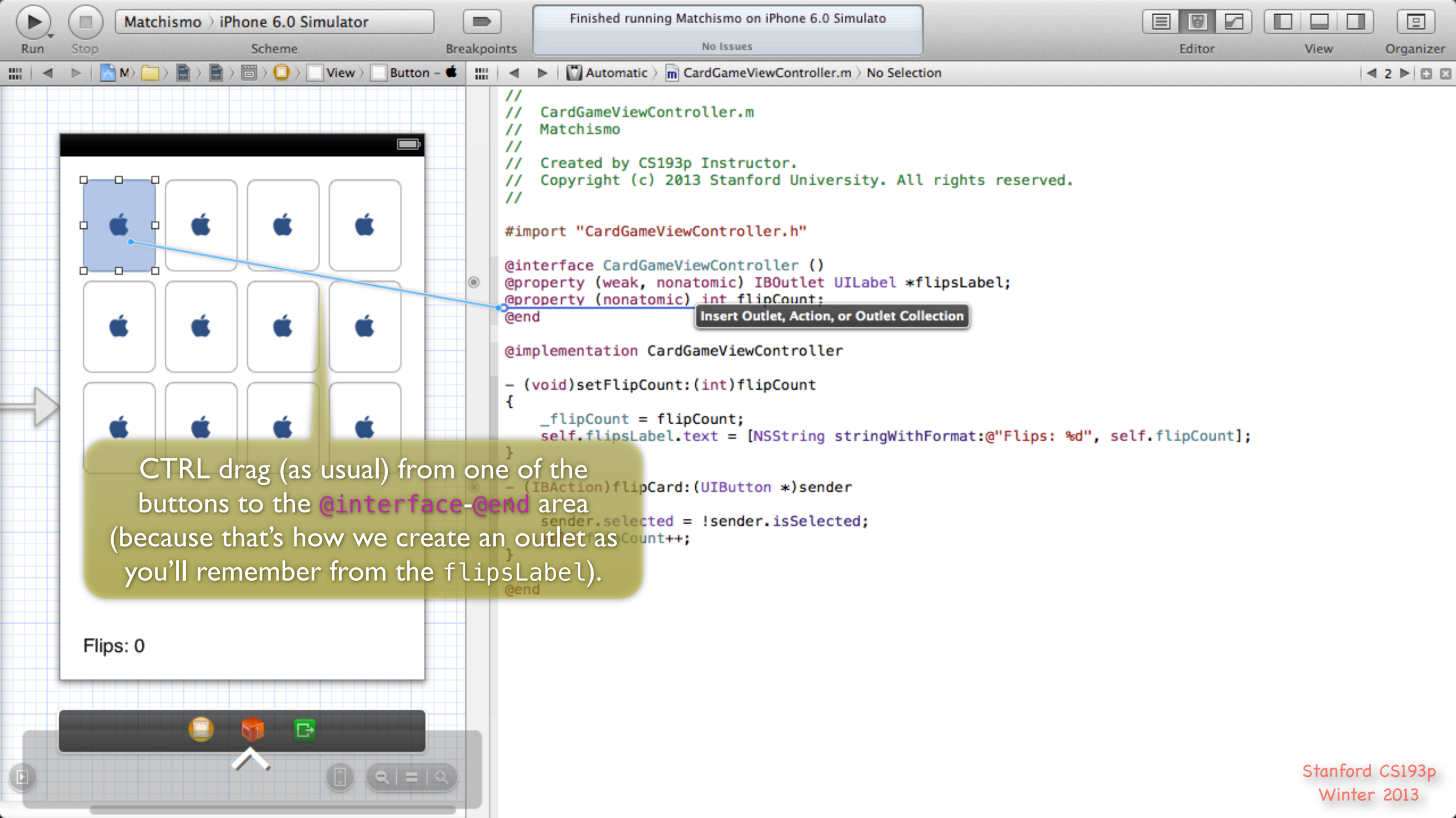

Matchismo
iPhone 6.0 Simulator
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Run
Stop
Scheme
Breakpoints
Editor
View
Organizer
Automatic
CardGameViewController.m
No Selection
Flips: 0
Do it all again to get 8 cards.
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

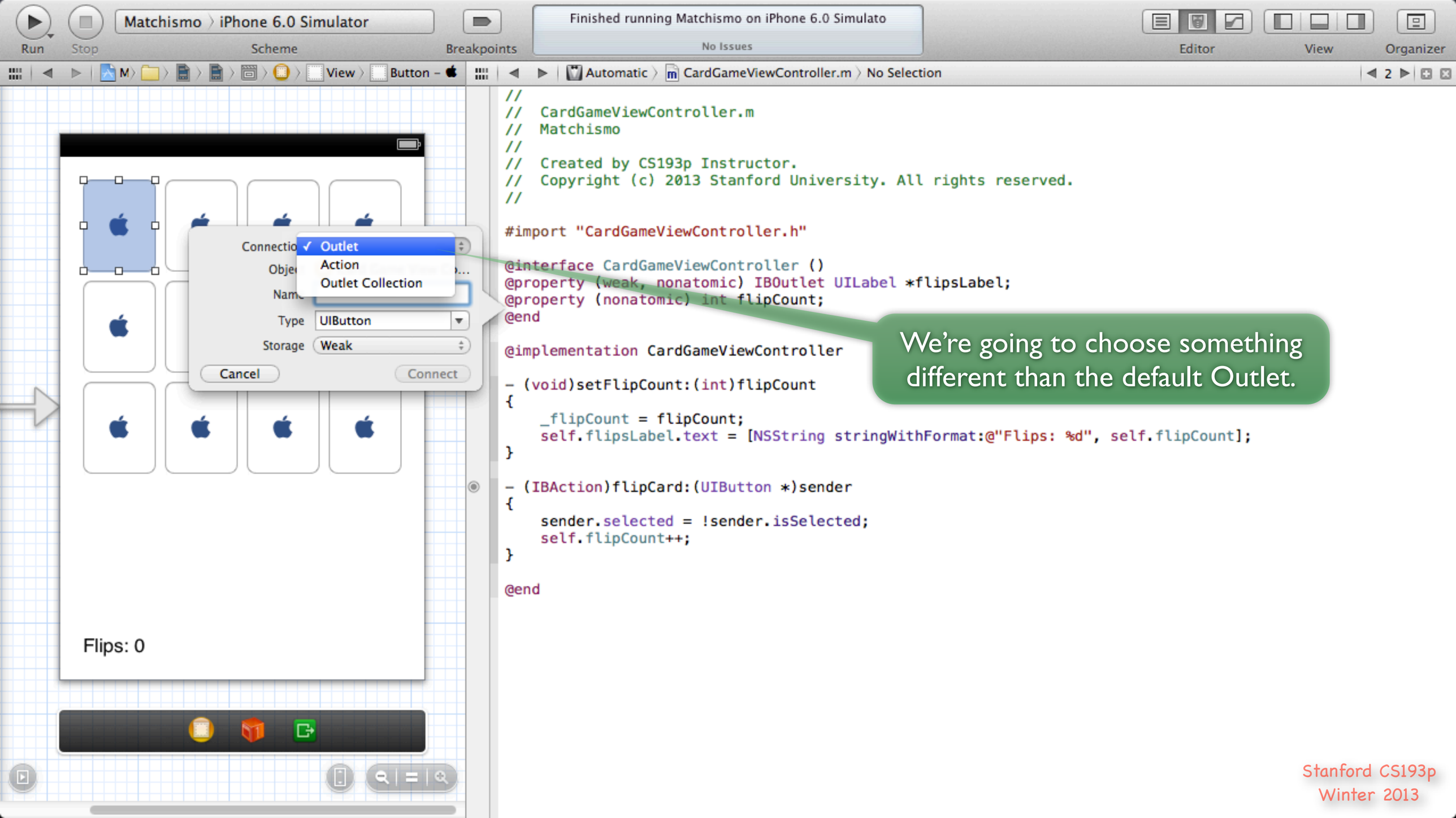

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 No Selection
2
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

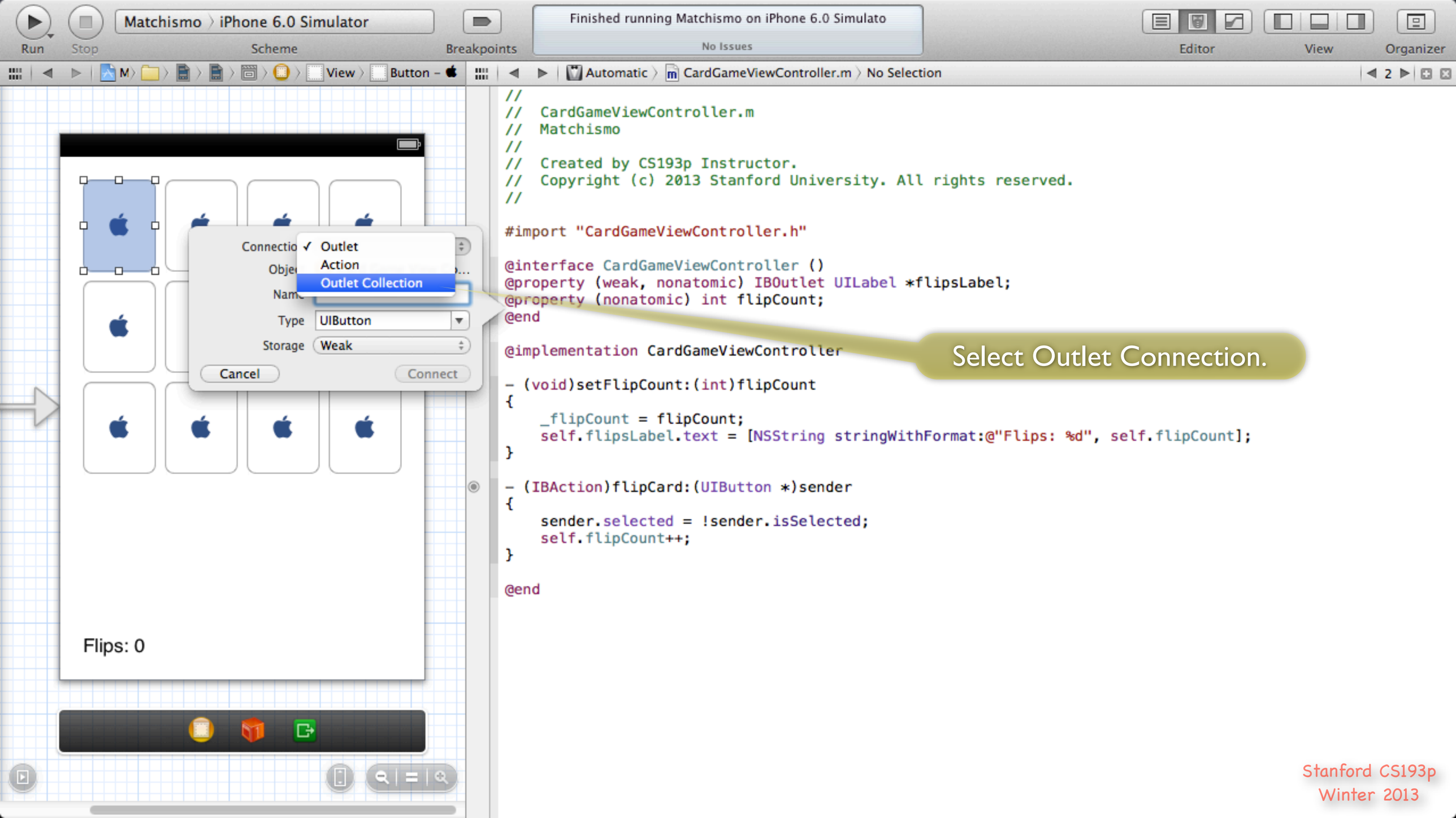

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
View
Button
Automatic
CardGameViewController.m
No Selection
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

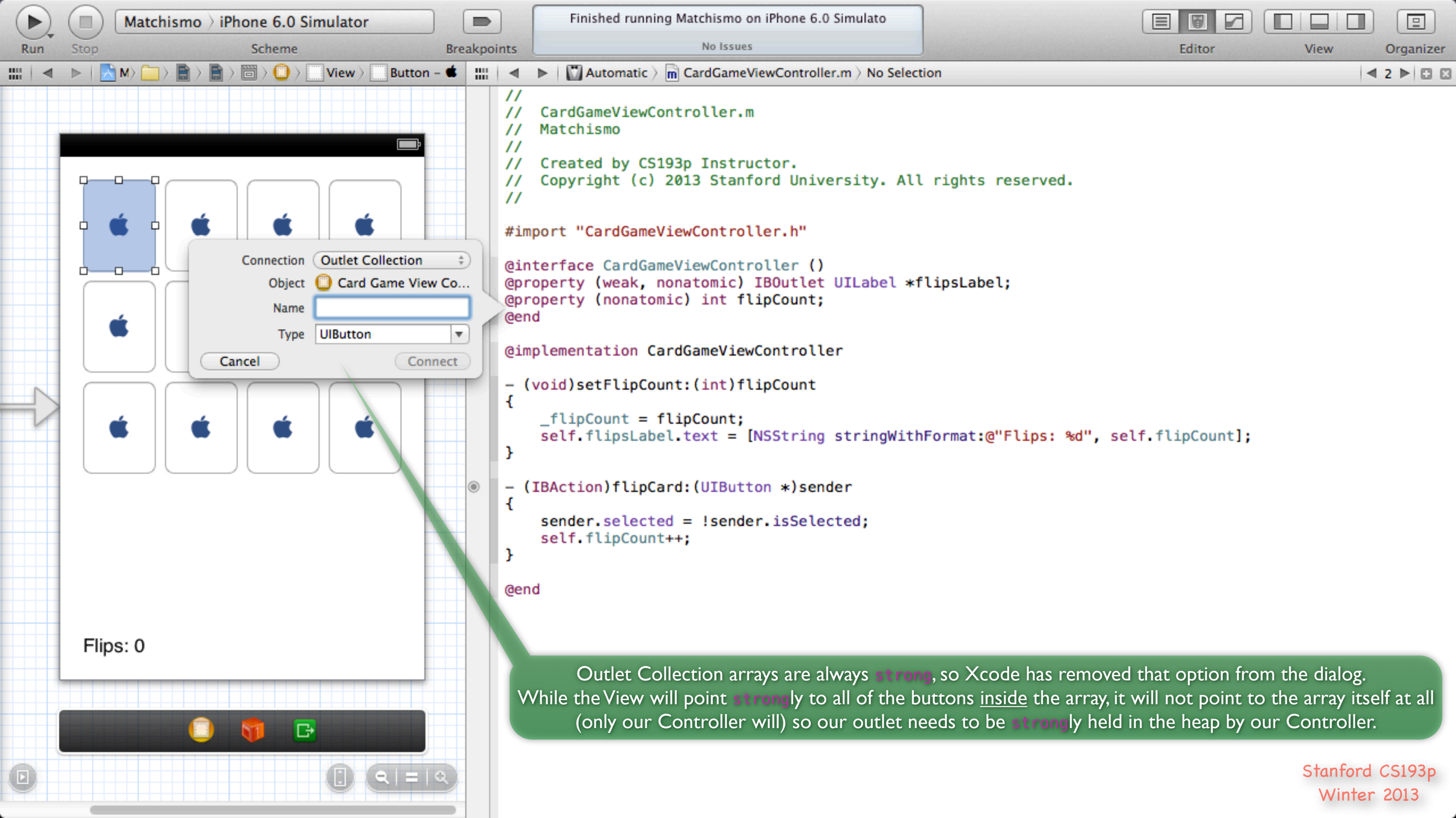

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
View
Button
Automatic
CardGameViewController.m
No Selection
Flips: 0
And again to get 12.
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end

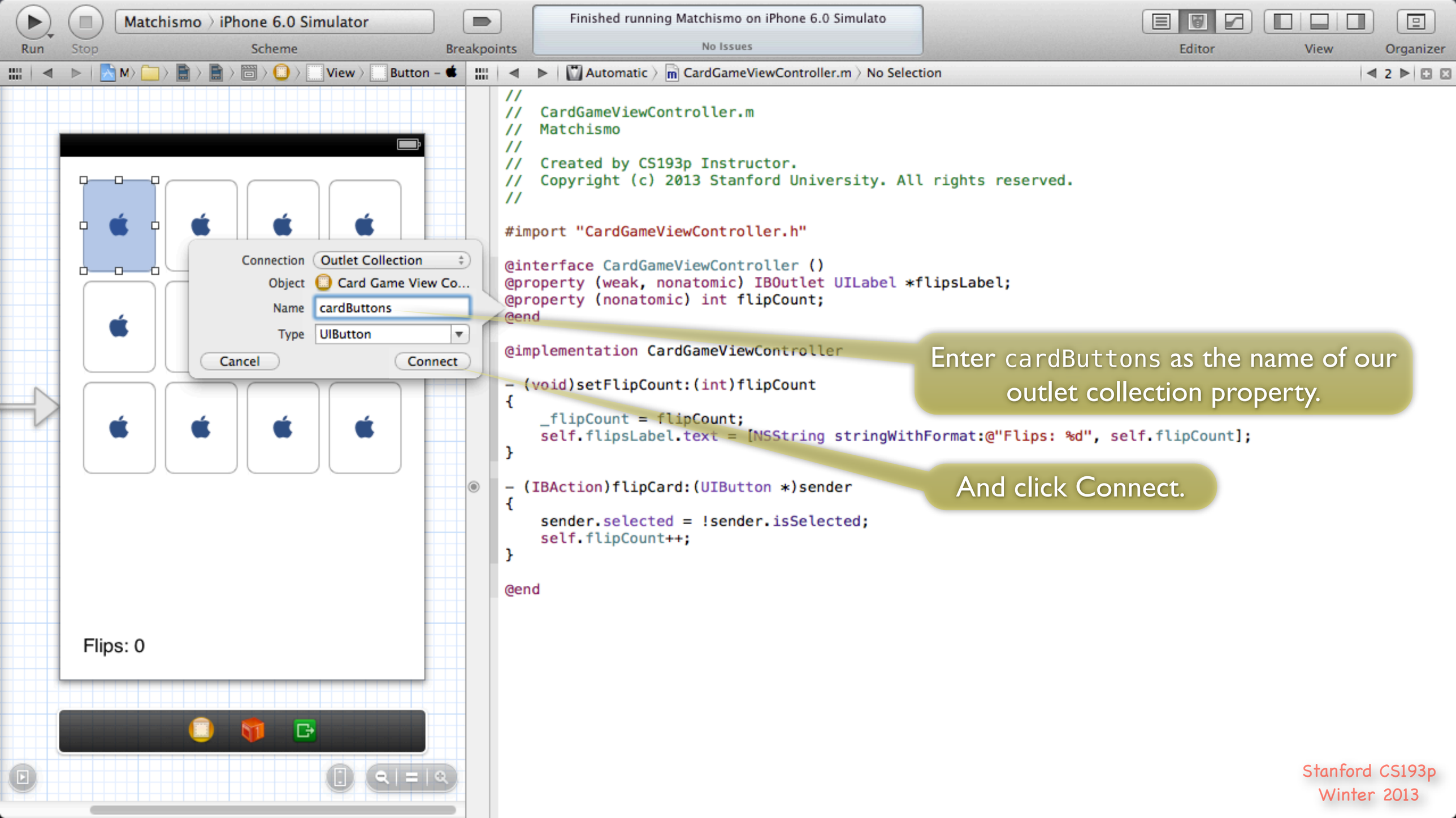

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 No Selection
2
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@implementation CardGameViewController
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Easy.
Next we're going to create a special kind of outlet which can point to ALL of the cards.
The outlet property will (obviously) be an array.
Stanford CS193p
Winter 2013

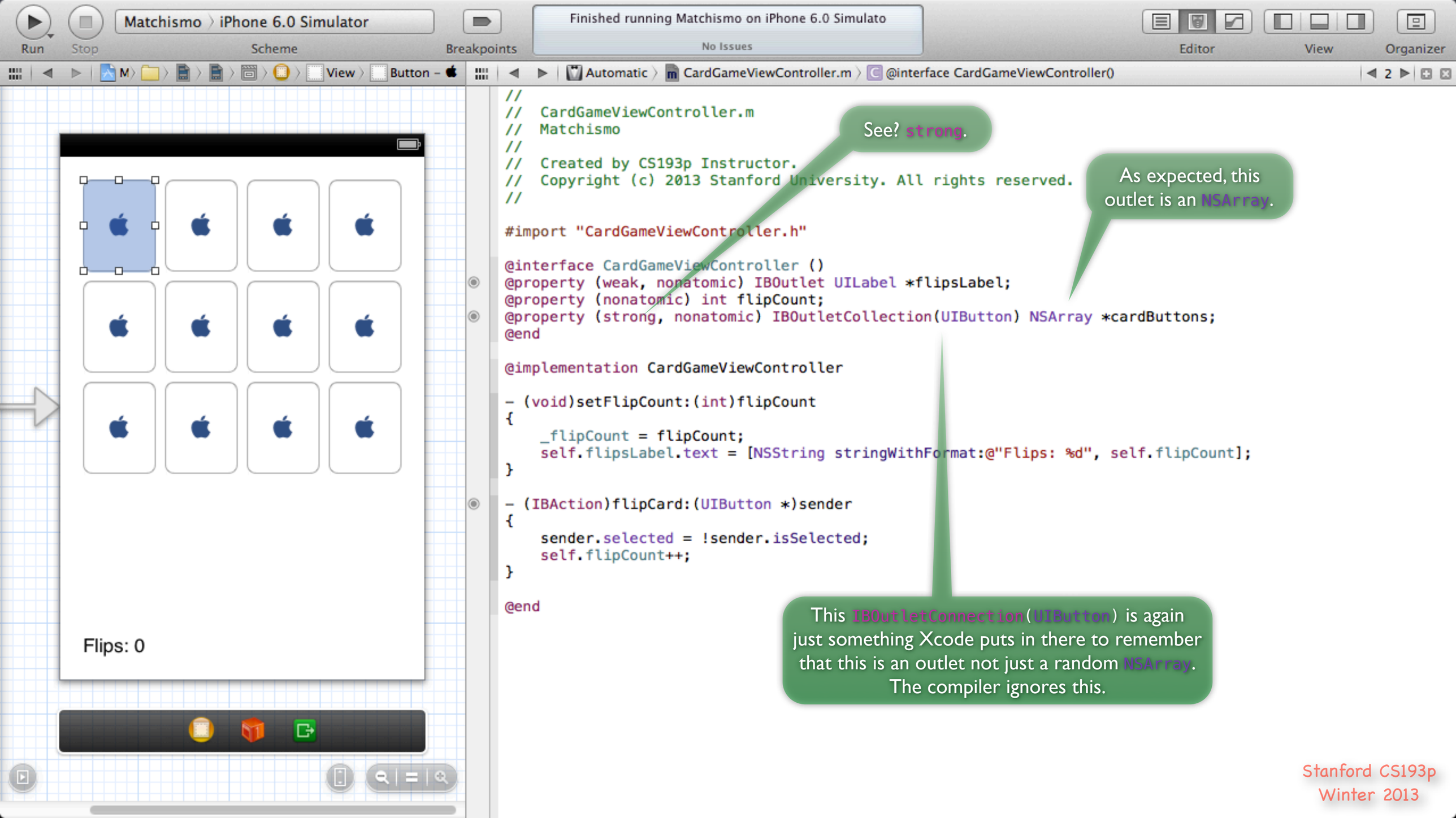

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic
CardGameViewController.m
No Selection
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
Insert Outlet, Action, or Outlet Collection
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
CTRL drag (as usual) from one of the buttons to the @interface-@end area (because that's how we create an outlet as you'll remember from the flipsLabel).
Flips: 0
Stanford CS193p
Winter 2013

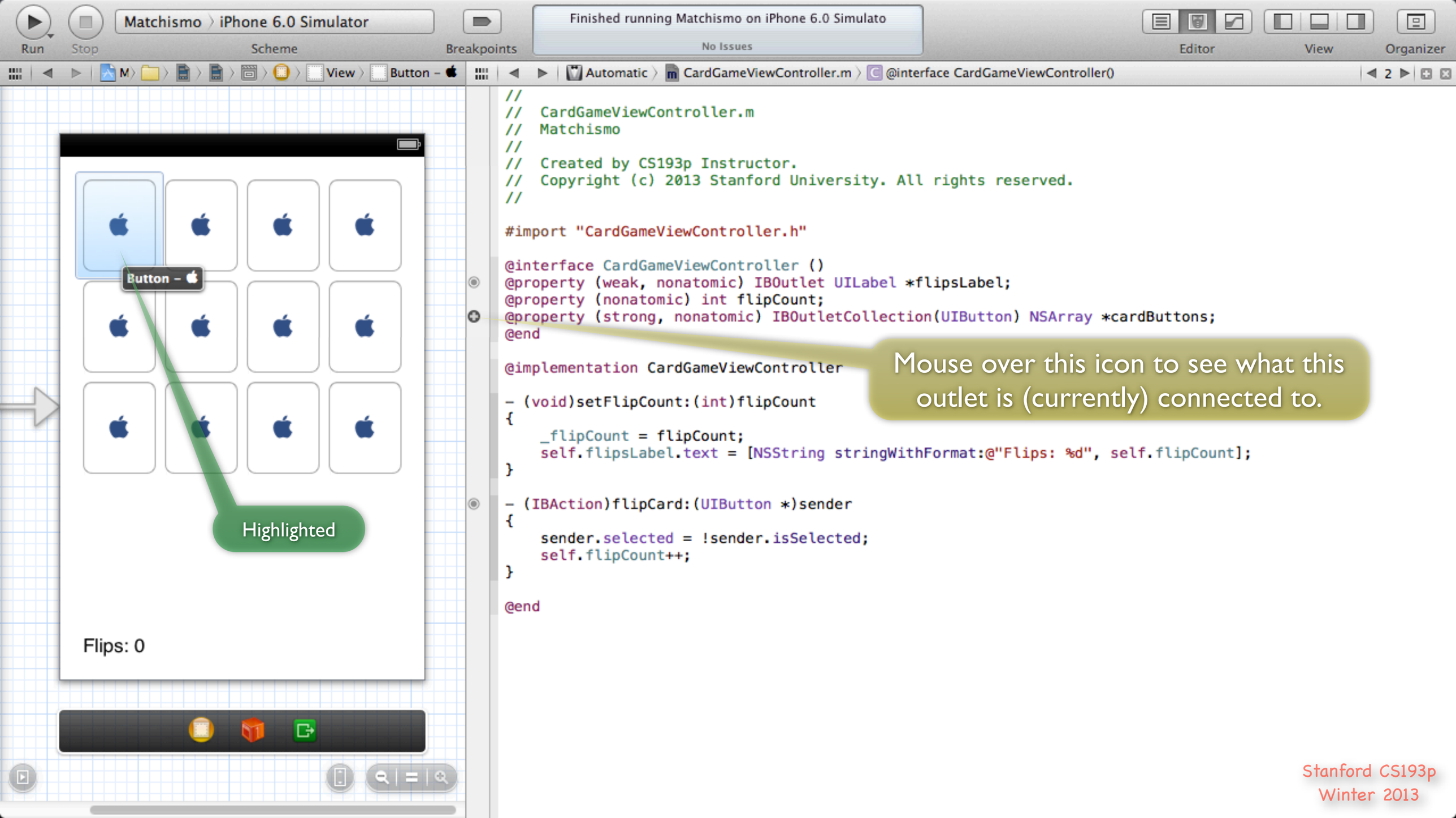

Matchismo 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic 〉 CardGameViewController.m 〉 No Selection
Outlet
Action
Outlet Collection
Type
UIButton
Storage
Weak
Cancel
Connect
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
We're going to choose something different than the default Outlet.
Stanford CS193p
Winter 2013

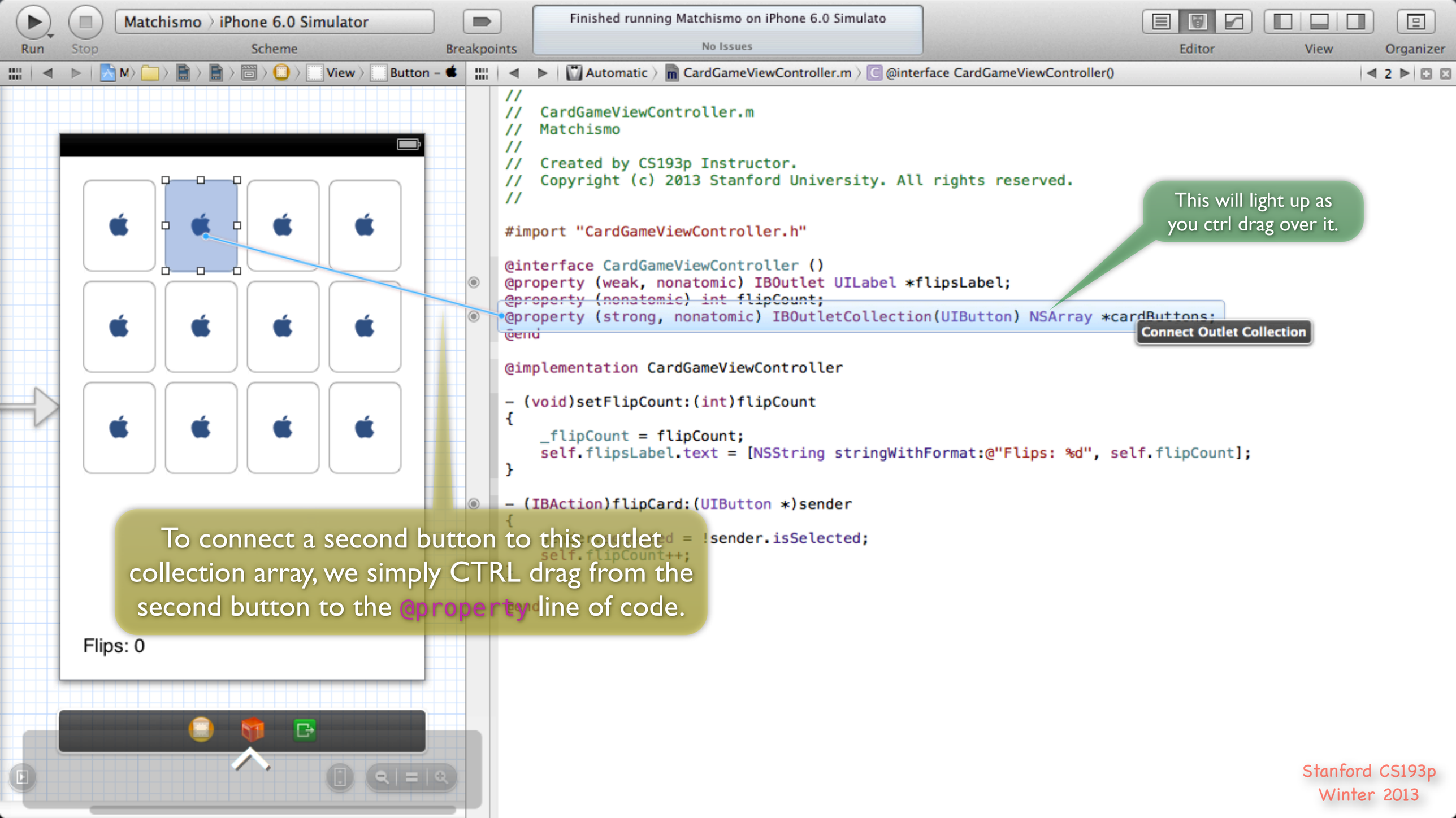

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
View 〉 Button –
Automatic 〉 CardGameViewController.m 〉 No Selection
Connectio
Outlet
Action
Outlet Collection
Obje
Nam
Type
UIButton
Storage
Weak
Cancel
Connect
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Select Outlet Connection.
Stanford CS193p
Winter 2013

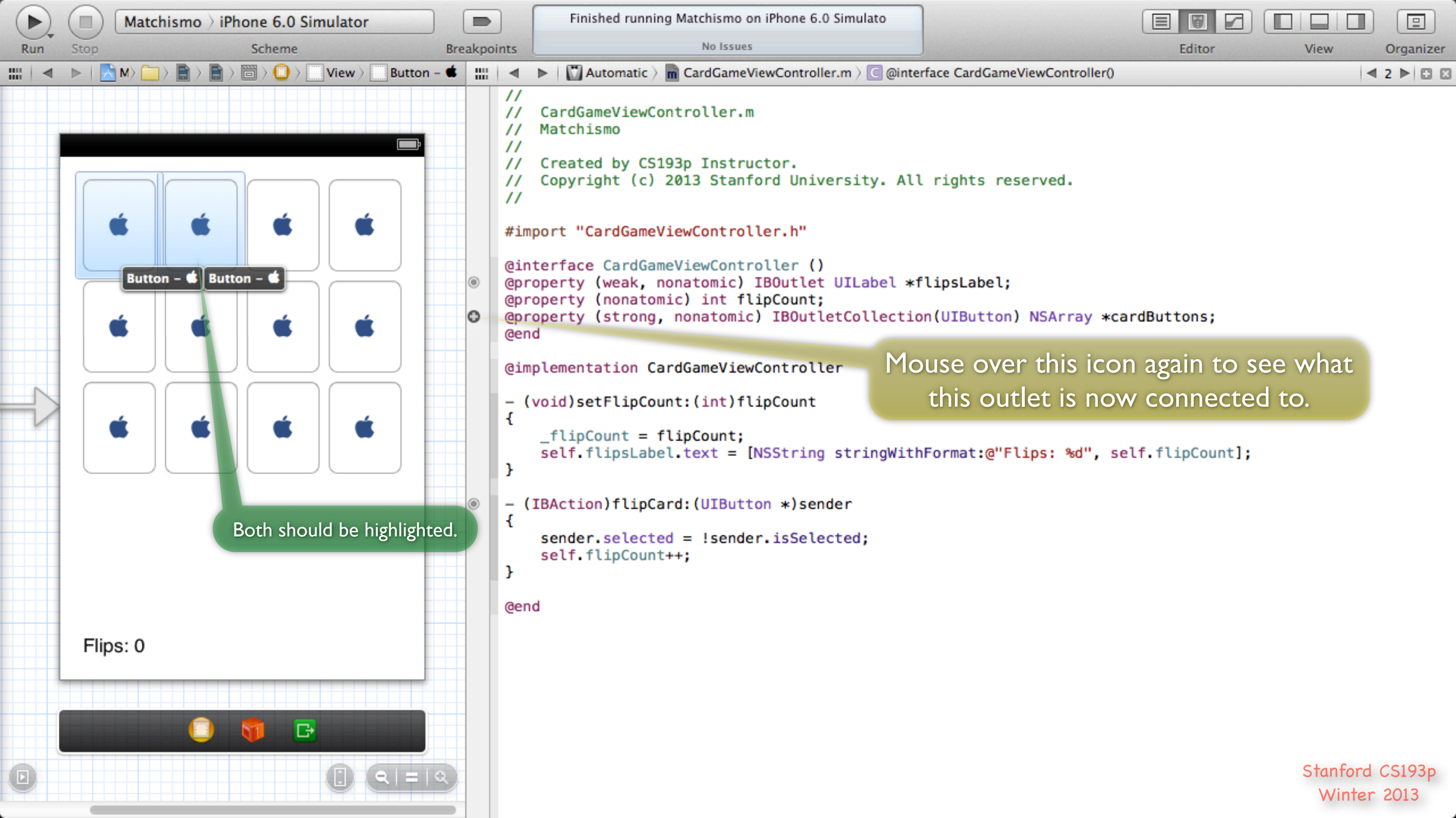

Matchismo 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic 〉 CardGameViewController.m 〉 No Selection
Connection Outlet Collection
Object Card Game View Co...
Name
Type UIButton
Cancel
Connect
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Outlet Collection arrays are always strong, so Xcode has removed that option from the dialog.
While the View will point strongly to all of the buttons inside the array, it will not point to the array itself at all
(only our Controller will) so our outlet needs to be strongly held in the heap by our Controller.
Stanford CS193p
Winter 2013

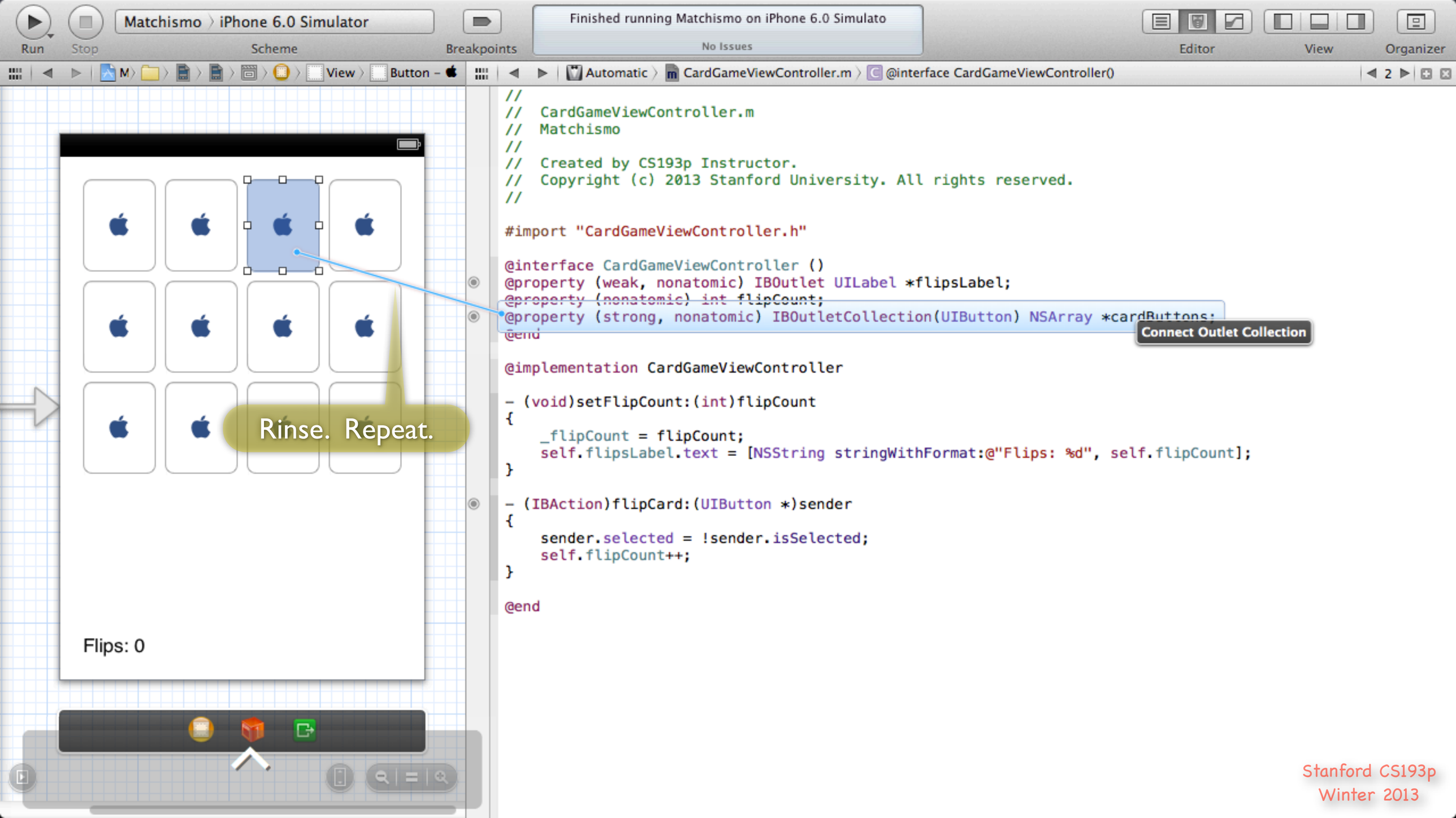

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic
CardGameViewController.m
No Selection
Connection
Outlet Collection
Object
Card Game View Co...
Name
cardButtons
Type
UIButton
Cancel
Connect
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Enter cardButtons as the name of our outlet collection property.
And click Connect.
Stanford CS193p
Winter 2013

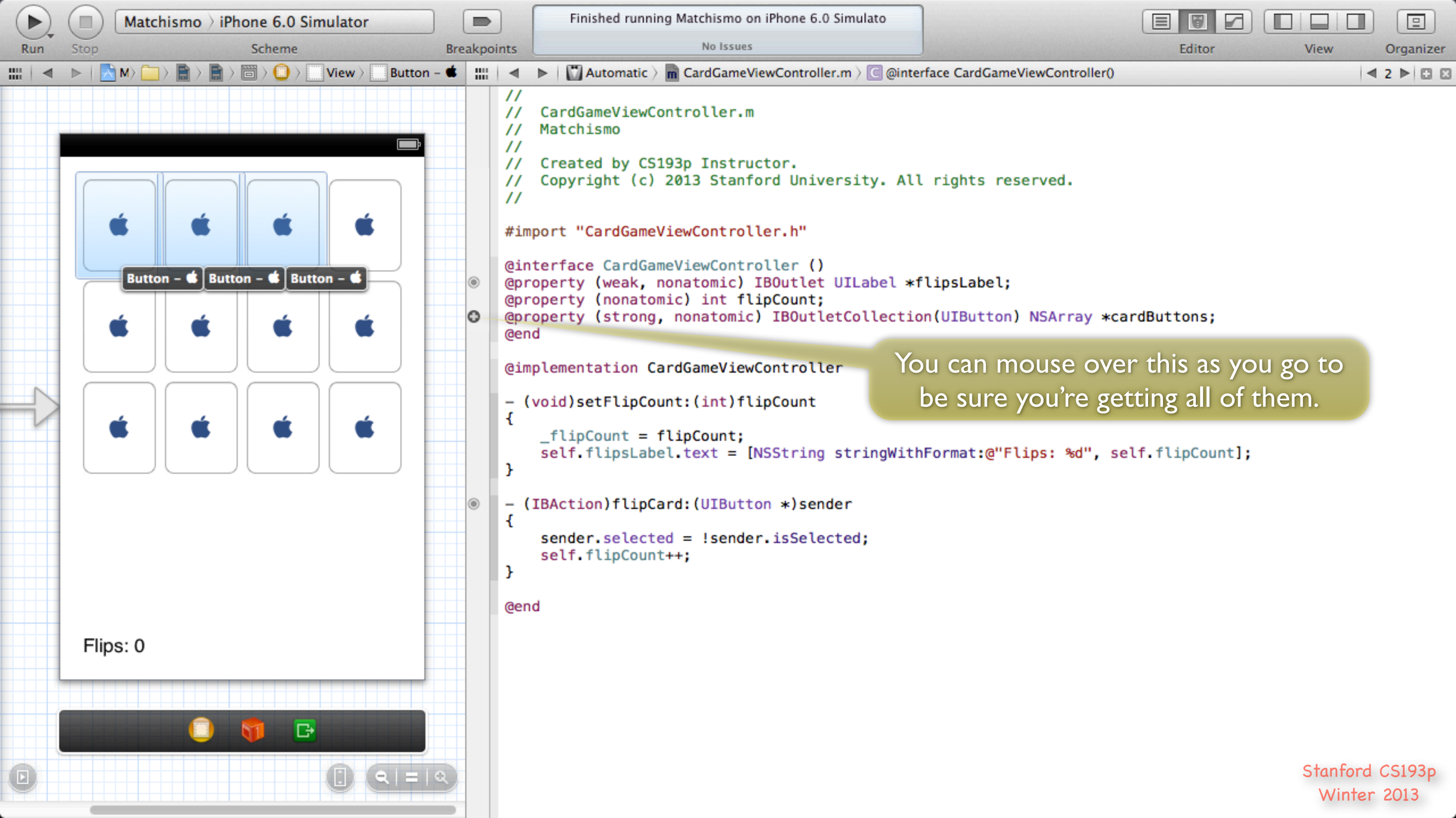

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
See? strong.
As expected, this outlet is an NSArray.
This IBOutletConnection(UIButton) is again just something Xcode puts in there to remember that this is an outlet not just a random NSArray. The compiler ignores this.
Stanford CS193p
Winter 2013

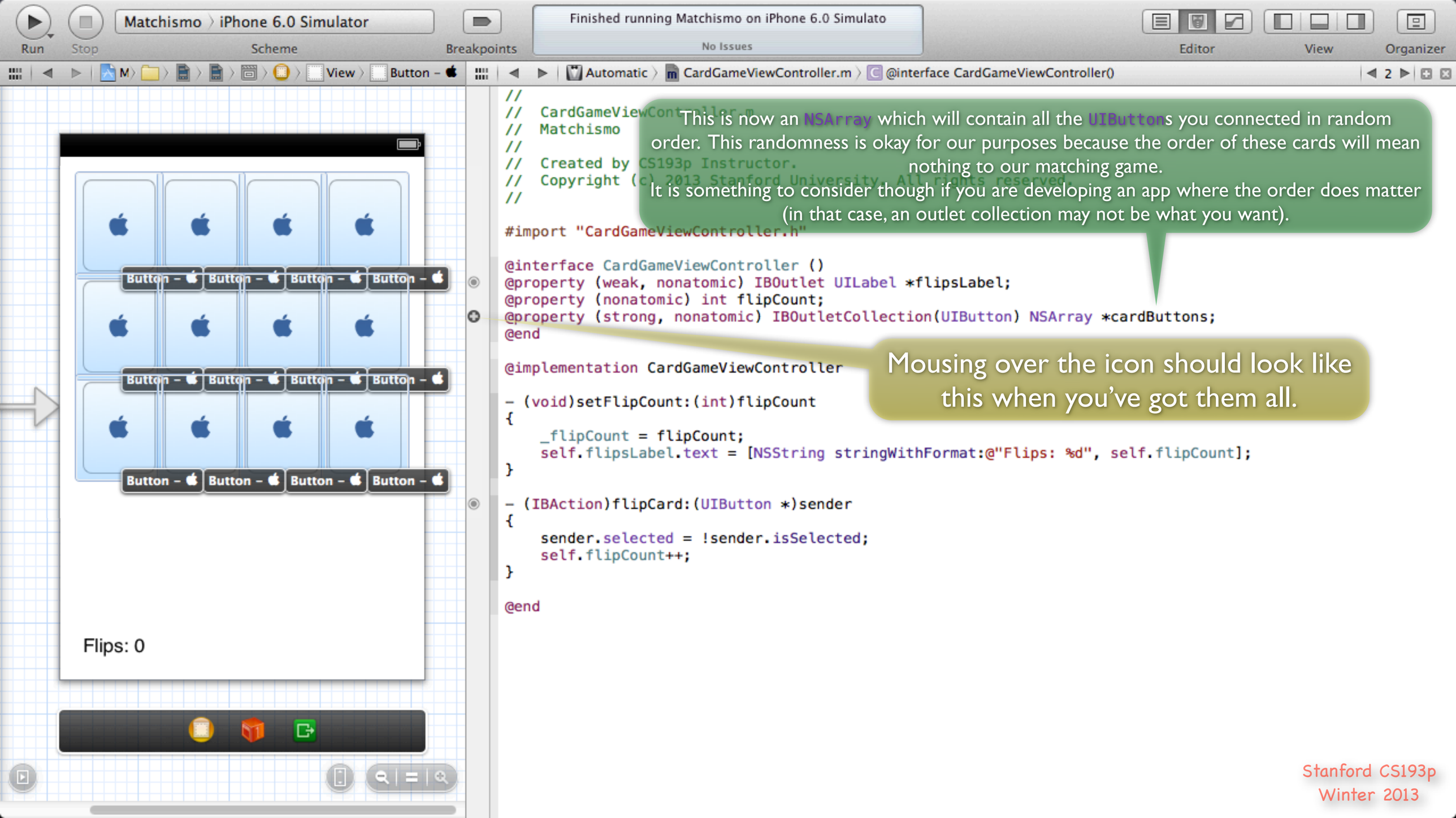

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic
CardGameViewController.m
@interface CardGameViewController()
Button –
Highlighted
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Mouse over this icon to see what this outlet is (currently) connected to.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
View 〉 Button
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.isSelected = !sender.isSelected;
self.flipCount++;
}
This will light up as you ctrl drag over it.
Connect Outlet Collection
To connect a second button to this outlet collection array, we simply CTRL drag from the second button to the @property line of code.
Flips: 0
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
Button – Button –
Flips: 0
Both should be highlighted.
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Mouse over this icon again to see what this outlet is now connected to.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
View 〉 Button
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@end
Connect Outlet Collection
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Rinse. Repeat.
Flips: 0
Stanford CS193p
Winter 2013

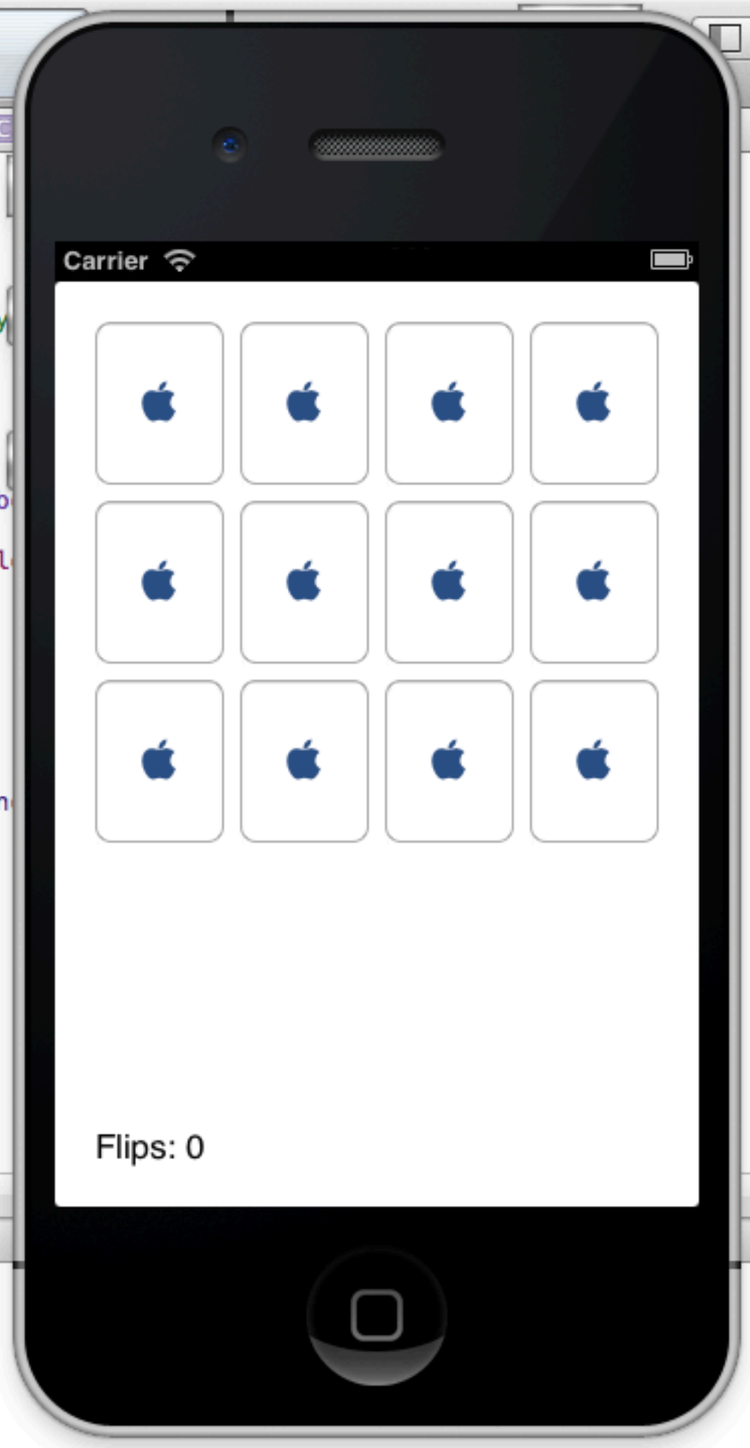

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Button –
Button –
Button –
Flips: 0
Automatic
CardGameViewController.m
@interface CardGameViewController()
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
You can mouse over this as you go to be sure you're getting all of them.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
Button –
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
This is now an NSArray which will contain all the UIButtons you connected in random order. This randomness is okay for our purposes because the order of these cards will mean nothing to our matching game.
It is something to consider though if you are developing an app where the order does matter (in that case, an outlet collection may not be what you want).
Mousing over the icon should look like this when you've got them all.
Stanford CS193p
Winter 2013

Run Stop Matchismo › iPhone 6.0 Simulator Scheme Breakpoints Running Matchismo on iPhone 6.0 Simulator No Issues View Organizer

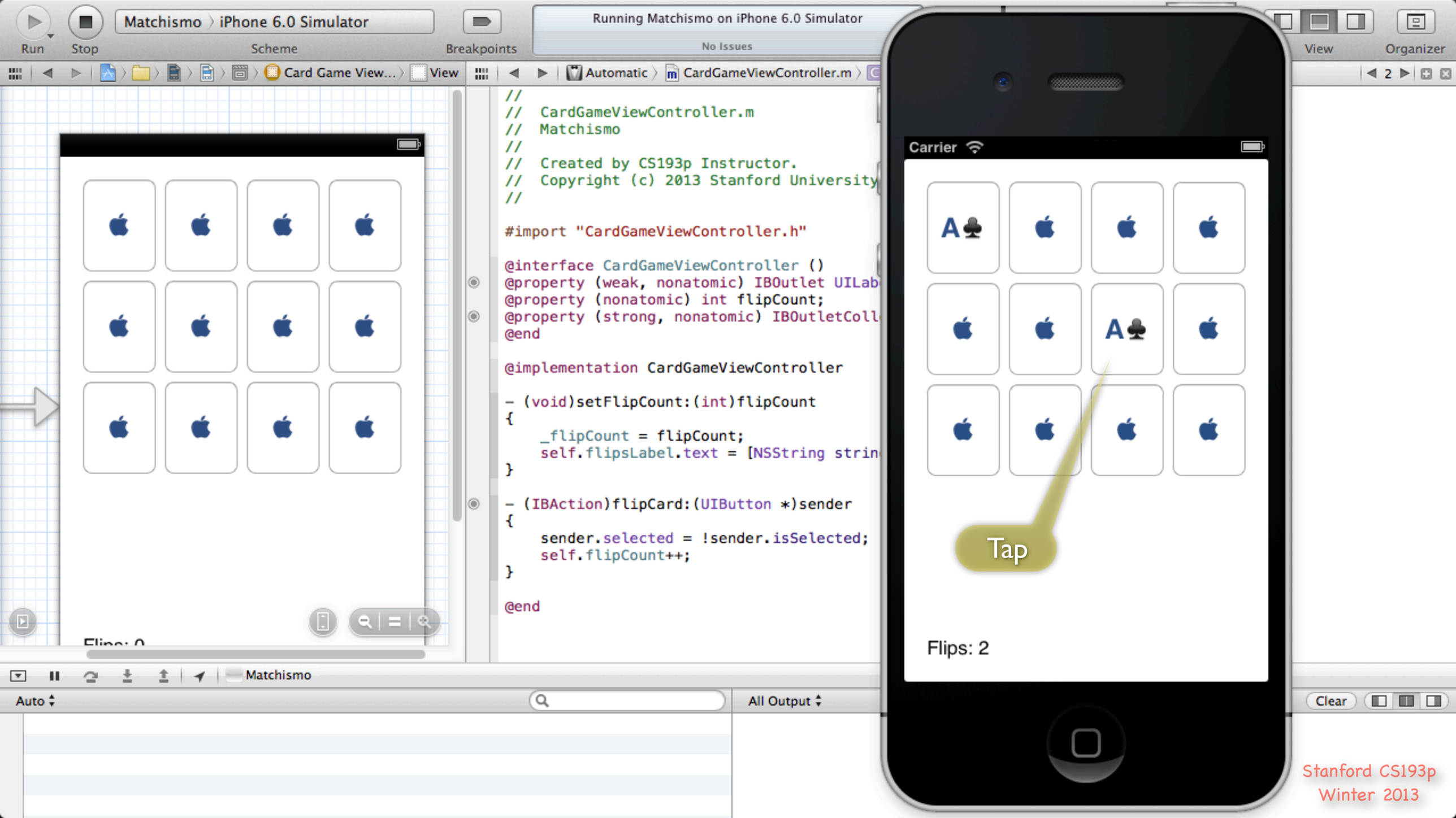
Let's Run and see what we've got so far.

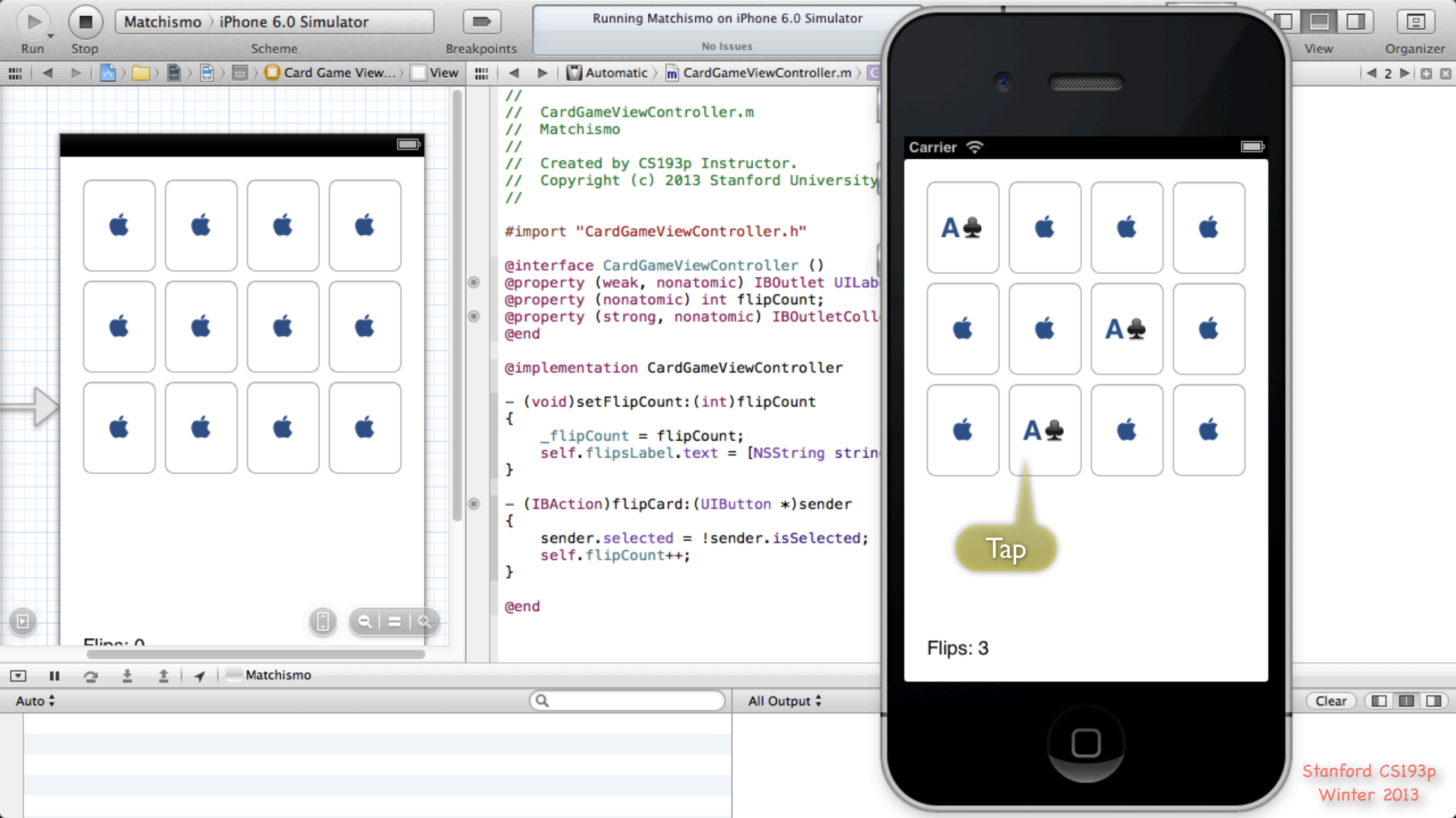
```
//
//  CardGameViewController.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University
//

#import "CardGameViewController.h"

@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@end

@implementation CardGameViewController

- (void)setFlipCount:(int)flipCount
{
    _flipCount = flipCount;
    self.flipsLabel.text = [NSString strin
}

- (IBAction)flipCard:(UIButton *)sender
{
    sender.selected = !sender.isSelected;
    self.flipCount++;
}

@end
```

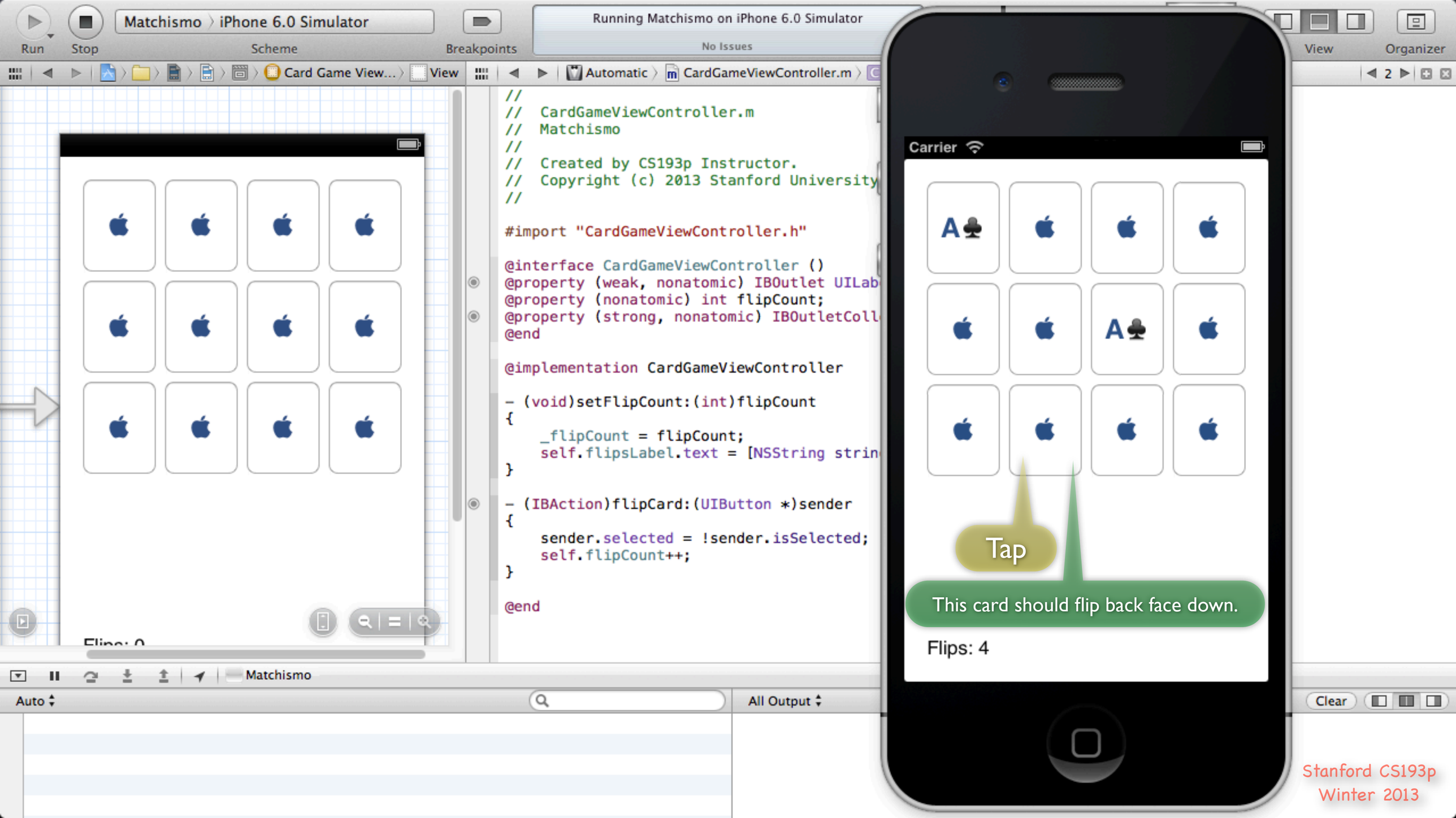


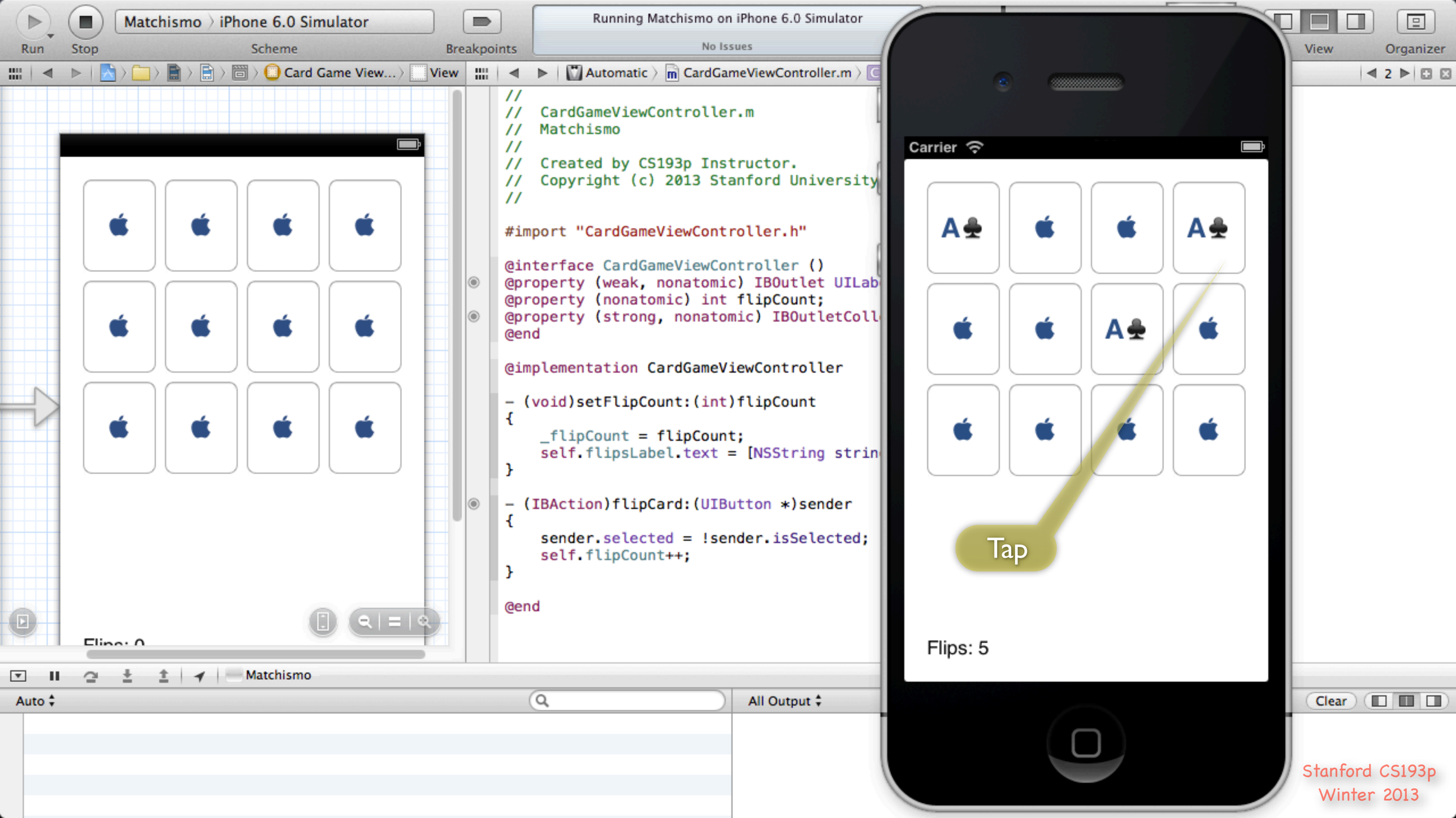
Stanford CS193p
Winter 2013

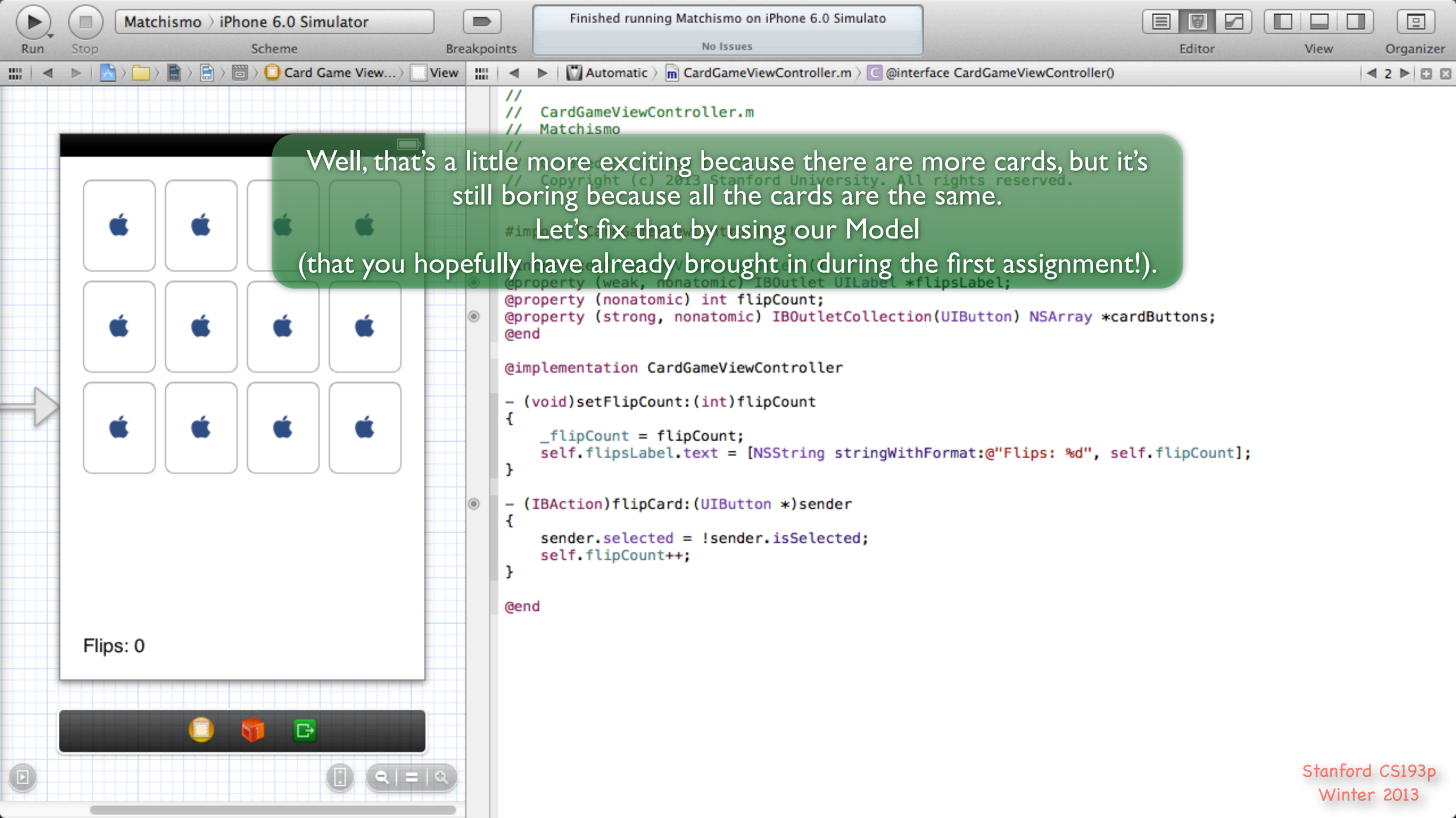

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
2
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Tapping should flip over any of the cards.
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♣
Flips: 1
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
2
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♣
A♣
Tap
Flips: 2
Stanford CS193p
Winter 2013


Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♣
A♣
A♣
Tap
Flips: 3
Stanford CS193p
Winter 2013


Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♣
A♣
Tap
This card should flip back face down.
Flips: 4
Stanford CS193p
Winter 2013

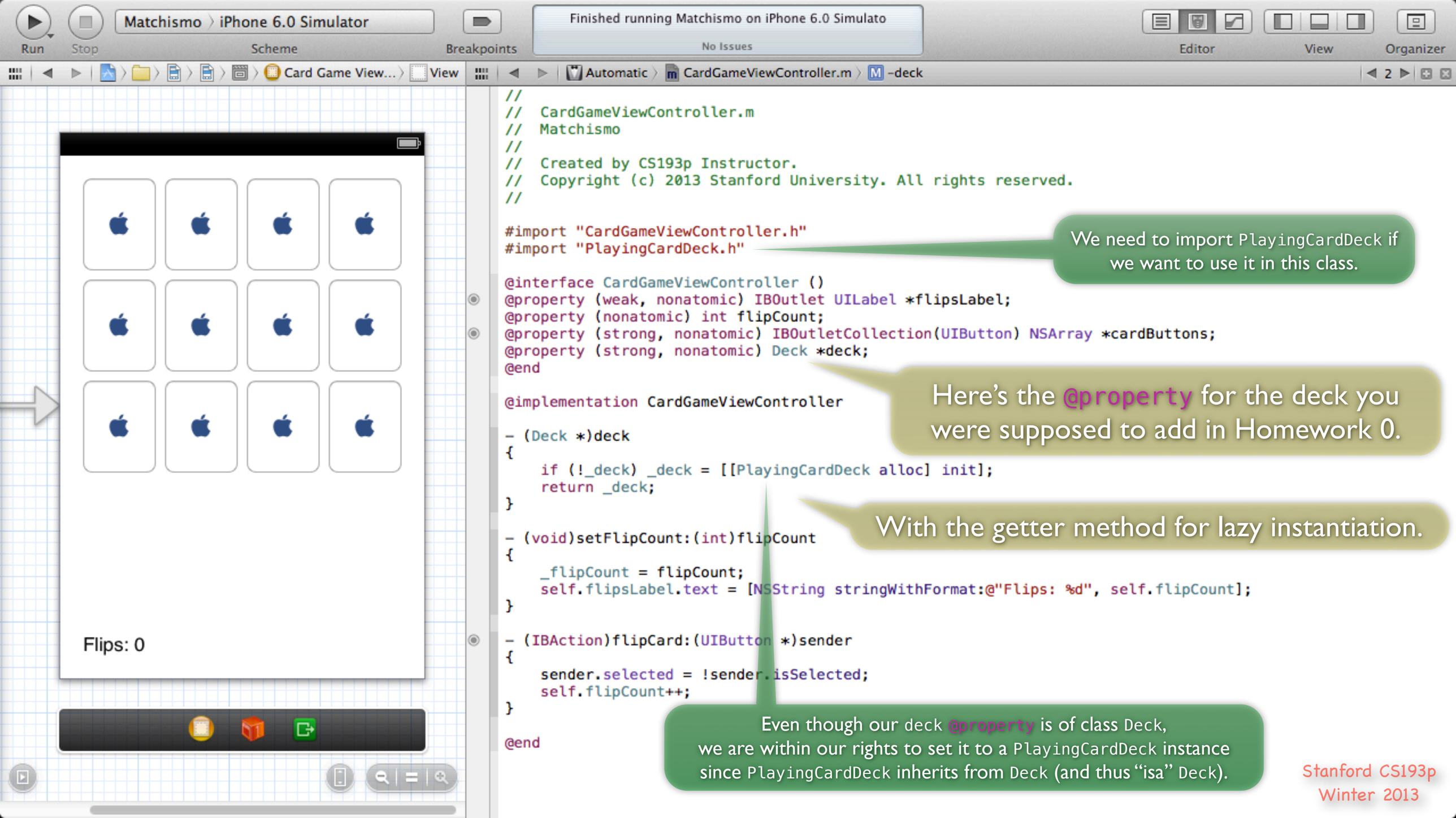

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
2
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University
//
#import "CardGameViewController.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@end
@implementation CardGameViewController
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♣
A♣
A♣
Tap
Flips: 5
Stanford CS193p
Winter 2013



Well, that's a little more exciting because there are more cards, but it's still boring because all the cards are the same.

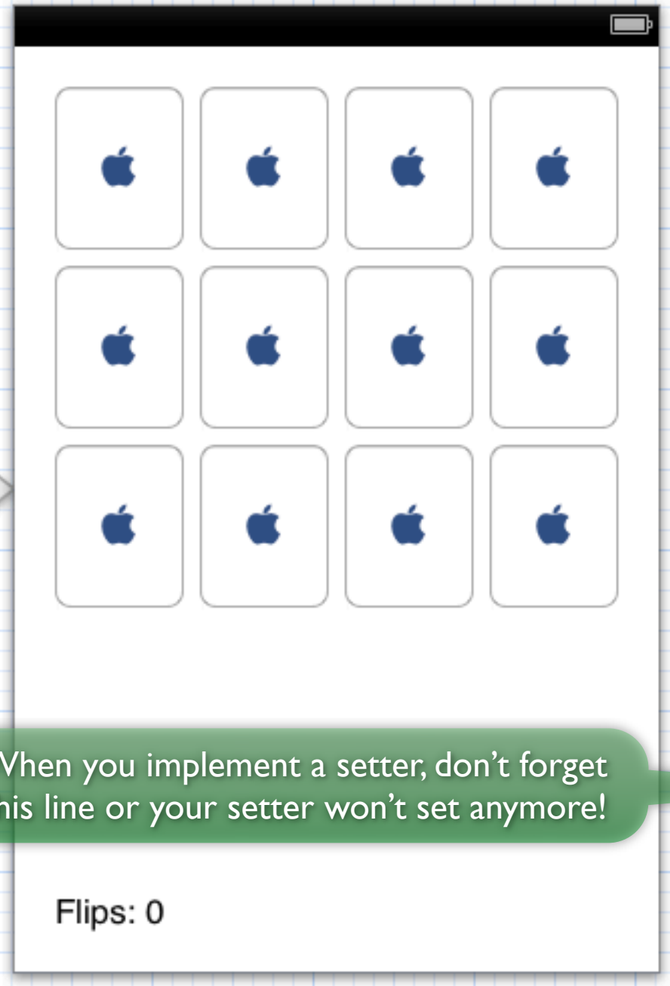
Let's fix that by using our Model

(that you hopefully have already brought in during the first assignment!).


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
-deck
Flips: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
We need to import PlayingCardDeck if we want to use it in this class.
Here's the @property for the deck you were supposed to add in Homework 0.
With the getter method for lazy instantiation.
Even though our deck @property is of class Deck, we are within our rights to set it to a PlayingCardDeck instance since PlayingCardDeck inherits from Deck (and thus "isa" Deck).
Stanford CS193p
Winter 2013

Run Stop Matchismo › iPhone 6.0 Simulator Scheme Breakpoints Finished running Matchismo on iPhone 6.0 Simulato No Issues Editor View Organizer

Card Game View... › View



Automatic › CardGameViewController.m › -setCardButtons: 2

```
//
//  CardGameViewController.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//

#import "CardGameViewController.h"
#import "PlayingCardDeck.h"

@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@end

@implementation CardGameViewController

- (Deck *)deck
{
    if (!_deck) _deck = [[PlayingCardDeck alloc] init];
    return _deck;
}

- (void)setCardButtons:(NSArray *)cardButtons
{
    _cardButtons = cardButtons;
    for (UIButton *cardButton in cardButtons) {
        Card *card = [self.deck drawRandomCard];
        [cardButton setTitle:card.contents forState:UIControlStateSelected];
    }
}

- (void)setFlipCount:(int)flipCount
{
    _flipCount = flipCount;
    self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}

- (IBAction)flipCard:(UIButton *)sender
{
```

Next we'll use a `for-in` loop to set the title for the Selected state for each `UIButton` in our `cardButtons` outlet collection.

All we are doing here is implementing the setter for the `cardButtons` `@property`. Even though this `@property` is set by iOS (it's set to the array of buttons), it is still just a `@property`, and we have every right to implement its setter if we want (and this is a convenient place to set our buttons up).

When you implement a setter, don't forget this line or your setter won't set anymore!

We saw this method when we looked at the documentation, remember?

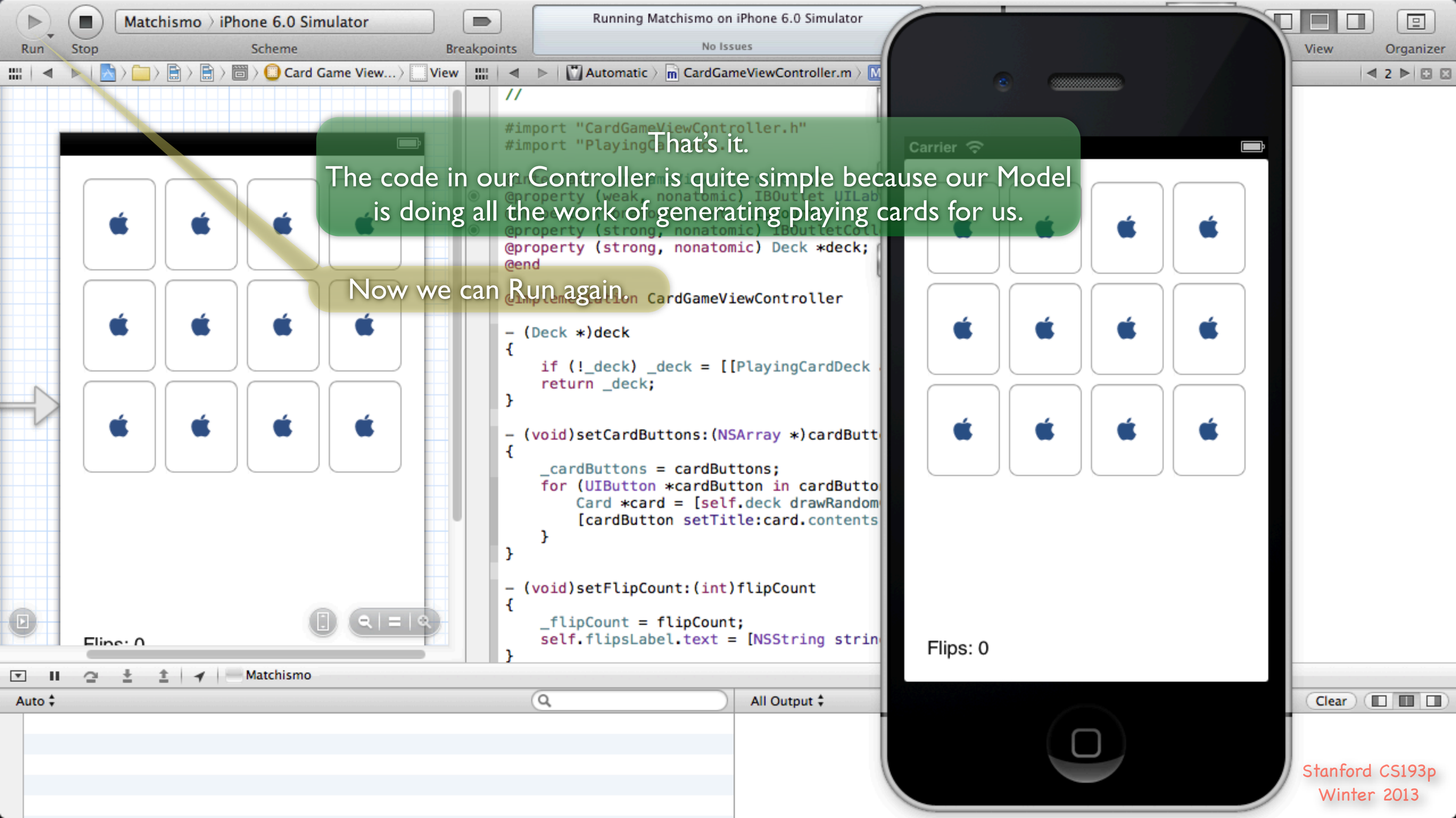
Deck's `drawRandomCard` method.

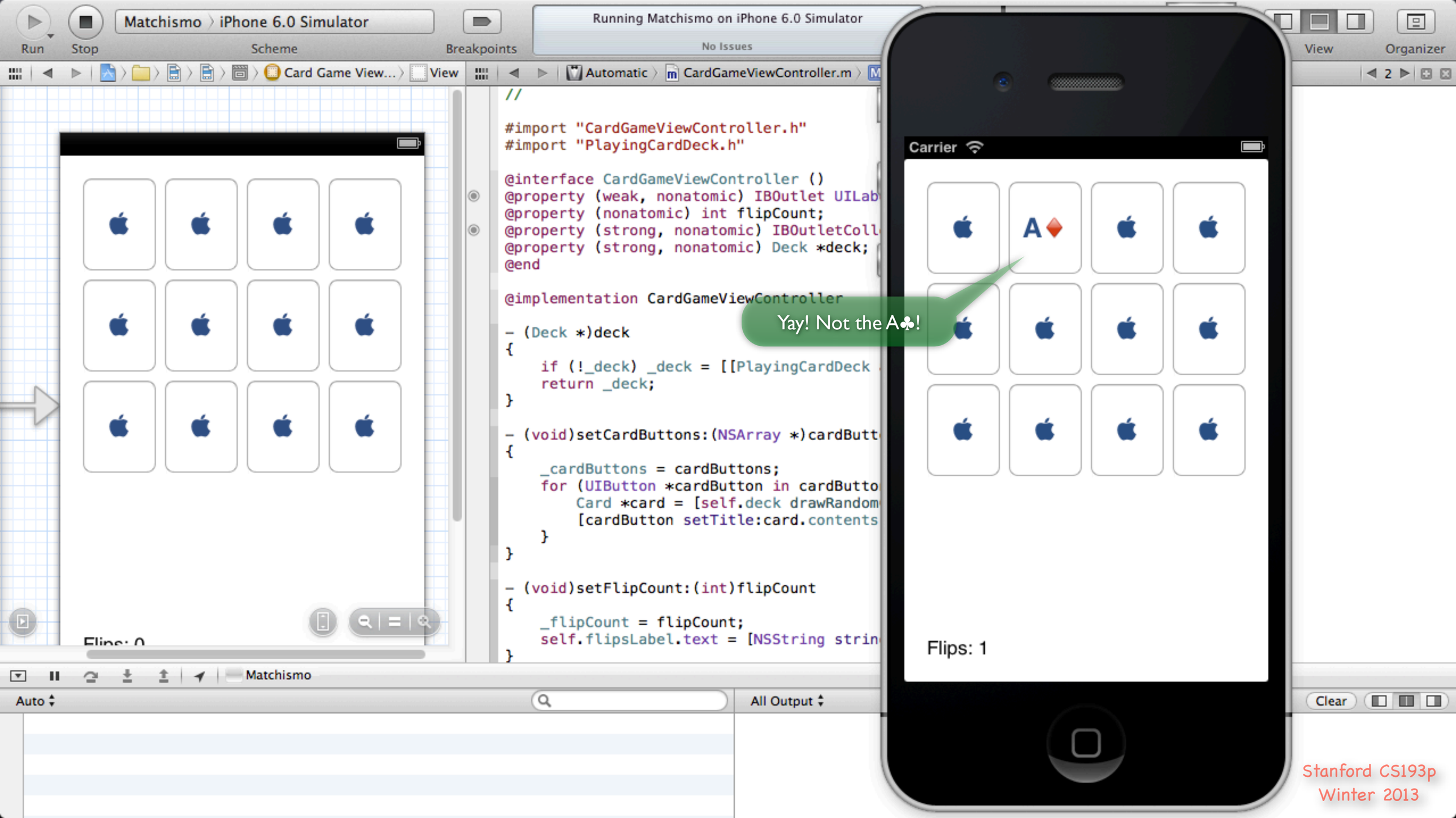
Stanford CS193p
Winter 2013

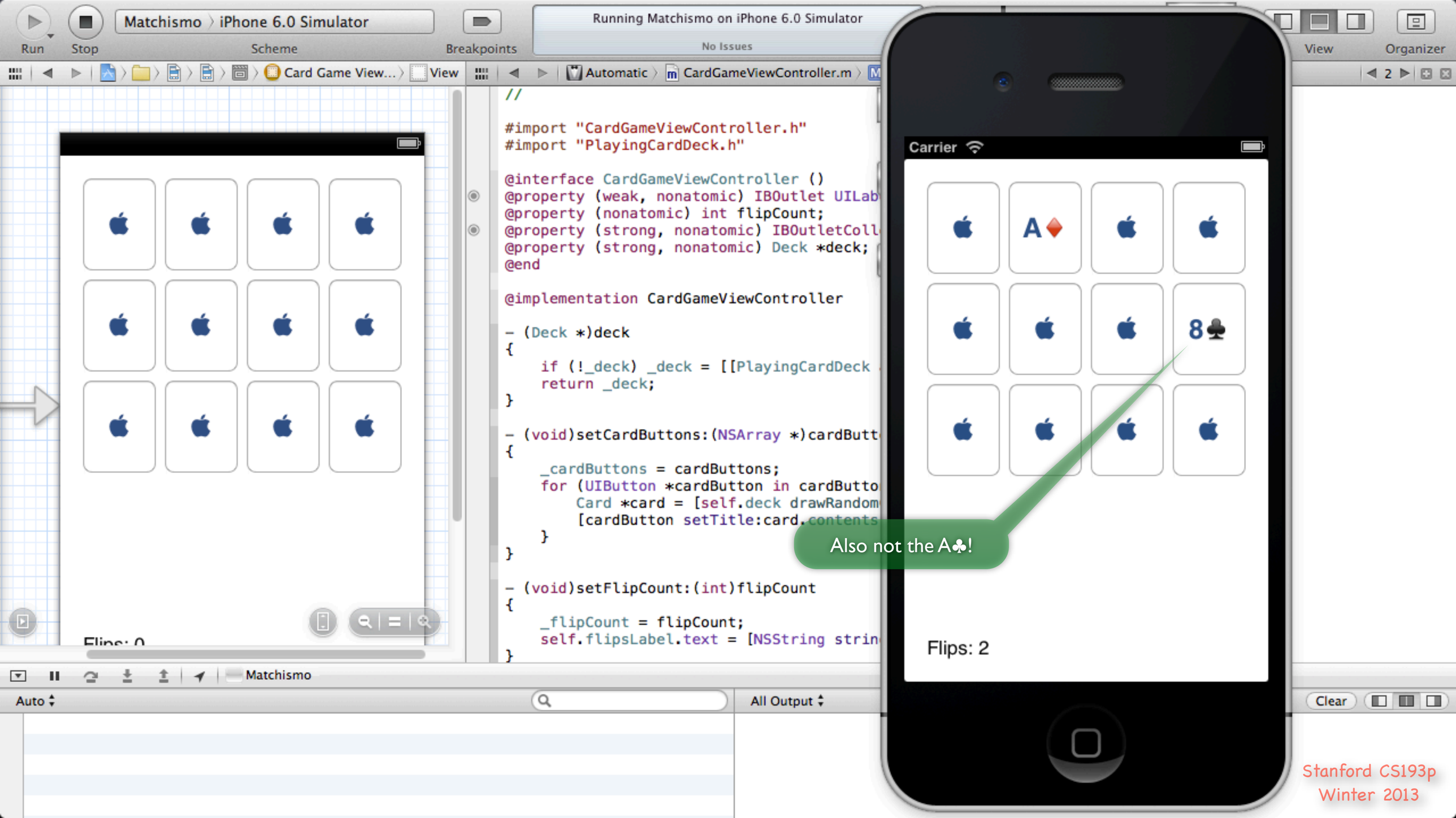
That's it.

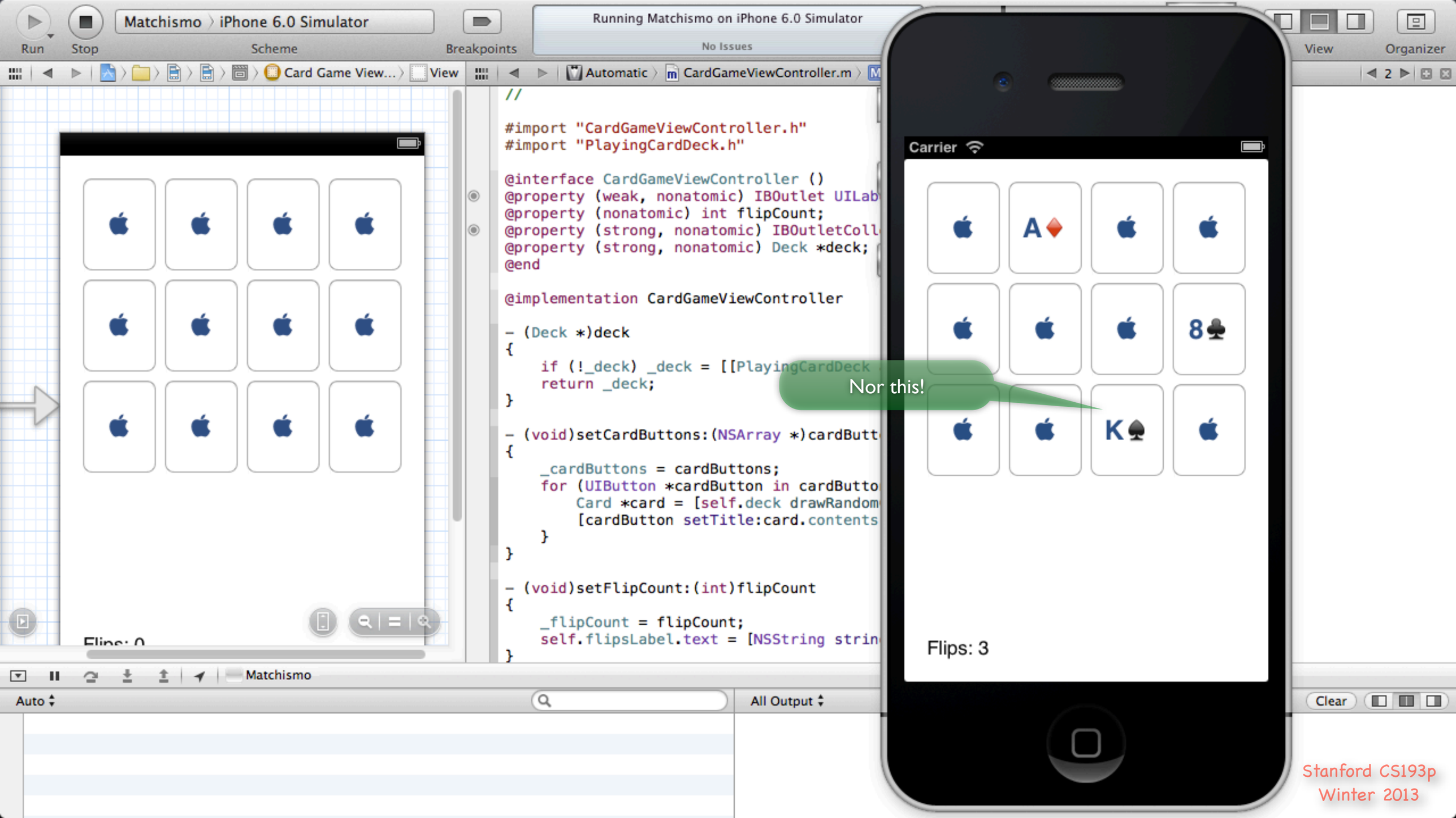
The code in our Controller is quite simple because our Model is doing all the work of generating playing cards for us.

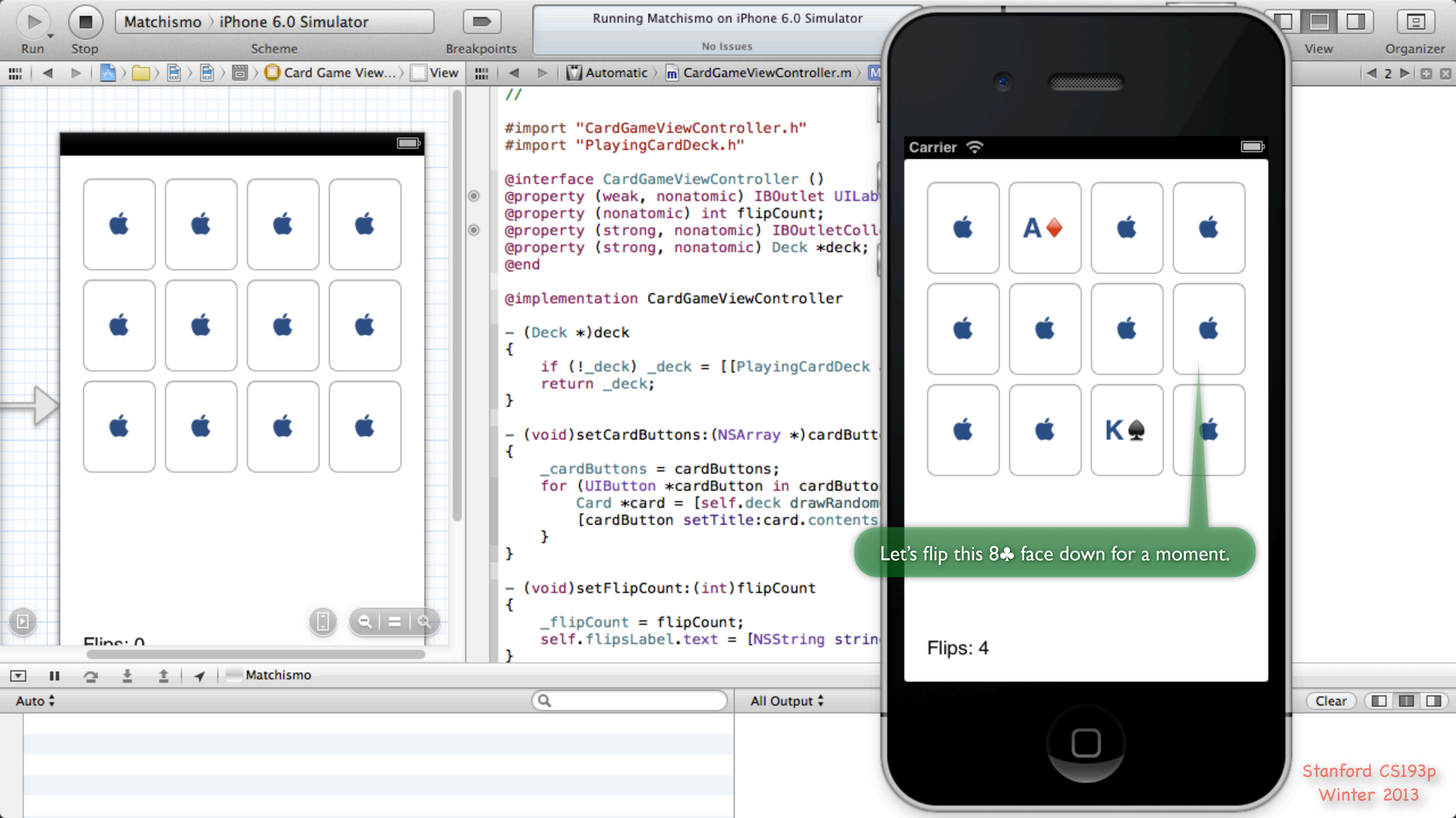
Now we can Run again.

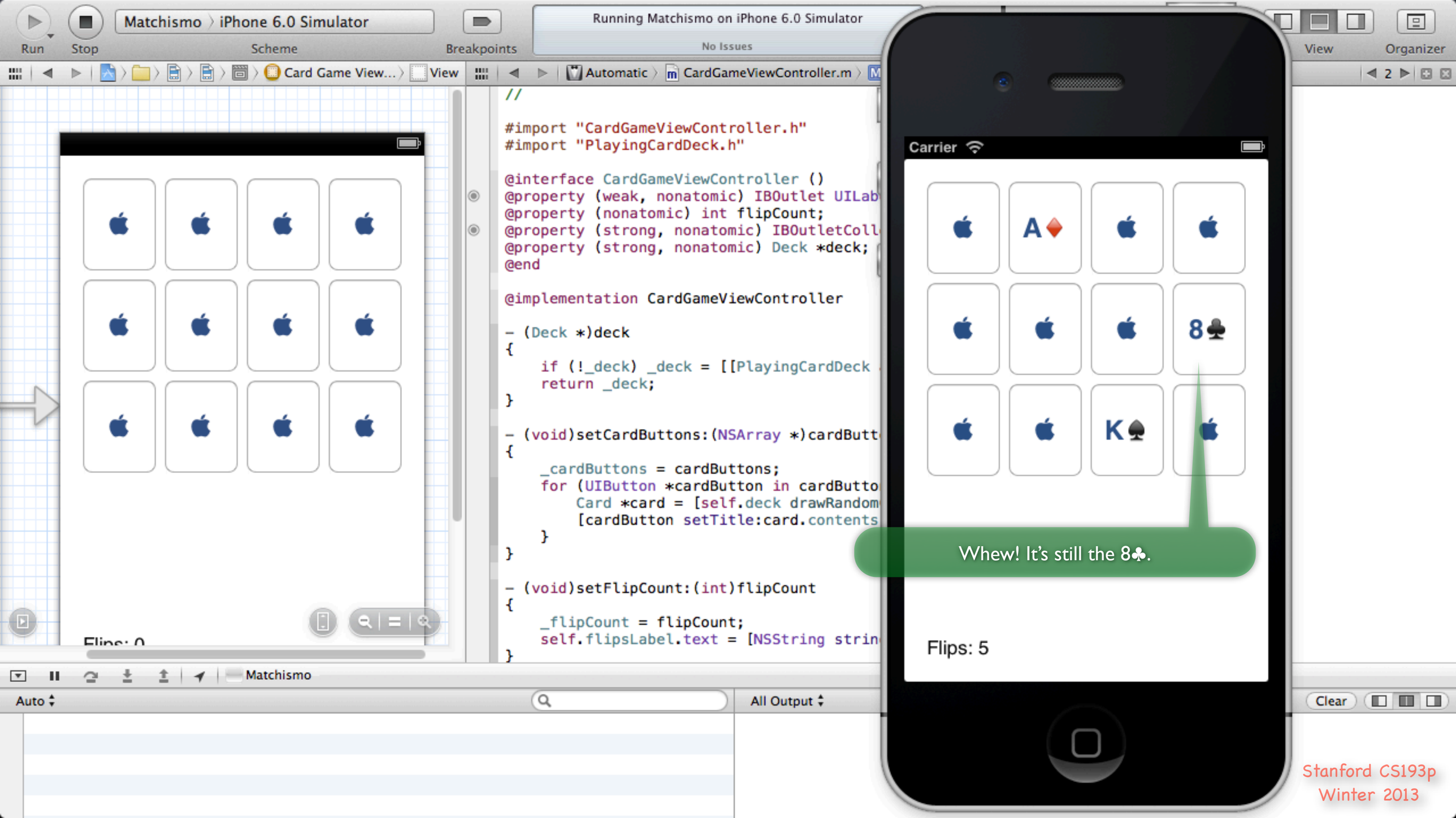


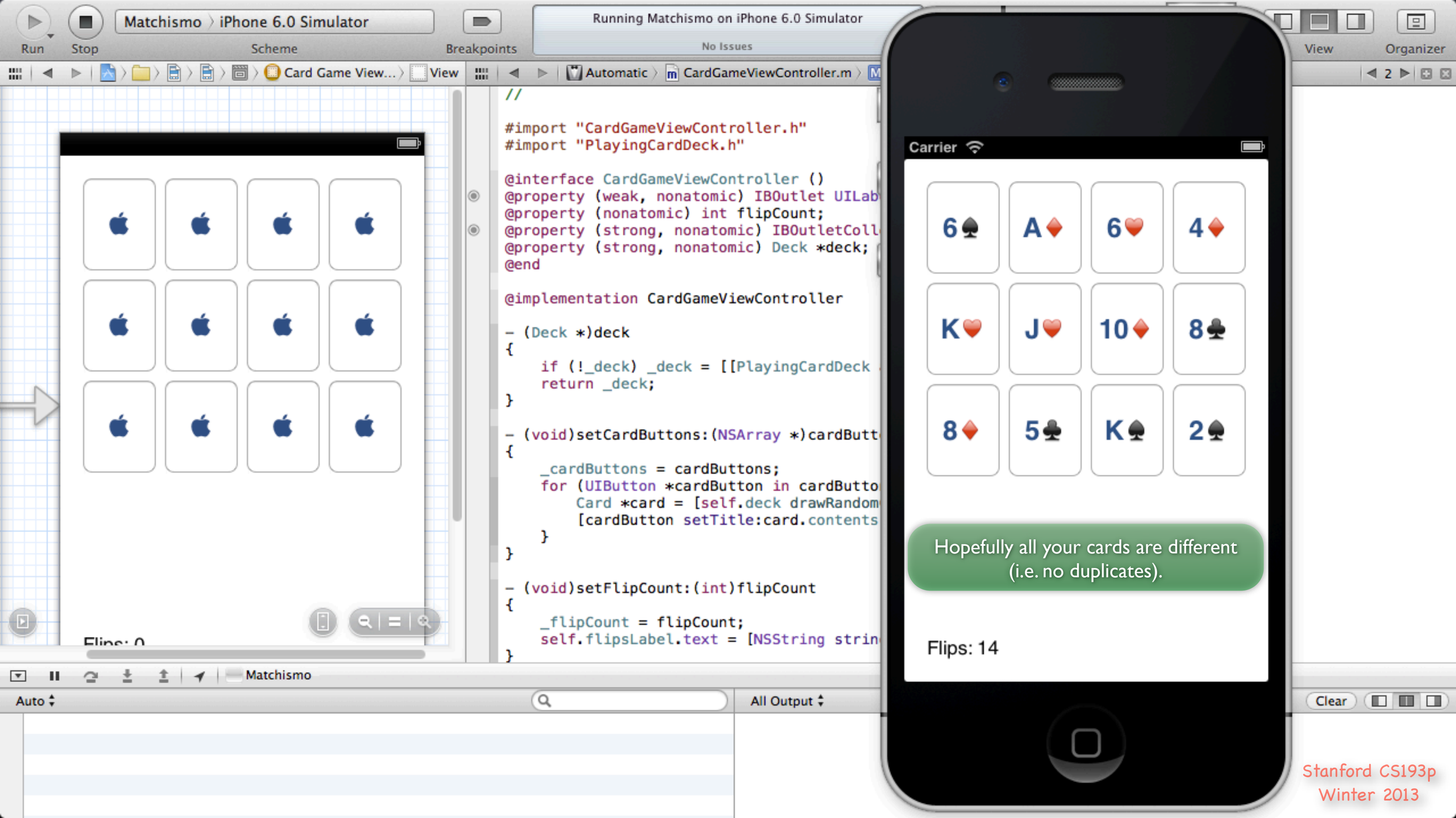

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButto
Card *card = [self.deck drawRandom
[cardButton setTitle:card.contents
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
Flips: 0
Carrier
Yay! Not the A♣!
Flips: 1
Matchismo
Auto
All Output
Clear
Stanford CS193p
Winter 2013

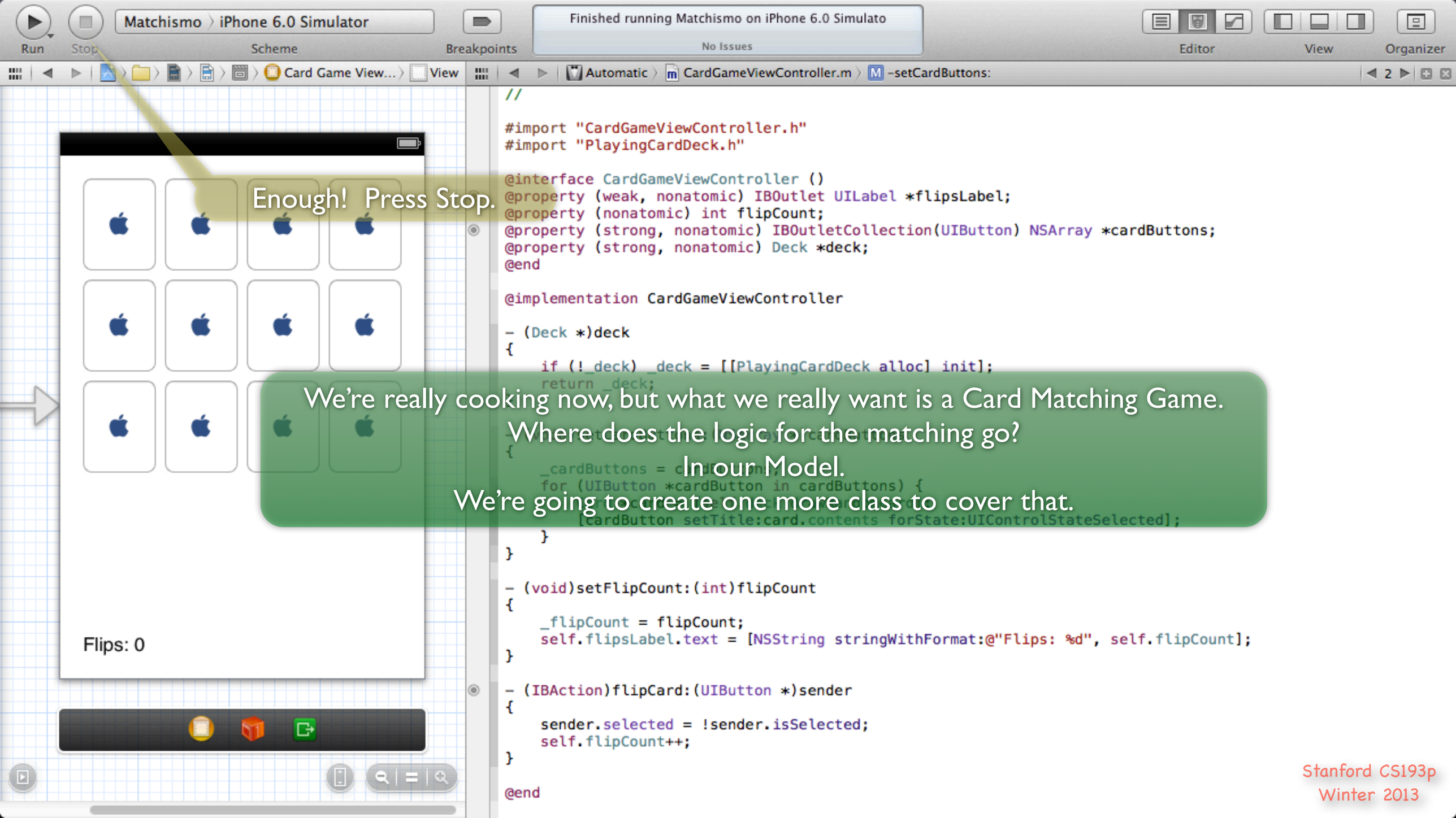

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButto
Card *card = [self.deck drawRandom
[cardButton setTitle:card.contents
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♦
8♣
Also not the A♣!
Flips: 2
Stanford CS193p
Winter 2013

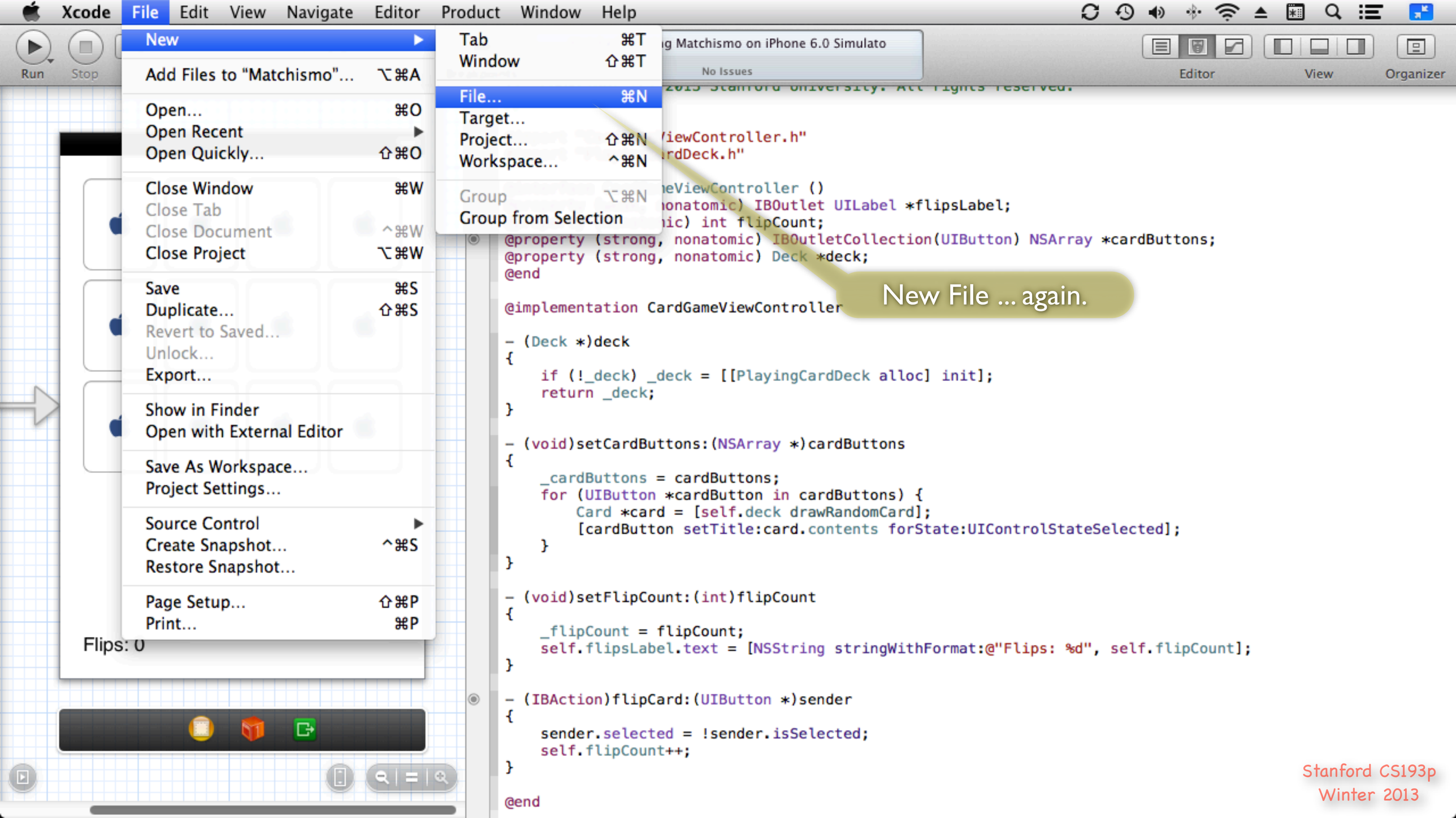

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
2
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButto
Card *card = [self.deck drawRandom
[cardButton setTitle:card.contents
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♦
8♣
K♠
Nor this!
Flips: 3
Stanford CS193p
Winter 2013

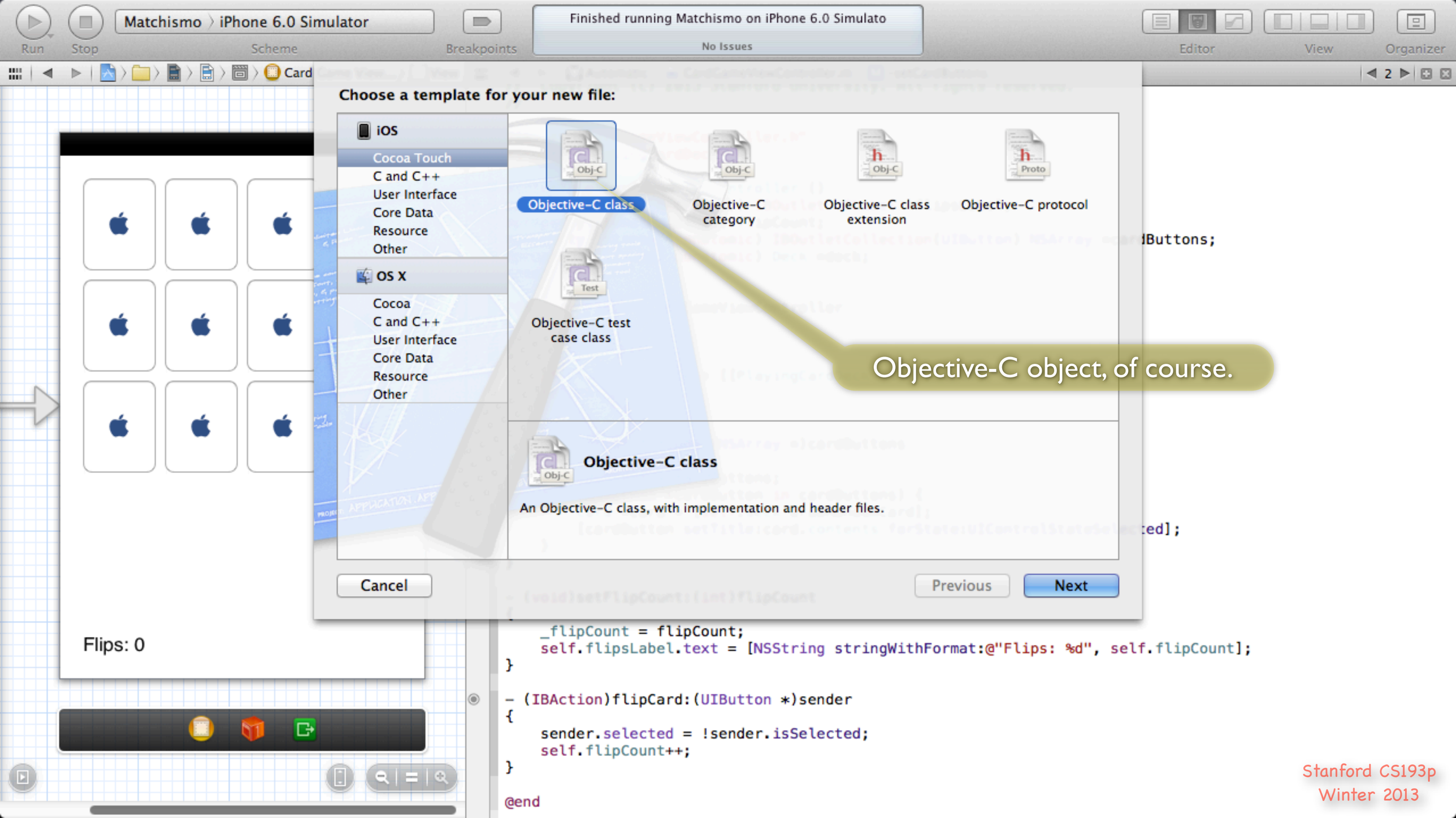

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButto
Card *card = [self.deck drawRandom
[cardButton setTitle:card.contents
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♦
K♠
Let's flip this 8♣ face down for a moment.
Flips: 4
Stanford CS193p
Winter 2013

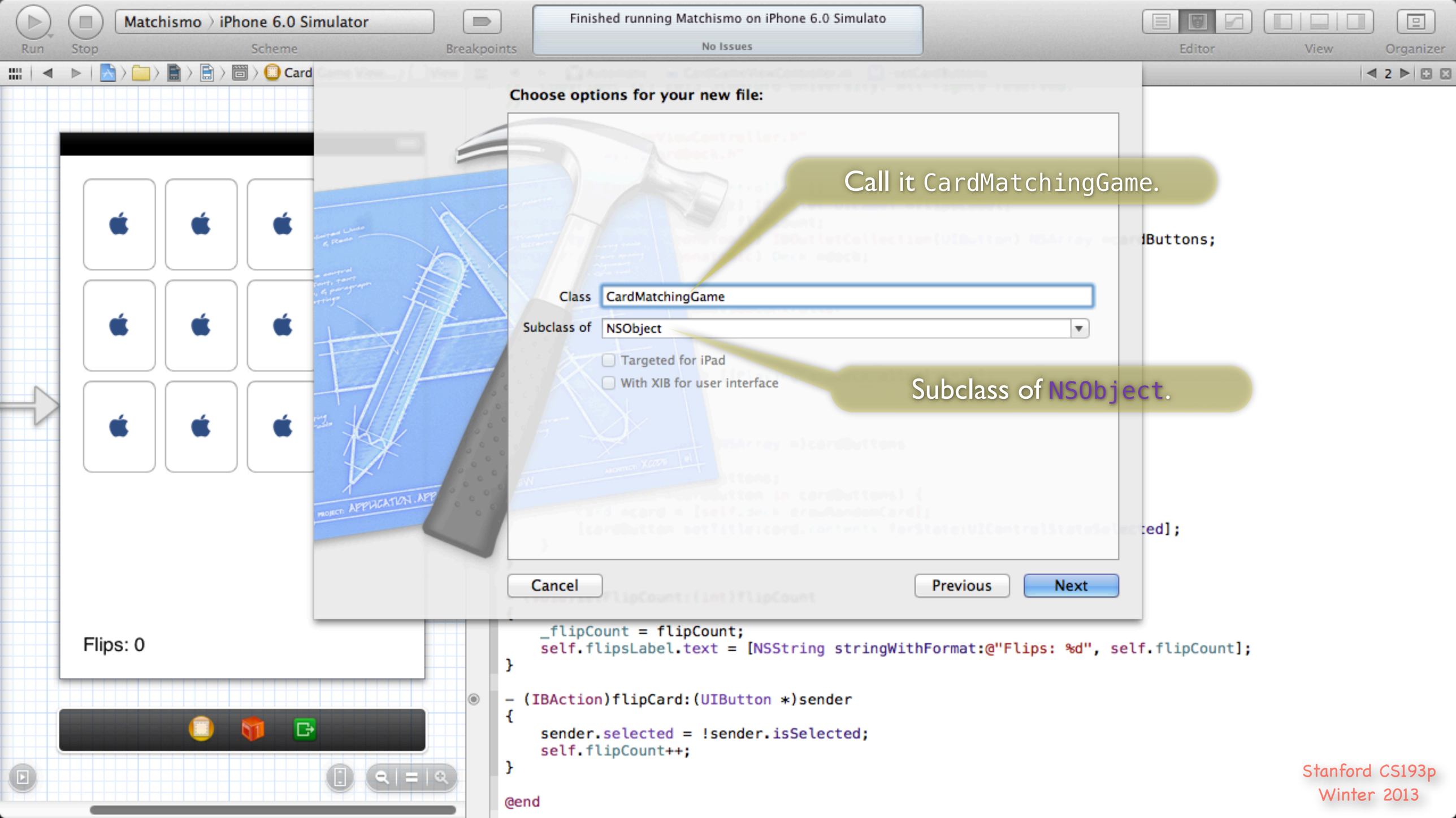

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButto
Card *card = [self.deck drawRandom
[cardButton setTitle:card.contents
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
A♦
8♣
K♠
Whew! It's still the 8♣.
Flips: 5
Stanford CS193p
Winter 2013

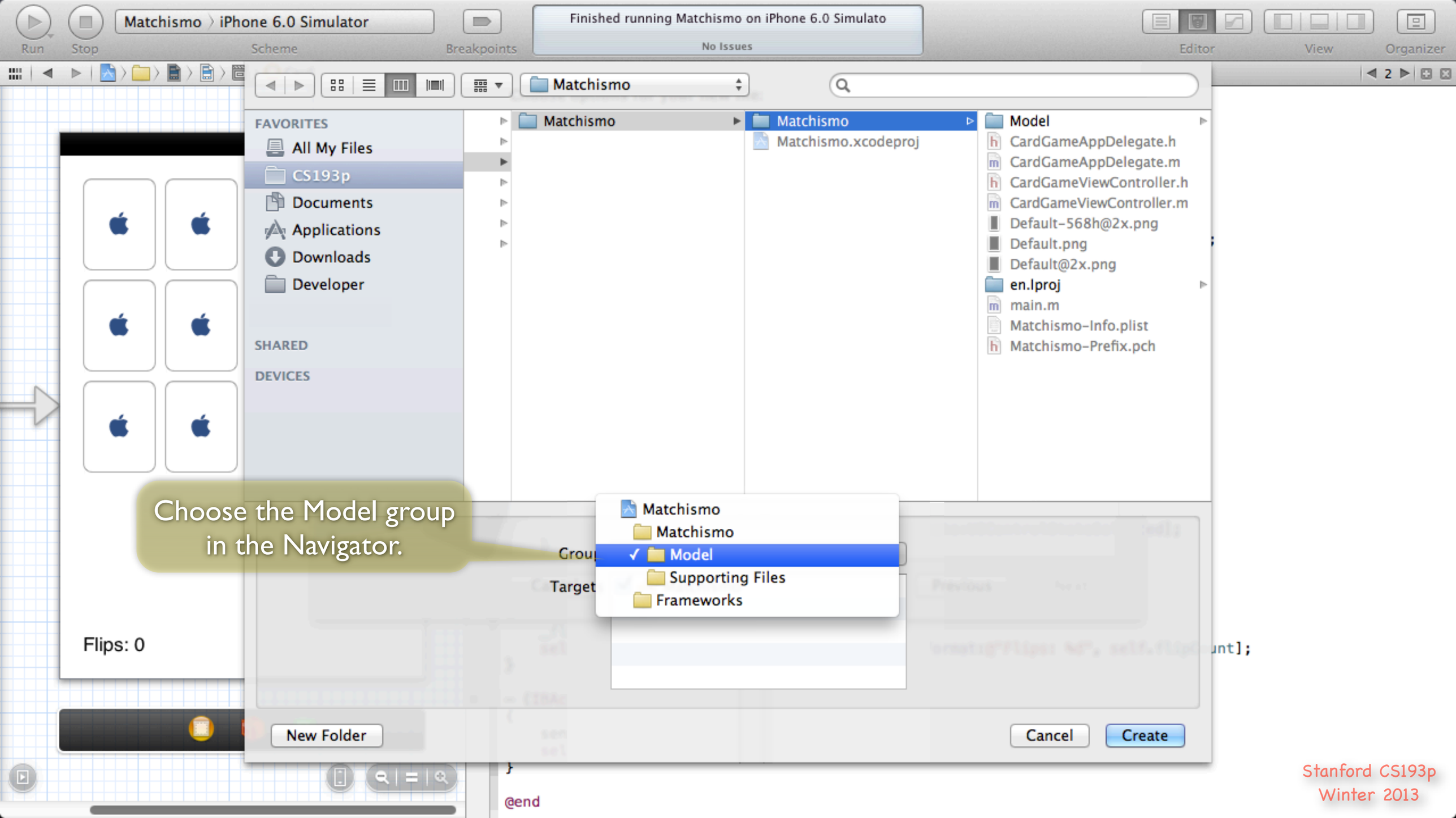

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButto
Card *card = [self.deck drawRandom
[cardButton setTitle:card.contents
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString strin
}
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
6♠
A♦
6♥
4♦
K♥
J♥
10♦
8♣
8♦
5♣
K♠
2♠
Hopefully all your cards are different
(i.e. no duplicates).
Flips: 14
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 -setCardButtons:
Enough! Press Stop.
We're really cooking now, but what we really want is a Card Matching Game.
Where does the logic for the matching go?
In our Model.
We're going to create one more class to cover that.
Flips: 0
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
for (UIButton *cardButton in cardButtons) {
[cardButton setTitle:card.contents forState:UIControlStateSelected];
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

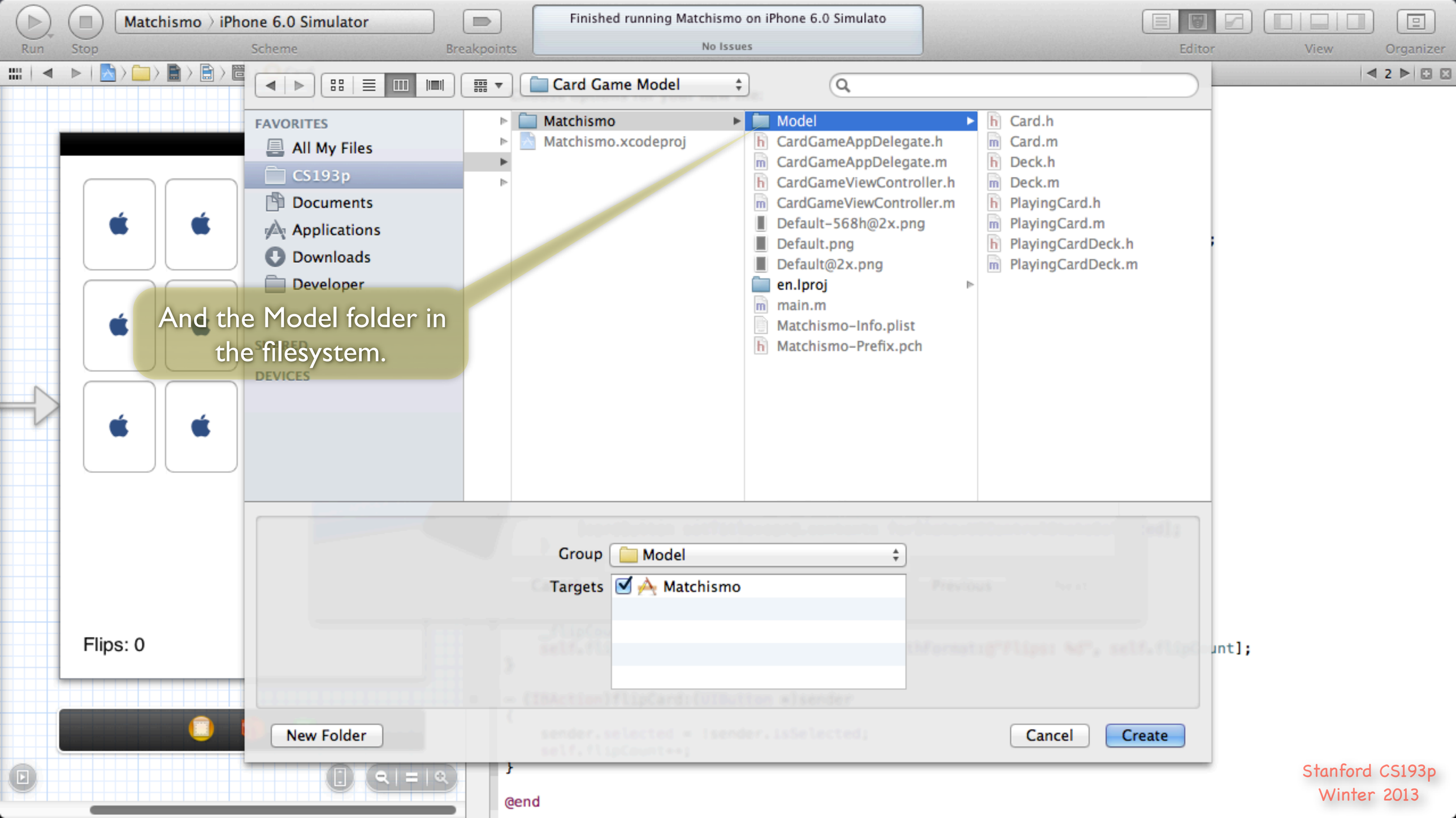

Xcode File Edit View Navigate Editor Product Window Help
New
Add Files to "Matchismo"... ⌥⌘A
Open... ⌘O
Open Recent
Open Quickly... ⇧⌘O
Close Window ⌘W
Close Tab
Close Document ^⌘W
Close Project ⌥⌘W
Save ⌘S
Duplicate... ⇧⌘S
Revert to Saved...
Unlock...
Export...
Show in Finder
Open with External Editor
Save As Workspace...
Project Settings...
Source Control
Create Snapshot... ^⌘S
Restore Snapshot...
Page Setup... ⇧⌘P
Print... ⌘P
Tab ⌘T
Window ⇧⌘T
File... ⌘N
Target...
Project... ⇧⌘N
Workspace... ^⌘N
Group ⌥⌘N
Group from Selection
Run
Stop
Editor
View
Organizer
Flips: 0
@property (strong, nonatomic) IBOutlet UILabel *flipsLabel;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButtons) {
Card *card = [self.deck drawRandomCard];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
}
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
New File ... again.
Stanford CS193p
Winter 2013

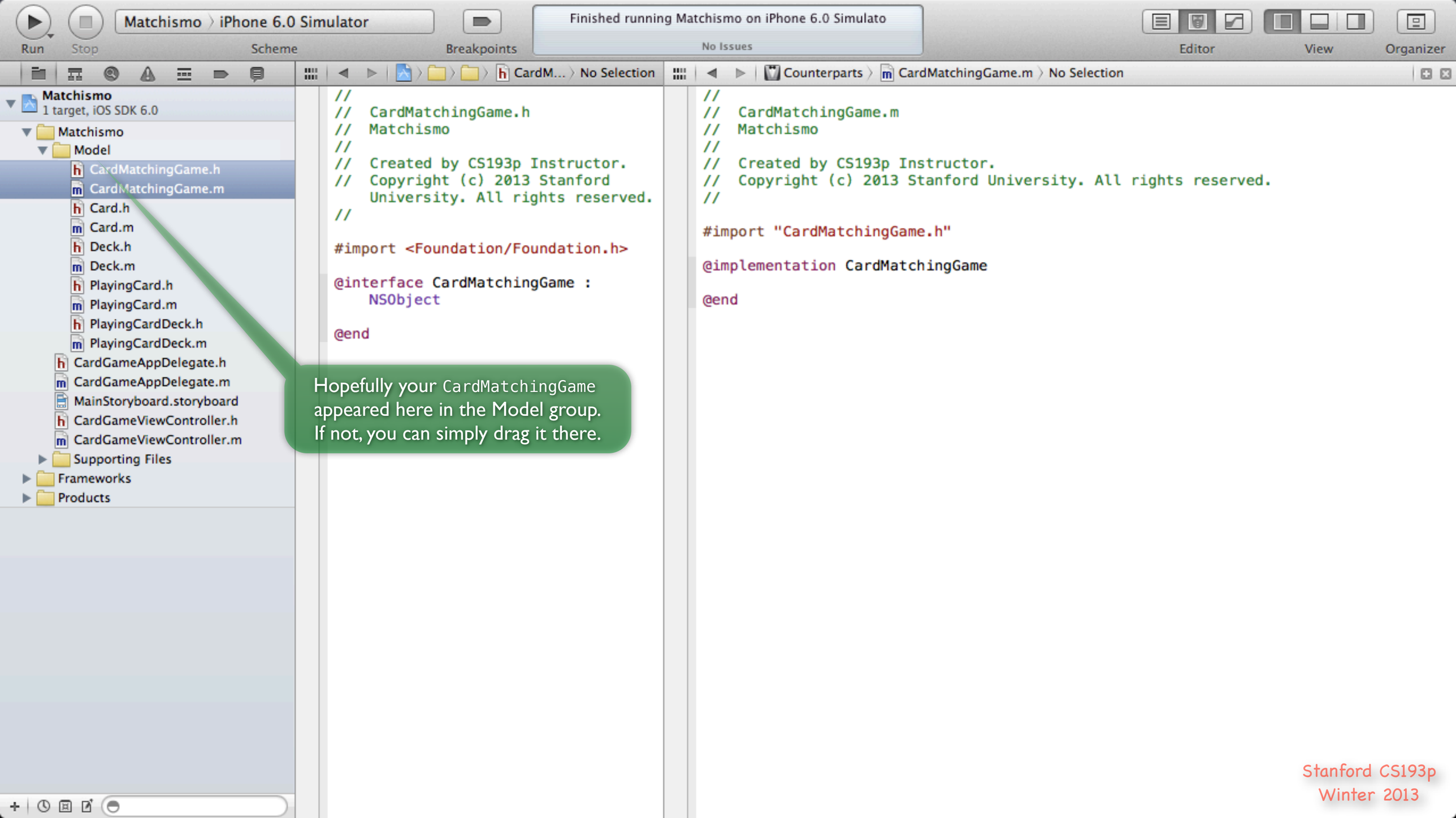

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Choose a template for your new file:
iOS
Cocoa Touch
C and C++
User Interface
Core Data
Resource
Other
OS X
Cocoa
C and C++
User Interface
Core Data
Resource
Other
Objective-C class
Objective-C category
Objective-C class extension
Objective-C protocol
Objective-C test case class
Objective-C class
An Objective-C class, with implementation and header files.
Cancel
Previous
Next
Objective-C object, of course.
Flips: 0
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

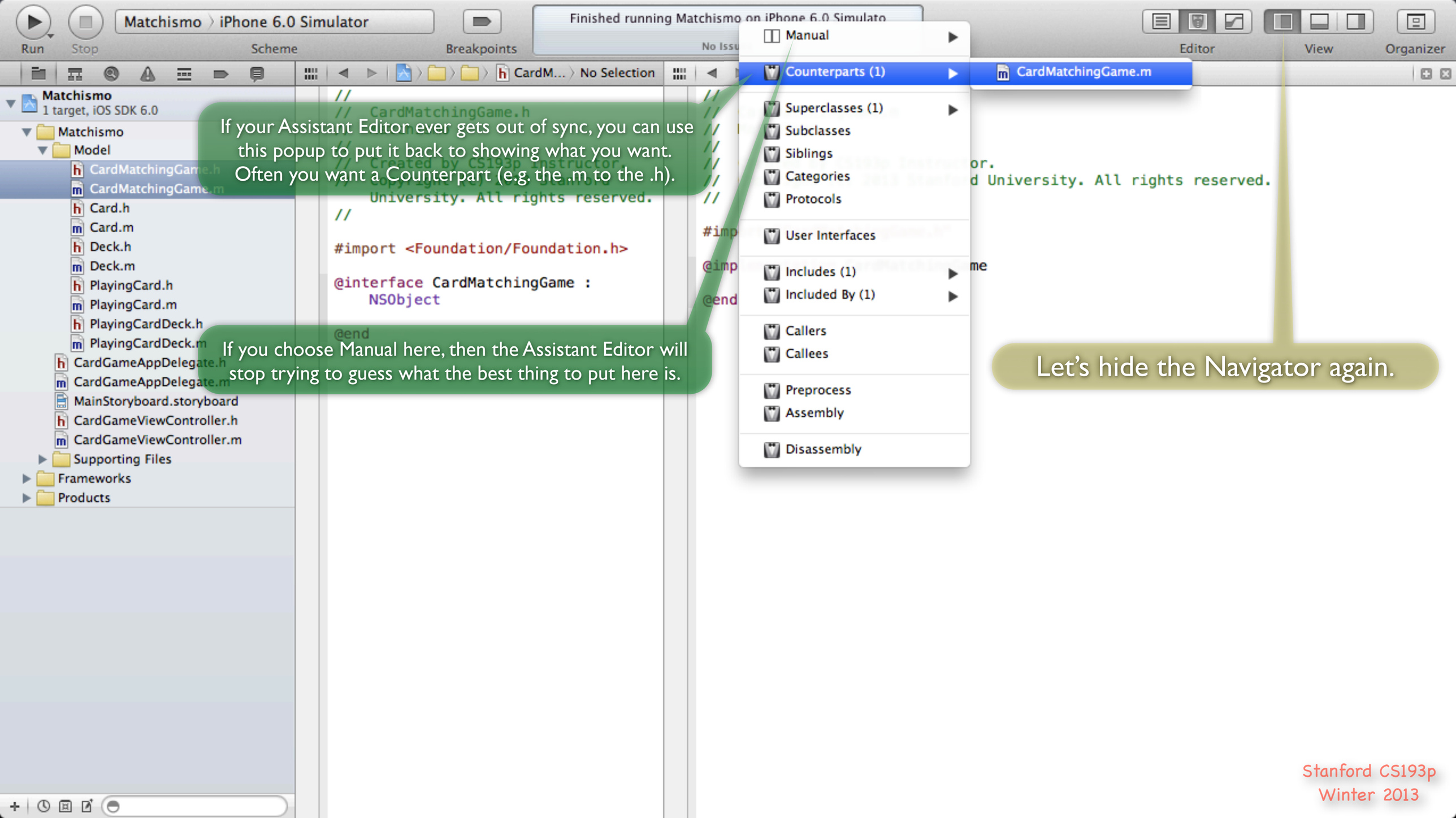

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Choose options for your new file:
Call it CardMatchingGame.
Class
CardMatchingGame
Subclass of
NSObject
Targeted for iPad
With XIB for user interface
Subclass of NSObject.
Cancel
Previous
Next
Flips: 0
_flipCount = flipCount;
self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
@end
Stanford CS193p
Winter 2013

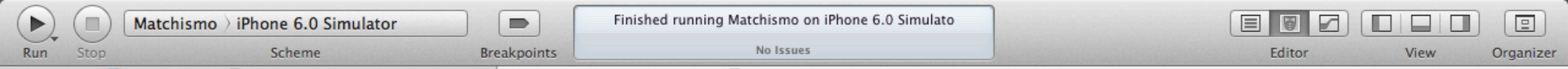


Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game Model
FAVORITES
All My Files
CS193p
Documents
Applications
Downloads
Developer
DEVICES
Matchismo
Matchismo.xcodeproj
Model
CardGameAppDelegate.h
CardGameAppDelegate.m
CardGameViewController.h
CardGameViewController.m
Default-568h@2x.png
Default.png
Default@2x.png
en.lproj
main.m
Matchismo-Info.plist
Matchismo-Prefix.pch
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
And the Model folder in the filesystem.
Group
Model
Targets
Matchismo
New Folder
Cancel
Create
Flips: 0
@end
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
CardM... 〉 No Selection
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford
University. All rights reserved.
//
#import <Foundation/Foundation.h>
@interface CardMatchingGame :
NSObject
@end
Counterparts 〉 CardMatchingGame.m 〉 No Selection
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@implementation CardMatchingGame
@end
Hopefully your CardMatchingGame appeared here in the Model group. If not, you can simply drag it there.


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
CardM... 〉 No Selection
//
// CardMatchingGame.h
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford
University. All rights reserved.
//
#import <Foundation/Foundation.h>
@interface CardMatchingGame :
NSObject
@end
Manual
Counterparts (1)
Superclasses (1)
Subclasses
Siblings
Categories
Protocols
User Interfaces
Includes (1)
Included By (1)
Callers
Callees
Preprocess
Assembly
Disassembly
CardMatchingGame.m
If your Assistant Editor ever gets out of sync, you can use this popup to put it back to showing what you want. Often you want a Counterpart (e.g. the .m to the .h).
If you choose Manual here, then the Assistant Editor will stop trying to guess what the best thing to put here is.
Let's hide the Navigator again.
Stanford CS193p
Winter 2013



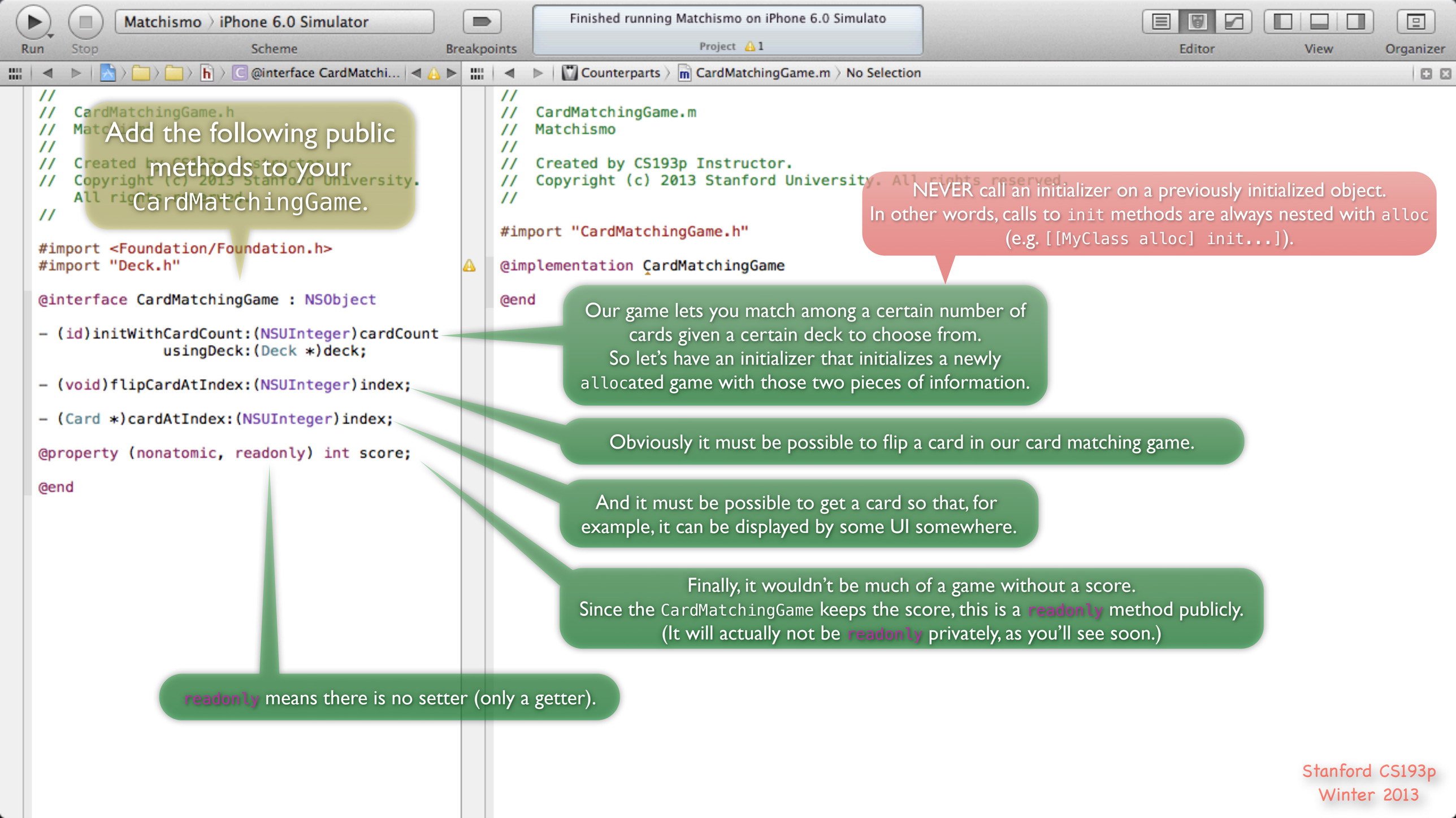
CardMatchingGame.h — No Selection

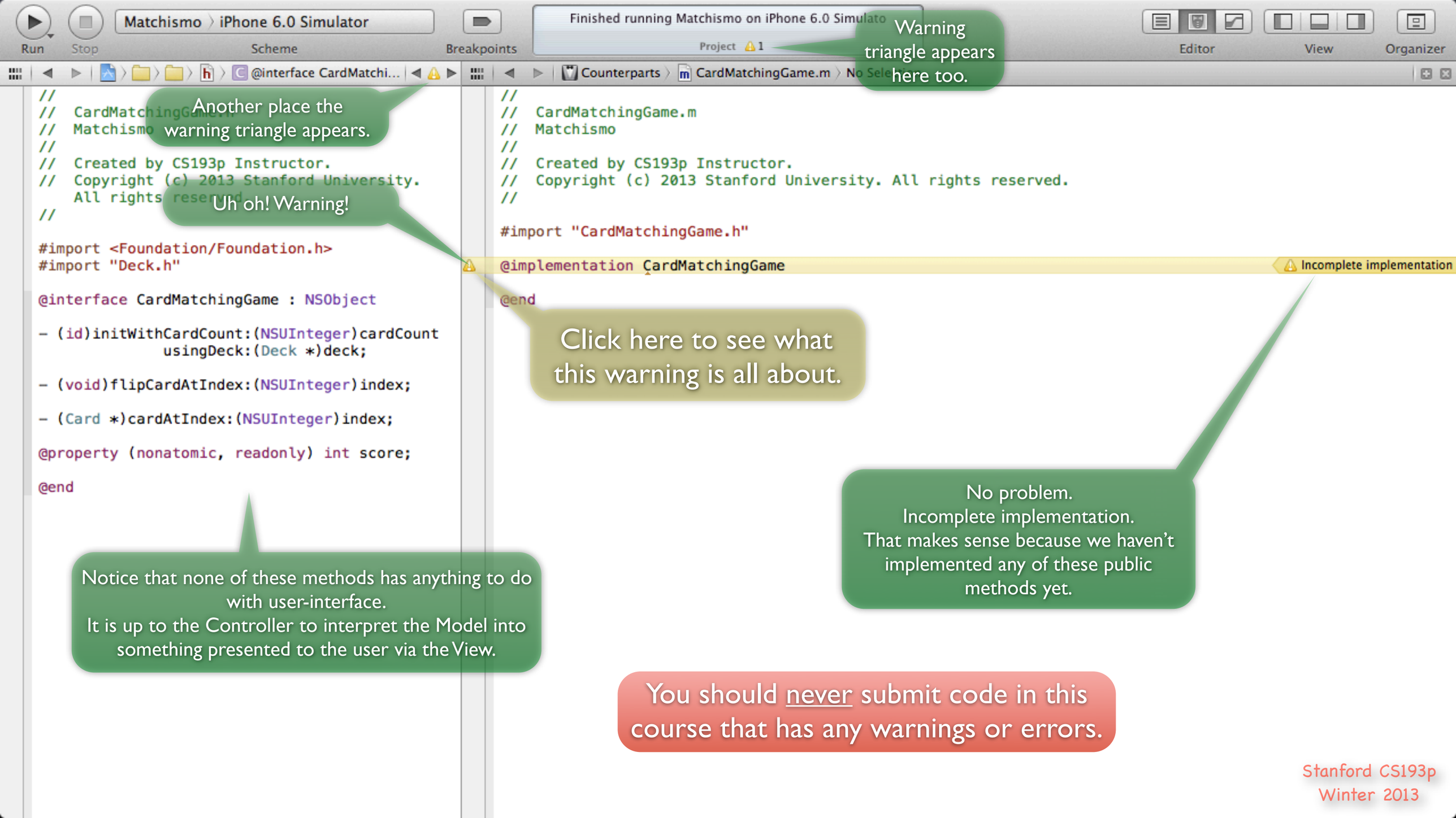
```
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
    All rights reserved.
//

#import <Foundation/Foundation.h>

@interface CardMatchingGame : NSObject

@end
```

CardMatchingGame.m — No Selection

```
//
//  CardMatchingGame.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University. All rights reserved.
//

#import "CardMatchingGame.h"

@implementation CardMatchingGame

@end
```

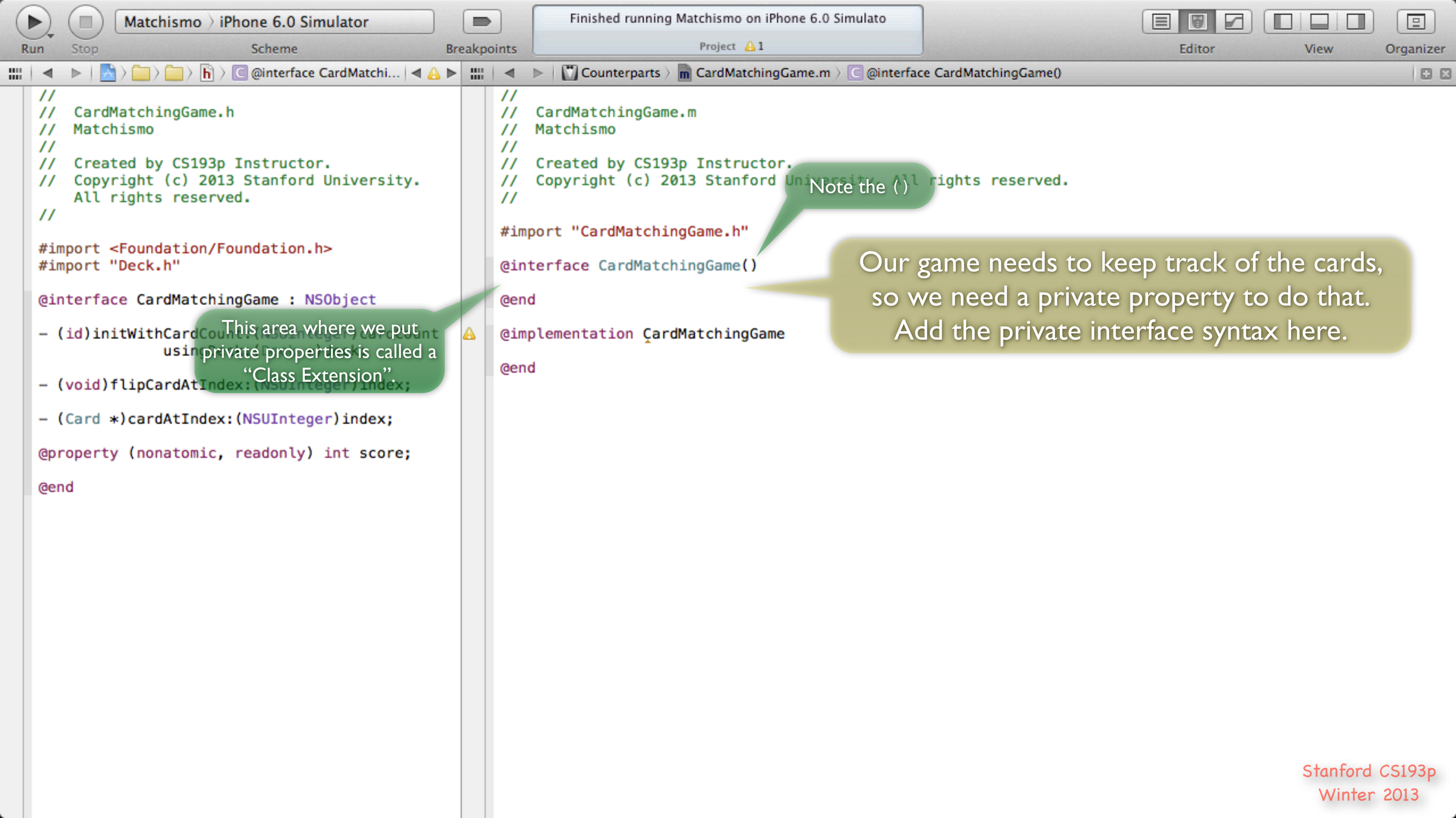
Often the best way to start designing a new class is to fill out its interface since that specifies what it is this class is responsible for doing.


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 No Selection
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@implementation CardMatchingGame
@end
Add the following public methods to your CardMatchingGame.
NEVER call an initializer on a previously initialized object.
In other words, calls to init methods are always nested with alloc
(e.g. [[MyClass alloc] init...]).
Our game lets you match among a certain number of cards given a certain deck to choose from.
So let's have an initializer that initializes a newly allocated game with those two pieces of information.
Obviously it must be possible to flip a card in our card matching game.
And it must be possible to get a card so that, for example, it can be displayed by some UI somewhere.
Finally, it wouldn't be much of a game without a score.
Since the CardMatchingGame keeps the score, this is a readonly method publicly.
(It will actually not be readonly privately, as you'll see soon.)
readonly means there is no setter (only a getter).
Stanford CS193p
Winter 2013




```
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//  All rights reserved.
//

#import <Foundation/Foundation.h>
#import "Deck.h"

@interface CardMatchingGame : NSObject

- (id)initWithCardCount:(NSUInteger)cardCount
              usingDeck:(Deck *)deck;

- (void)flipCardAtIndex:(NSUInteger)index;

- (Card *)cardAtIndex:(NSUInteger)index;

@property (nonatomic, readonly) int score;

@end
```

```
//
//  CardMatchingGame.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University. All rights reserved.
//

#import "CardMatchingGame.h"

@implementation CardMatchingGame

@end
```


Matchismo 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 @interface CardMatchingGame()
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
This area where we put private properties is called a "Class Extension".
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@end
@implementation CardMatchingGame
@end
Note the ()
Our game needs to keep track of the cards, so we need a private property to do that. Add the private interface syntax here.
Stanford CS193p
Winter 2013

```
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//  All rights reserved.
//

#import <Foundation/Foundation.h>
#import "Deck.h"

@interface CardMatchingGame : NSObject

- (id)initWithCardCount:(NSUInteger)cardCount
              usingDeck:(Deck *)deck;

- (void)flipCardAtIndex:(NSUInteger)index;

- (Card *)cardAtIndex:(NSUInteger)index;

@property (nonatomic, readonly) int score;

@end
```



```
//
//  CardMatchingGame.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University. All rights reserved.
//

#import "CardMatchingGame.h"

@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@end

@implementation CardMatchingGame

@end
```

Add a private `@property` which is a mutable `NSArray`. The objects in this array will be of class `Card`.

Indeed there is no way to express in Objective-C that this array should only have `Card` objects in it. One might argue that that is a shortcoming.

Matchismo › iPhone 6.0 Simulator

Run Stop Scheme Breakpoints

Finished running Matchismo on iPhone 6.0 Simulato

Project ⚠1

Editor View Organizer

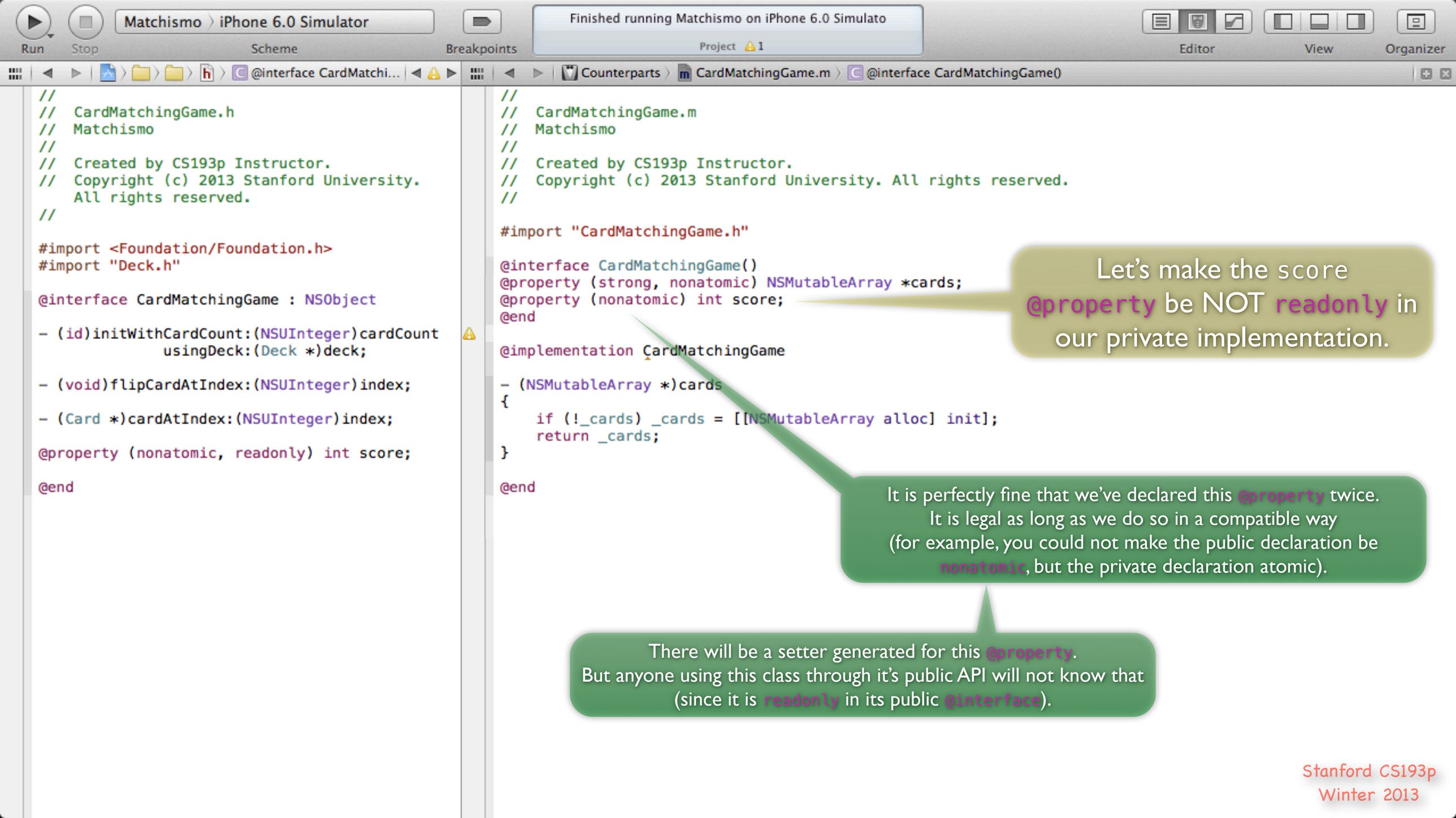
@interface CardMatchi...

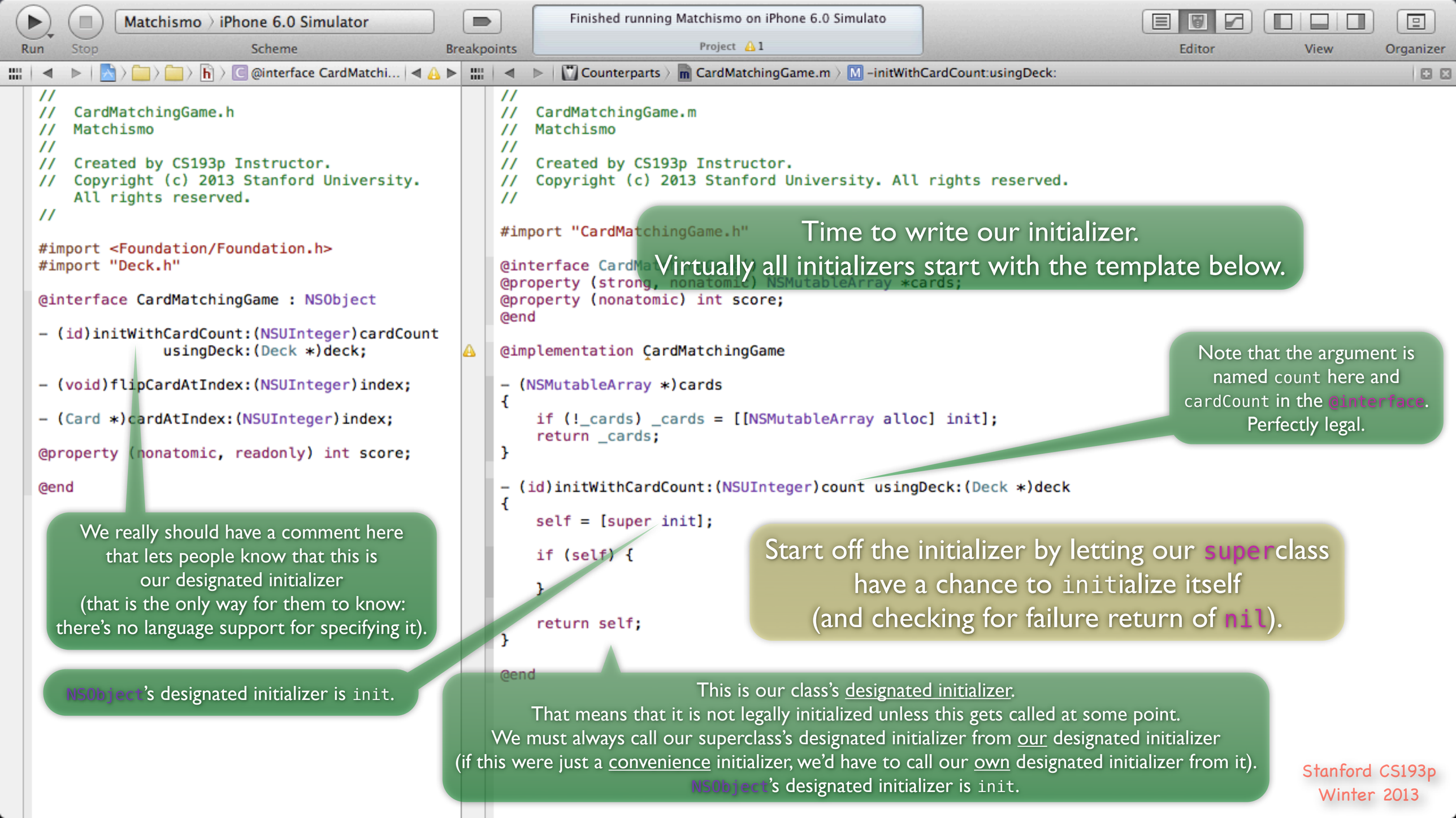
```objc
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//  All rights reserved.
//

#import <Foundation/Foundation.h>
#import "Deck.h"

@interface CardMatchingGame : NSObject

- (id)initWithCardCount:(NSUInteger)cardCount
              usingDeck:(Deck *)deck;

- (void)flipCardAtIndex:(NSUInteger)index;

- (Card *)cardAtIndex:(NSUInteger)index;

@property (nonatomic, readonly) int score;

@end
```

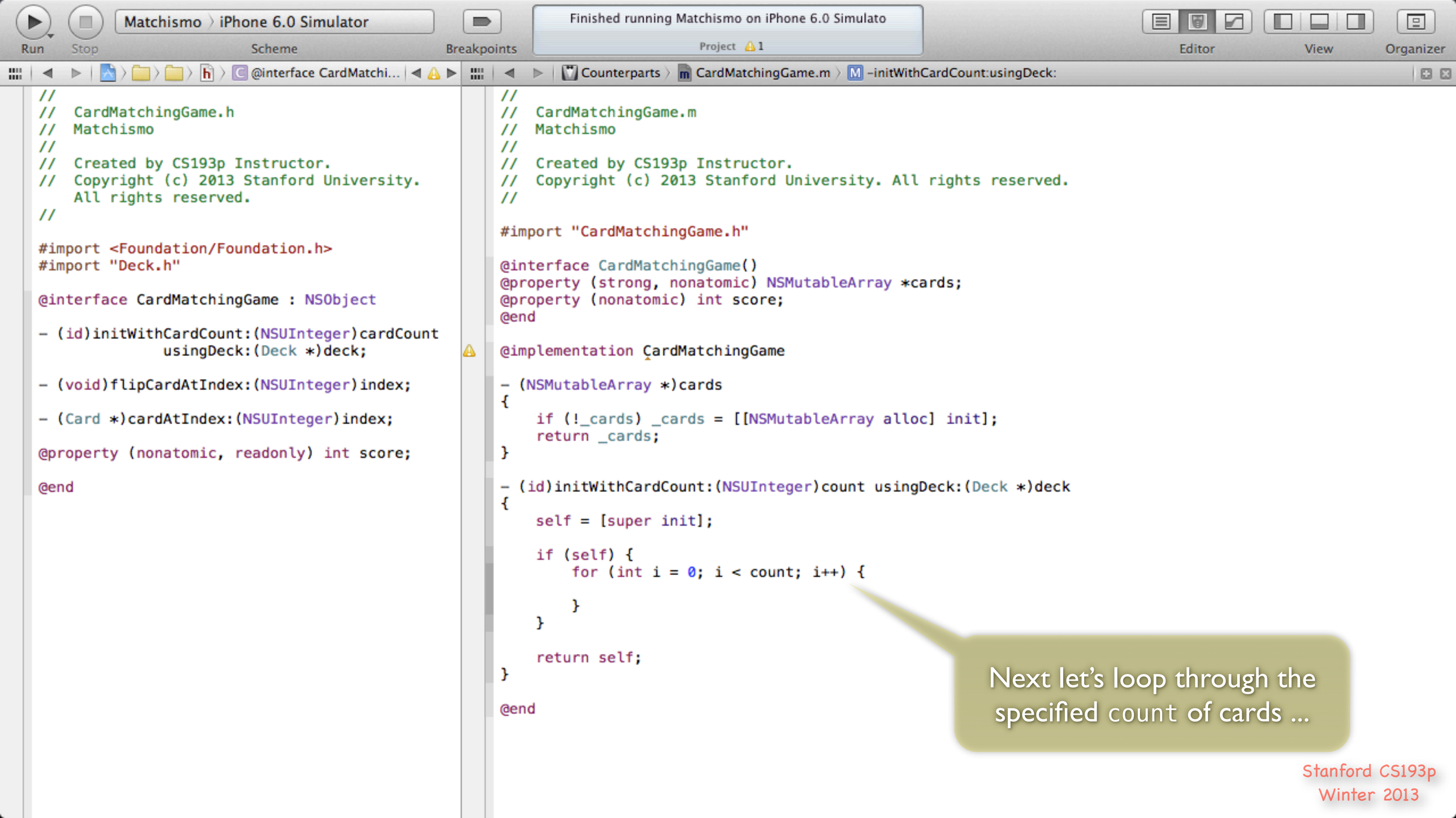
Counterparts › CardMatchingGame.m › @interface CardMatchingGame()

```objc
//
//  CardMatchingGame.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University. All rights reserved.
//

#import "CardMatchingGame.h"

@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@end

@implementation CardMatchingGame

- (NSMutableArray *)cards
{
    if (!_cards) _cards = [[NSMutableArray alloc] init];
    return _cards;
}

@end
```

Lazy instantiation!
Hopefully this is quite familiar to you by now.


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 @interface CardMatchingGame()
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{
if (!_cards) _cards = [[NSMutableArray alloc] init];
return _cards;
}
@end
Let's make the score @property be NOT readonly in our private implementation.
It is perfectly fine that we've declared this @property twice.
It is legal as long as we do so in a compatible way
(for example, you could not make the public declaration be nonatomic, but the private declaration atomic).
There will be a setter generated for this @property.
But anyone using this class through it's public API will not know that
(since it is readonly in its public @interface).
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 -initWithCardCount:usingDeck:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{
if (!_cards) _cards = [[NSMutableArray alloc] init];
return _cards;
}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{
self = [super init];
if (self) {
}
return self;
}
@end
Time to write our initializer.
Virtually all initializers start with the template below.
Note that the argument is named count here and cardCount in the @interface. Perfectly legal.
We really should have a comment here that lets people know that this is our designated initializer (that is the only way for them to know: there's no language support for specifying it).
Start off the initializer by letting our superclass have a chance to initialize itself (and checking for failure return of nil).
NSObject's designated initializer is init.
This is our class's designated initializer.
That means that it is not legally initialized unless this gets called at some point.
We must always call our superclass's designated initializer from our designated initializer
(if this were just a convenience initializer, we'd have to call our own designated initializer from it).
NSObject's designated initializer is init.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 -initWithCardCount:usingDeck:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{
if (!_cards) _cards = [[NSMutableArray alloc] init];
return _cards;
}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{
self = [super init];
if (self) {
for (int i = 0; i < count; i++) {
}
}
return self;
}
@end
Next let's loop through the specified count of cards ...
Stanford CS193p
Winter 2013

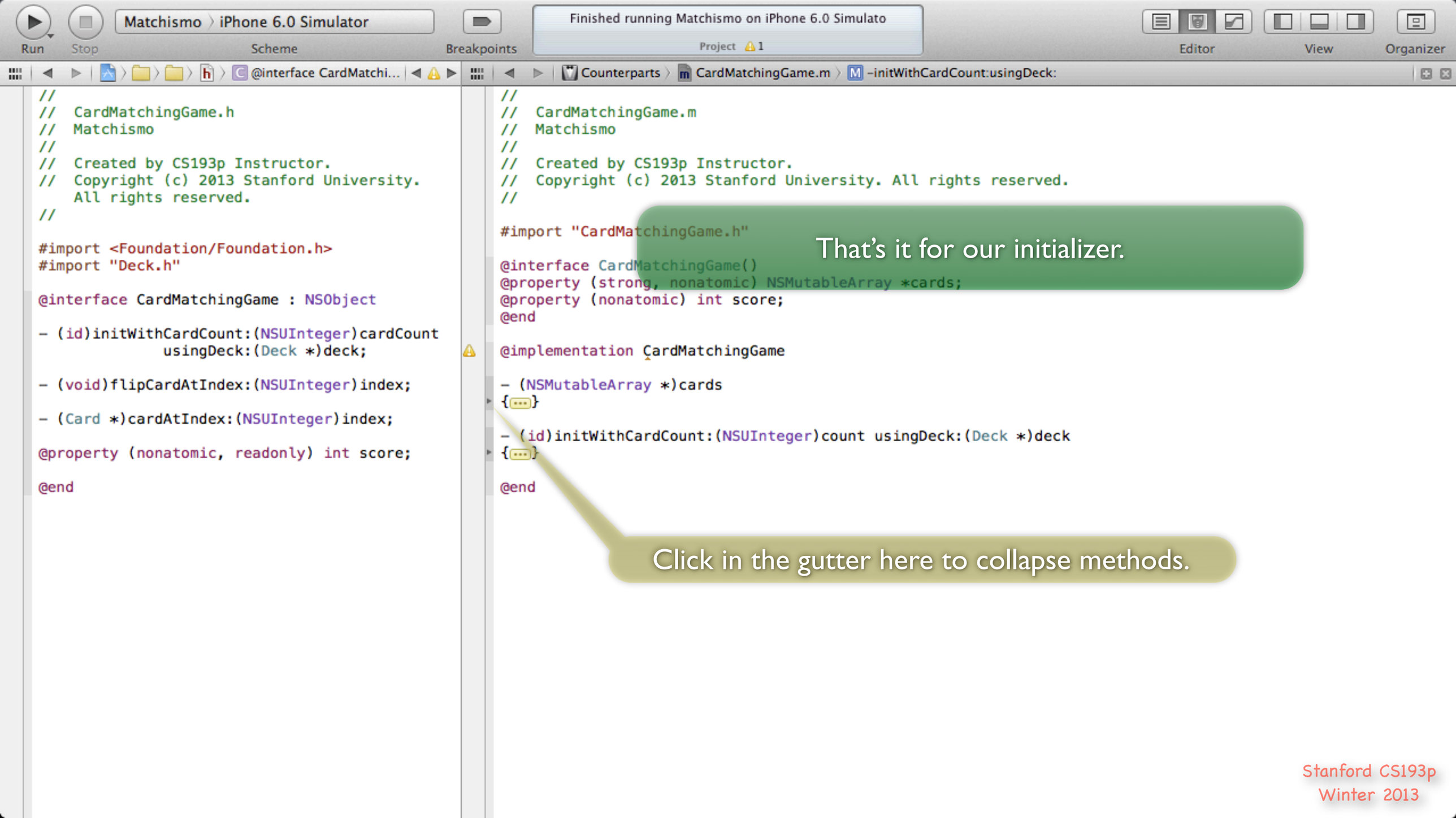

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 -initWithCardCount:usingDeck:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{
if (!_cards) _cards = [[NSMutableArray alloc] init];
return _cards;
}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{
self = [super init];
if (self) {
for (int i = 0; i < count; i++) {
Card *card = [deck drawRandomCard];
}
}
return self;
}
@end
... and draw a drawRandomCard from the specified deck.
Stanford CS193p
Winter 2013

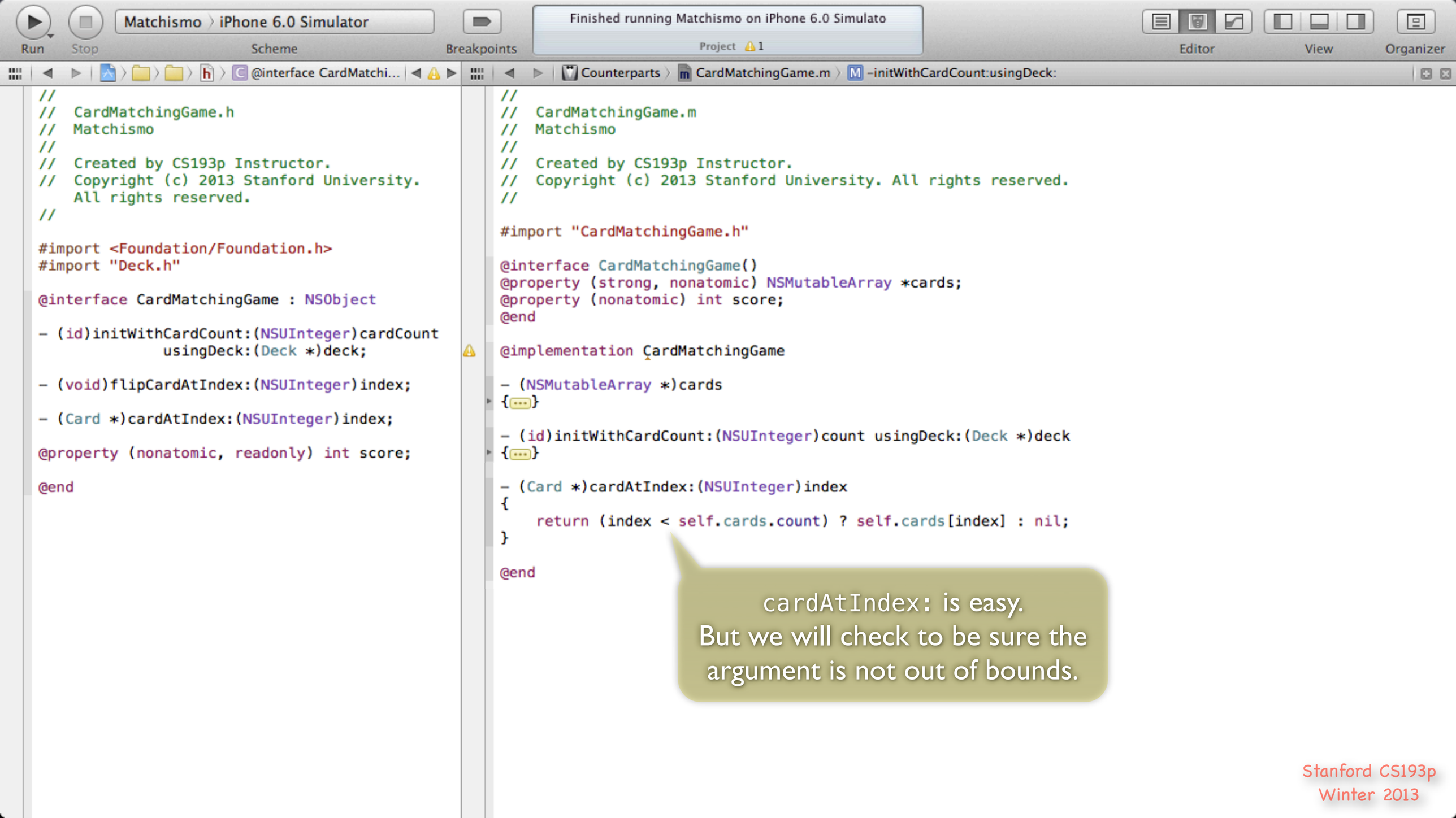
```
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//  All rights reserved.
//

#import <Foundation/Foundation.h>
#import "Deck.h"

@interface CardMatchingGame : NSObject

- (id)initWithCardCount:(NSUInteger)cardCount
              usingDeck:(Deck *)deck;

- (void)flipCardAtIndex:(NSUInteger)index;

- (Card *)cardAtIndex:(NSUInteger)index;

@property (nonatomic, readonly) int score;

@end
```

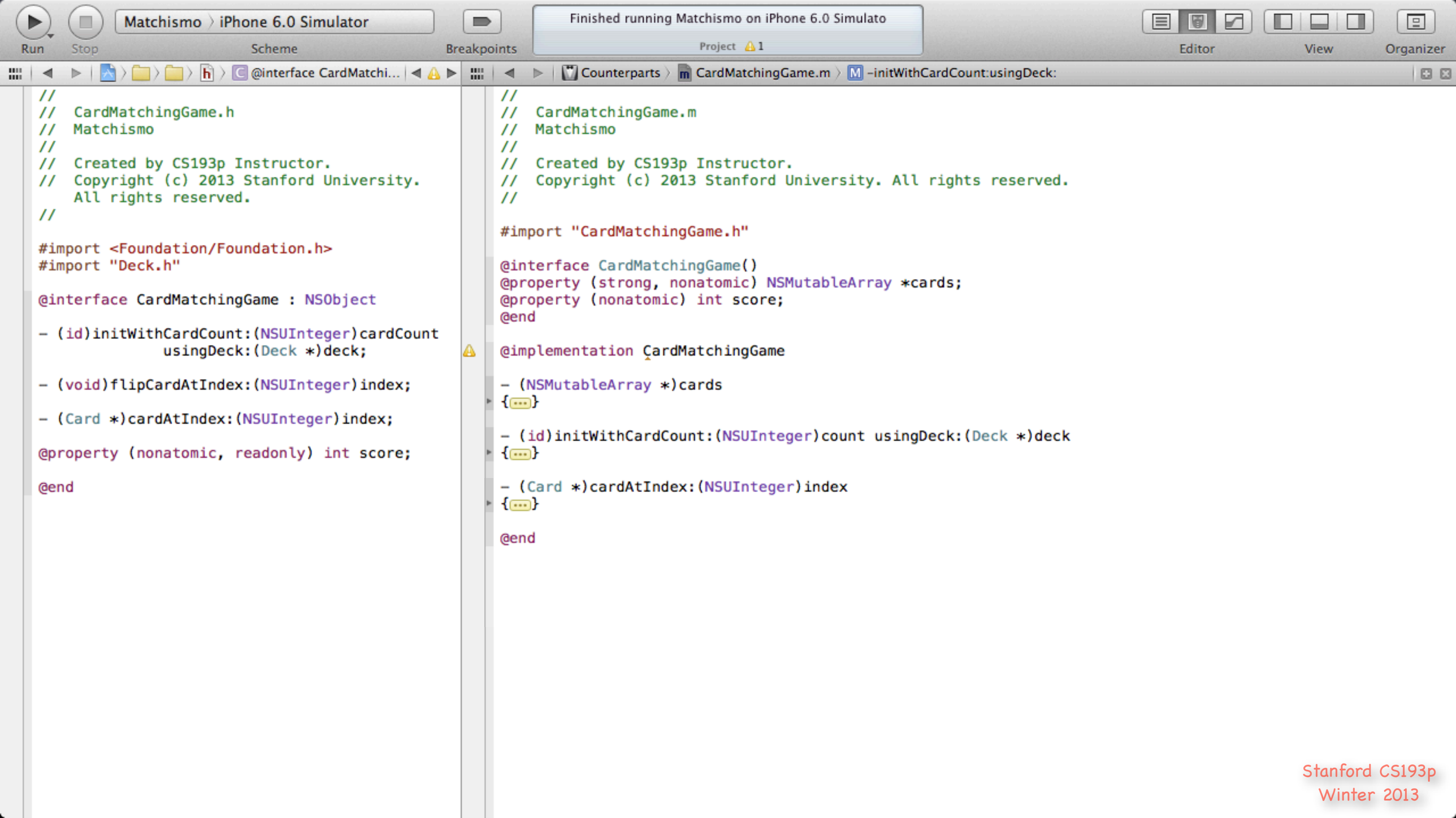
Note that we will return `nil` if we cannot initialize properly given the arguments passed.


```
//
//  CardMatchingGame.m
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University. All rights reserved.
//

#import "CardMatchingGame.h"

@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end

@implementation CardMatchingGame

- (NSMutableArray *)cards
{
    if (!_cards) _cards = [[NSMutableArray alloc] init];
    return _cards;
}

- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{
    self = [super init];

    if (self) {
        for (int i = 0; i < count; i++) {
            Card *card = [deck drawRandomCard];
            if (!card) {
                self = nil;
            } else {
                self.cards[i] = card;
            }
        }
    }

    return self;
}

@end
```

We'll protect ourselves from bad (or insufficient) decks.
Adding `nil` to an `NSMutableArray` will crash.


Matchismo 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 -initWithCardCount:usingDeck:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{...}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
@end
That's it for our initializer.
Click in the gutter here to collapse methods.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
@interface CardMatchi...
Counterparts 〉 CardMatchingGame.m 〉 -initWithCardCount:usingDeck:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
//
// CardMatchingGame.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{...}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
- (Card *)cardAtIndex:(NSUInteger)index
{
return (index < self.cards.count) ? self.cards[index] : nil;
}
@end
cardAtIndex: is easy.
But we will check to be sure the
argument is not out of bounds.



Stanford CS193p
Winter 2013

```
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//  All rights reserved.
//

#import <Foundation/Foundation.h>
#import "Deck.h"

@interface CardMatchingGame : NSObject

- (id)initWithCardCount:(NSUInteger)cardCount
              usingDeck:(Deck *)deck;

- (void)flipCardAtIndex:(NSUInteger)index;

- (Card *)cardAtIndex:(NSUInteger)index;

@property (nonatomic, readonly) int score;

@end
```



```
#import "CardMatchingGame.h"

@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end

@implementation CardMatchingGame

- (NSMutableArray *)cards
{...}

- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}

- (Card *)cardAtIndex:(NSUInteger)index
{...}

- (void)flipCardAtIndex:(NSUInteger)index
{
    Card *card = [self cardAtIndex:index];

    if (!card.isUnplayable) {
        card.faceUp = !card.isFaceUp;
    }
}

@end
```

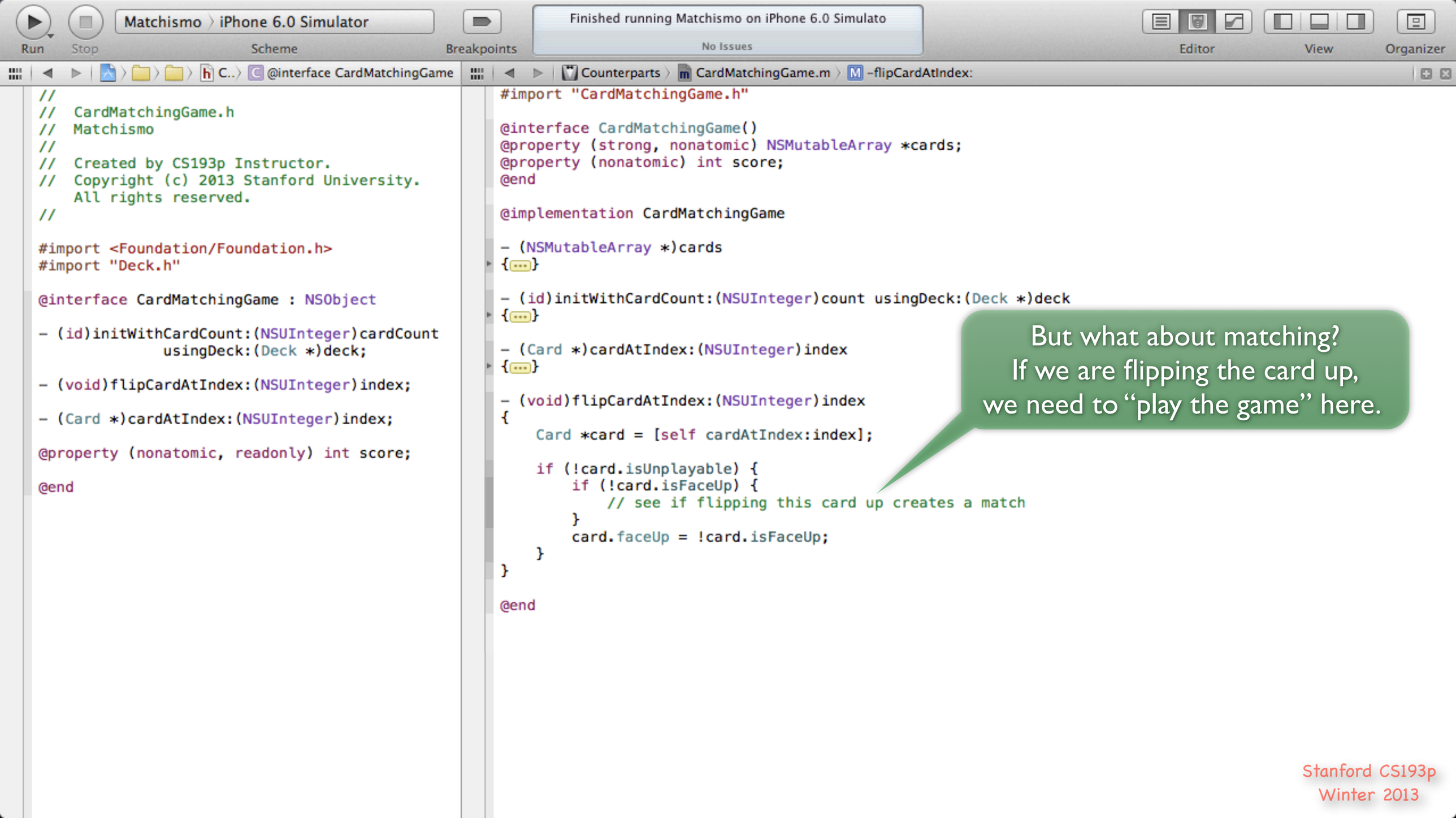
`flipCardAtIndex:` is the guts of our class.
It's where the game logic lives.

Start by grabbing the card to flip and making sure it is playable.

If so, we can flip it.

Here's a reminder of what `Card`'s `@interface` looks like:

```
@interface Card : NSObject

@property (strong, nonatomic) NSString *contents;

@property (nonatomic, getter=isFaceUp) BOOL faceUp;
@property (nonatomic, getter=isUnplayable) BOOL unplayable;

- (int)match:(NSArray *)otherCards;

@end
```


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
C.. 〉 @interface CardMatchingGame
Counterparts 〉 CardMatchingGame.m 〉 -flipCardAtIndex:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{...}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
- (Card *)cardAtIndex:(NSUInteger)index
{...}
- (void)flipCardAtIndex:(NSUInteger)index
{
Card *card = [self cardAtIndex:index];
if (!card.isUnplayable) {
if (!card.isFaceUp) {
// see if flipping this card up creates a match
}
card.faceUp = !card.isFaceUp;
}
}
@end
But what about matching?
If we are flipping the card up,
we need to “play the game” here.
Stanford CS193p
Winter 2013

```
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//  All rights reserved.
//

#import <Foundation/Foundation.h>
#import "Deck.h"

@interface CardMatchingGame : NSObject

- (id)initWithCardCount:(NSUInteger)cardCount
              usingDeck:(Deck *)deck;

- (void)flipCardAtIndex:(NSUInteger)index;

- (Card *)cardAtIndex:(NSUInteger)index;

@property (nonatomic, readonly) int score;

@end
```



```
#import "CardMatchingGame.h"

@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end

@implementation CardMatchingGame

- (NSMutableArray *)cards
{...}

- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}

- (Card *)cardAtIndex:(NSUInteger)index
{...}

- (void)flipCardAtIndex:(NSUInteger)index
{
    Card *card = [self cardAtIndex:index];

    if (!card.isUnplayable) {
        if (!card.isFaceUp) {
            for (Card *otherCard in self.cards) {
                if (otherCard.isFaceUp && !otherCard.isUnplayable) {

                }
            }
        }
        card.faceUp = !card.isFaceUp;
    }
}

@end
```

Loop through the other cards looking for another face up, playable one.

```
//
//  CardMatchingGame.h
//  Matchismo
//
//  Created by CS193p Instructor.
//  Copyright (c) 2013 Stanford University.
//  All rights reserved.
//

#import <Foundation/Foundation.h>
#import "Deck.h"

@interface CardMatchingGame : NSObject

- (id)initWithCardCount:(NSUInteger)cardCount
              usingDeck:(Deck *)deck;

- (void)flipCardAtIndex:(NSUInteger)index;

- (Card *)cardAtIndex:(NSUInteger)index;

@property (nonatomic, readonly) int score;

@end
```



```
#import "CardMatchingGame.h"

@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end

@implementation CardMatchingGame

- (NSMutableArray *)cards
{...}

- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}

- (Card *)cardAtIndex:(NSUInteger)index
{...}

- (void)flipCardAtIndex:(NSUInteger)index
{
    Card *card = [self cardAtIndex:index];

    if (!card.isUnplayable) {
        if (!card.isFaceUp) {
            for (Card *otherCard in self.cards) {
                if (otherCard.isFaceUp && !otherCard.isUnplayable) {
                    int matchScore = [card match:@[otherCard]];
                }
            }
        }
        card.faceUp = !card.isFaceUp;
```

If we find it, check to see if it matches using `Card`'s `match:` method.

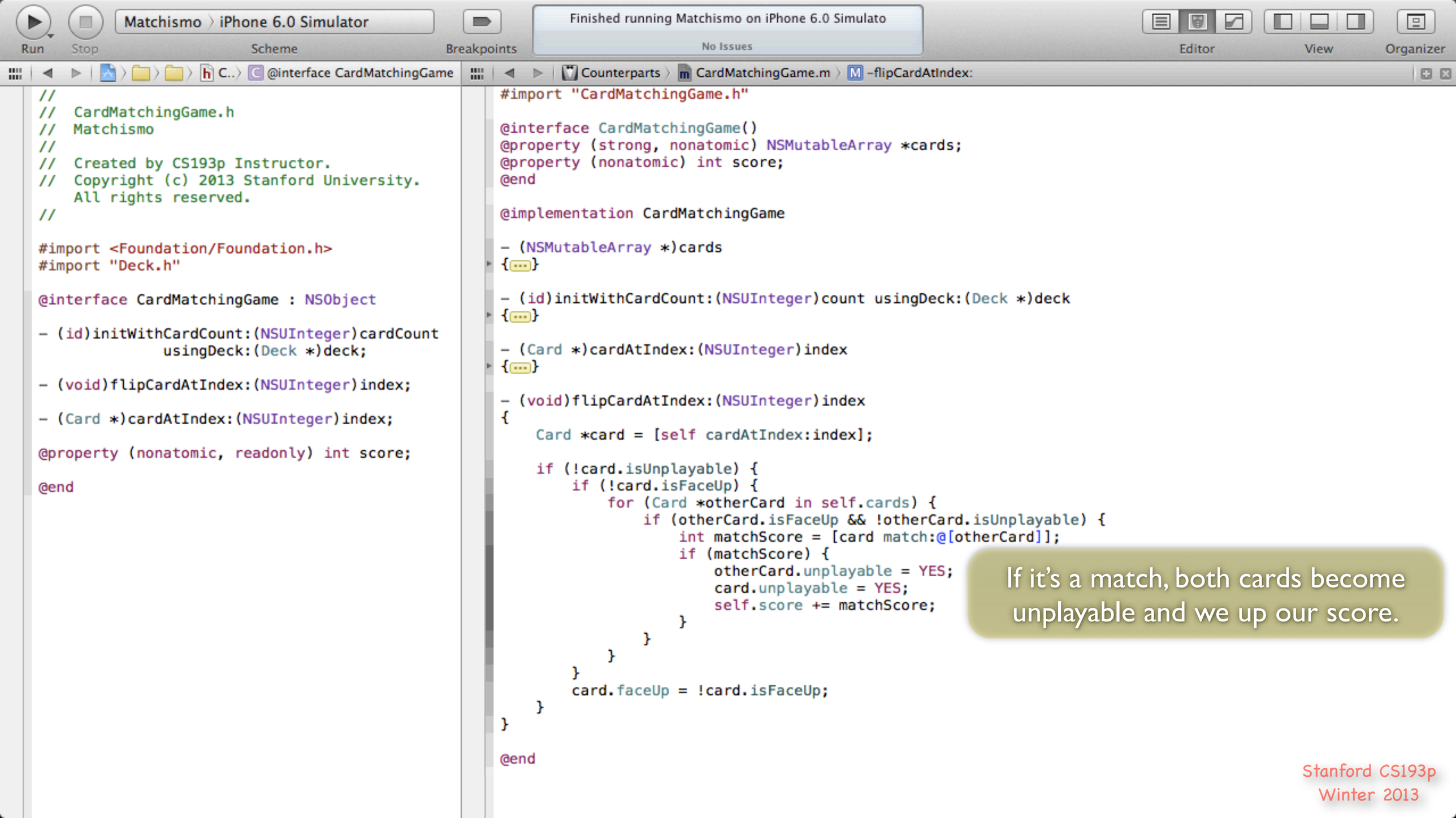
`match:` takes an `NSArray` of other cards in case a subclass can match multiple cards.
Since our matching game is only a 2-card matching game, we just create an array with one card in it.

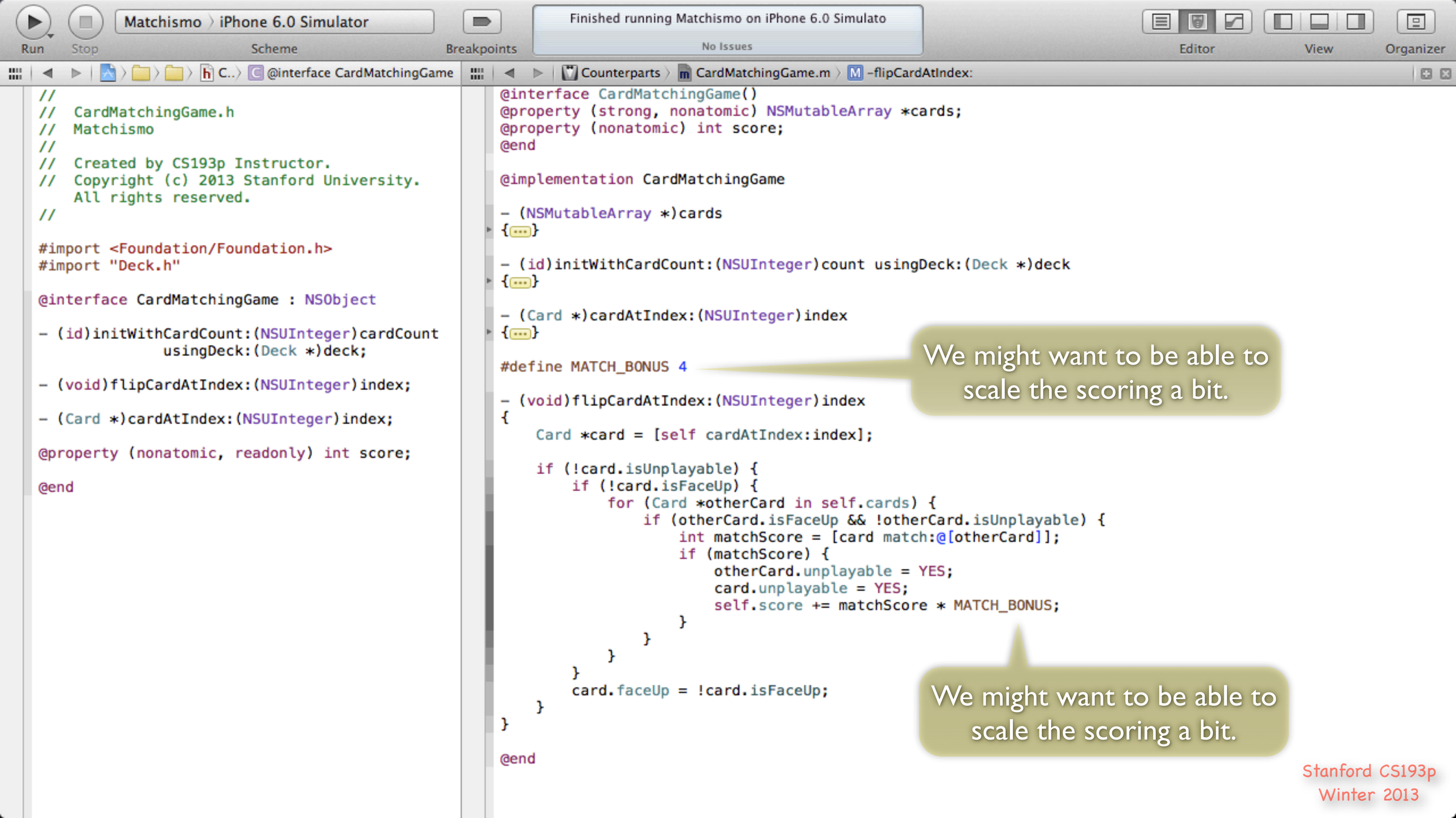
`match:` returns how good a match it was (zero if not a match).

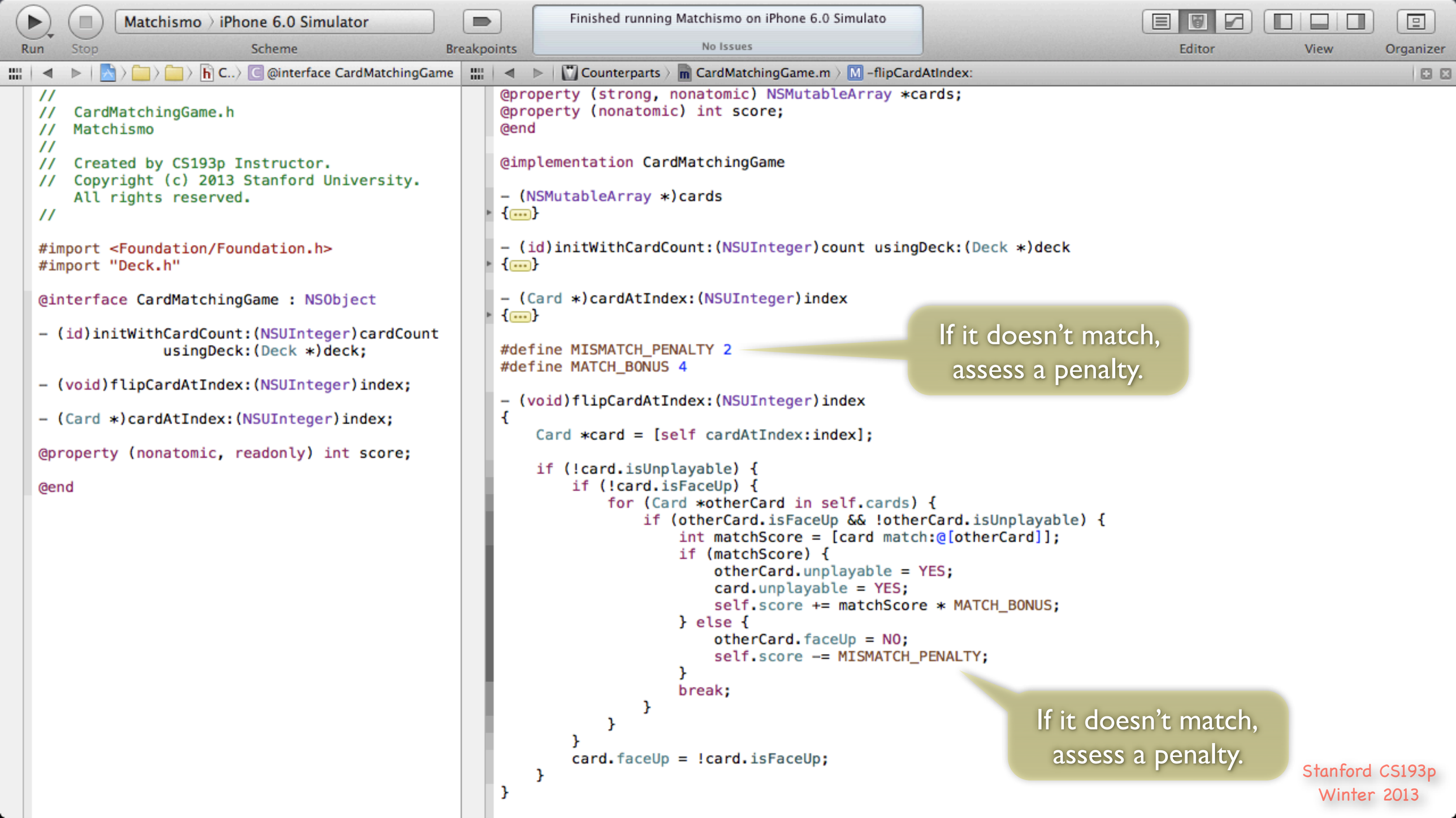
Yes, it is perfectly legal to use the `@[]` array creation syntax here!

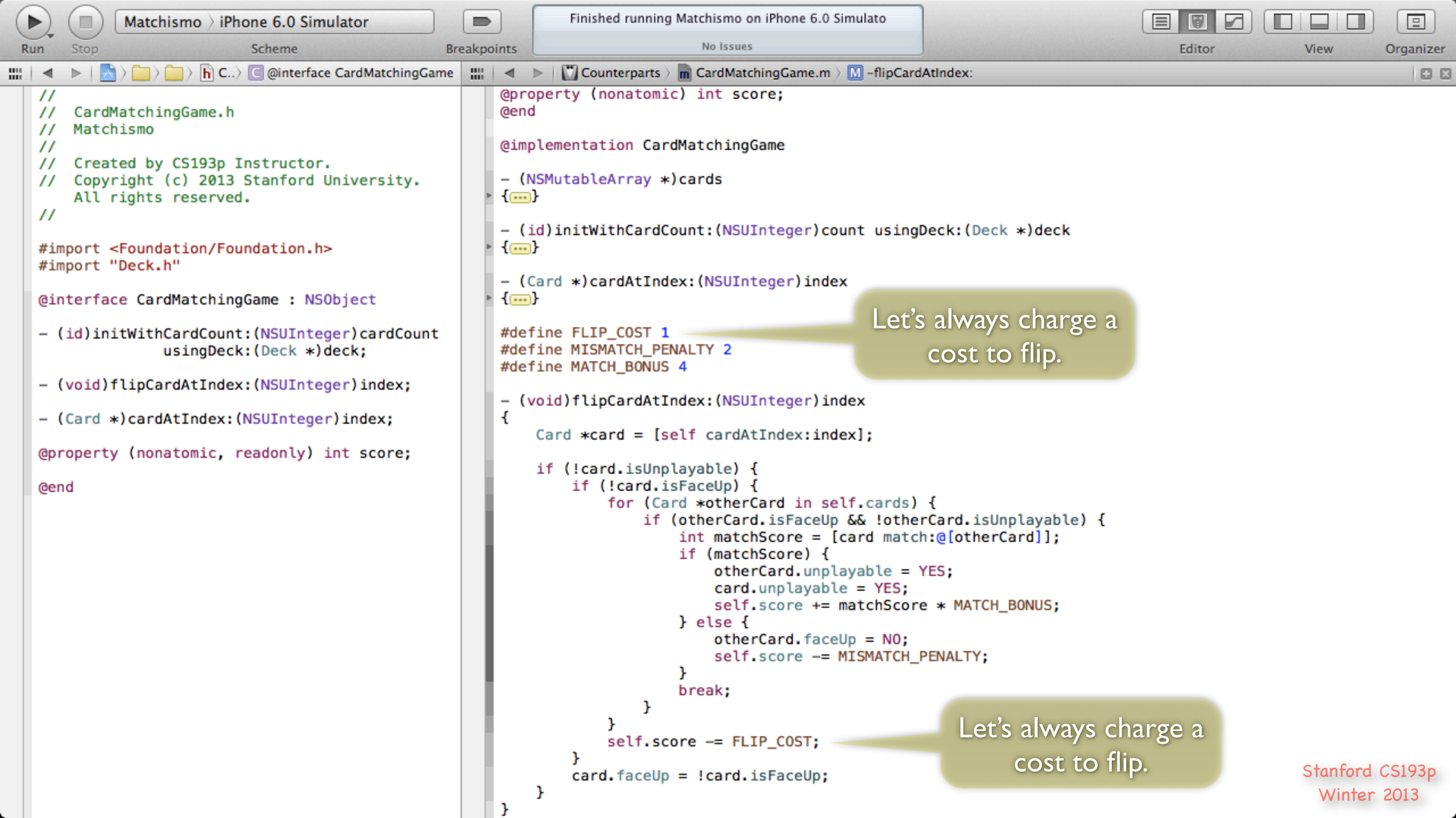
Here's a reminder of what `Card`'s `@interface` looks like:

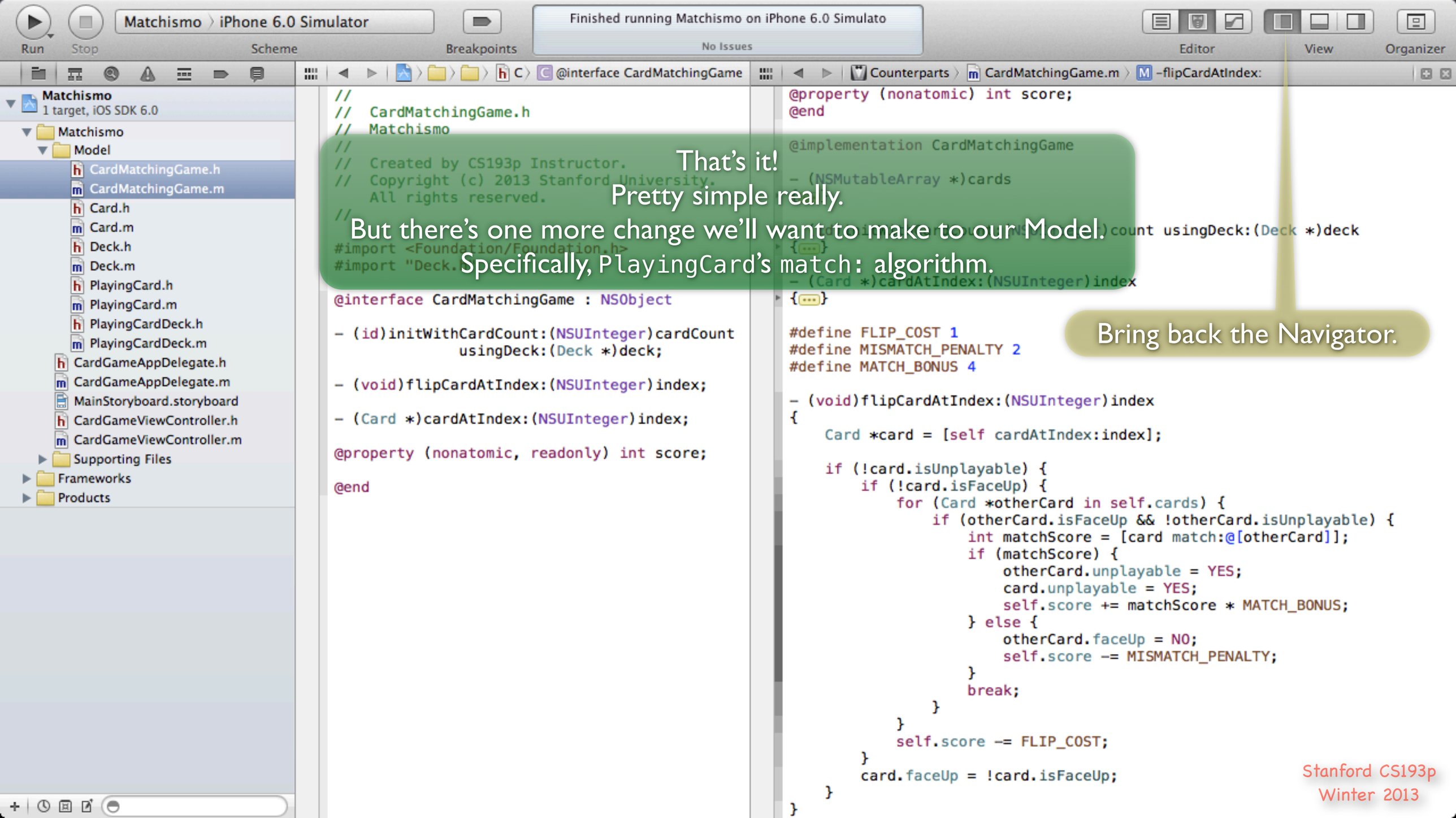
```
@interface Card : NSObject

@property (strong, nonatomic) NSString *contents;

@property (nonatomic, getter=isFaceUp) BOOL faceUp;
@property (nonatomic, getter=isUnplayable) BOOL unplayable;

- (int)match:(NSArray *)otherCards;

@end
```


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
C.. 〉 @interface CardMatchingGame
Counterparts 〉 CardMatchingGame.m 〉 -flipCardAtIndex:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
#import "CardMatchingGame.h"
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{...}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
- (Card *)cardAtIndex:(NSUInteger)index
{...}
- (void)flipCardAtIndex:(NSUInteger)index
{
Card *card = [self cardAtIndex:index];
if (!card.isUnplayable) {
if (!card.isFaceUp) {
for (Card *otherCard in self.cards) {
if (otherCard.isFaceUp && !otherCard.isUnplayable) {
int matchScore = [card match:@[otherCard]];
if (matchScore) {
otherCard.unplayable = YES;
card.unplayable = YES;
self.score += matchScore;
}
}
}
}
card.faceUp = !card.isFaceUp;
}
}
@end
If it's a match, both cards become unplayable and we up our score.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
C.. 〉 @interface CardMatchingGame
Counterparts 〉 CardMatchingGame.m 〉 -flipCardAtIndex:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
@interface CardMatchingGame()
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{...}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
- (Card *)cardAtIndex:(NSUInteger)index
{...}
#define MATCH_BONUS 4
- (void)flipCardAtIndex:(NSUInteger)index
{
Card *card = [self cardAtIndex:index];
if (!card.isUnplayable) {
if (!card.isFaceUp) {
for (Card *otherCard in self.cards) {
if (otherCard.isFaceUp && !otherCard.isUnplayable) {
int matchScore = [card match:@[otherCard]];
if (matchScore) {
otherCard.unplayable = YES;
card.unplayable = YES;
self.score += matchScore * MATCH_BONUS;
}
}
}
}
card.faceUp = !card.isFaceUp;
}
}
@end
We might want to be able to scale the scoring a bit.
We might want to be able to scale the scoring a bit.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
C.. 〉 @interface CardMatchingGame
Counterparts 〉 CardMatchingGame.m 〉 -flipCardAtIndex:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
@property (strong, nonatomic) NSMutableArray *cards;
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{...}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
- (Card *)cardAtIndex:(NSUInteger)index
{...}
#define MISMATCH_PENALTY 2
#define MATCH_BONUS 4
- (void)flipCardAtIndex:(NSUInteger)index
{
Card *card = [self cardAtIndex:index];
if (!card.isUnplayable) {
if (!card.isFaceUp) {
for (Card *otherCard in self.cards) {
if (otherCard.isFaceUp && !otherCard.isUnplayable) {
int matchScore = [card match:@[otherCard]];
if (matchScore) {
otherCard.unplayable = YES;
card.unplayable = YES;
self.score += matchScore * MATCH_BONUS;
} else {
otherCard.faceUp = NO;
self.score -= MISMATCH_PENALTY;
}
break;
}
}
}
card.faceUp = !card.isFaceUp;
}
}
If it doesn't match, assess a penalty.
If it doesn't match, assess a penalty.
Stanford CS193p
Winter 2013

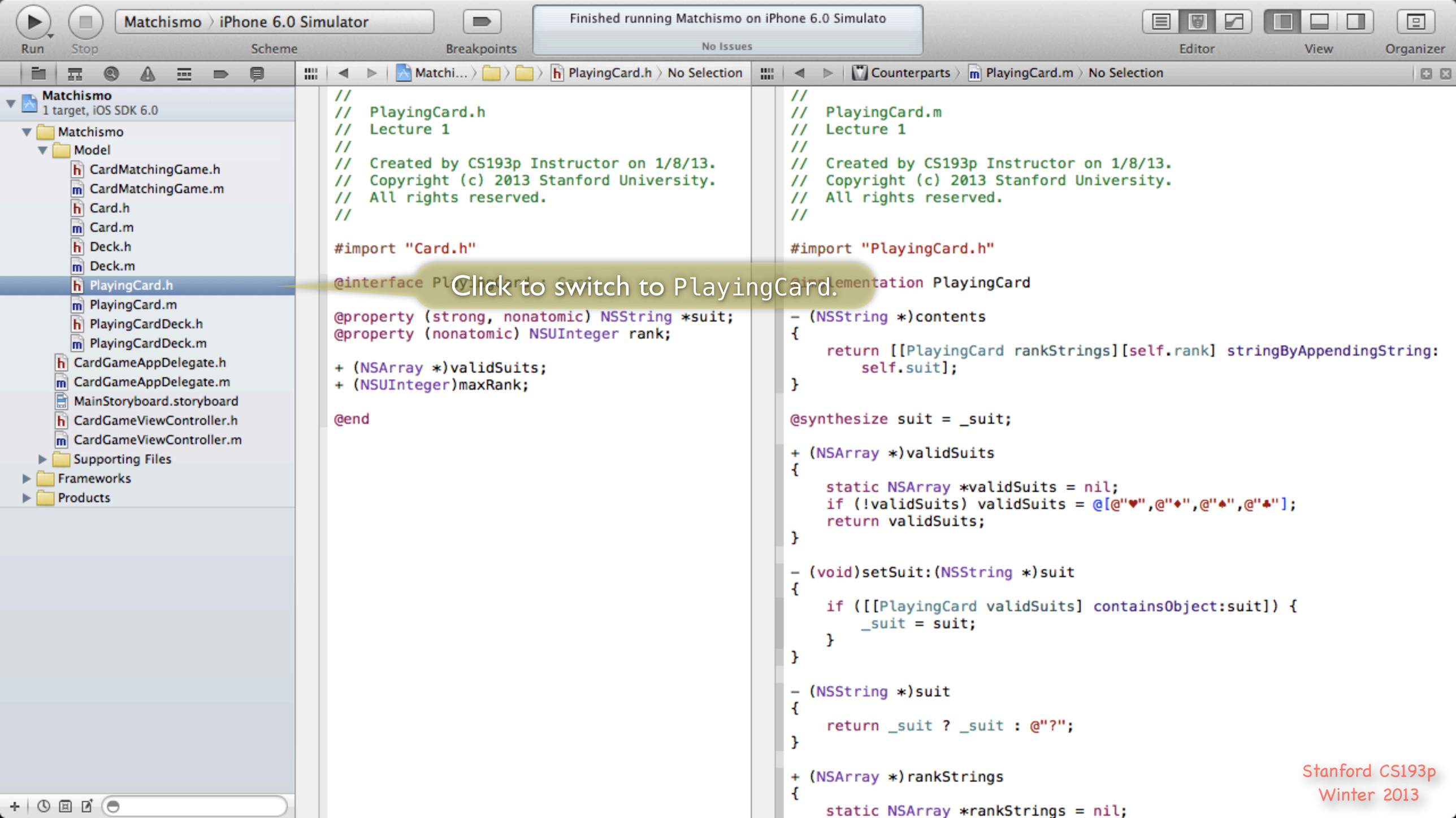

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
C.. 〉 @interface CardMatchingGame
Counterparts 〉 CardMatchingGame.m 〉 -flipCardAtIndex:
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
{...}
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
- (Card *)cardAtIndex:(NSUInteger)index
{...}
#define FLIP_COST 1
#define MISMATCH_PENALTY 2
#define MATCH_BONUS 4
- (void)flipCardAtIndex:(NSUInteger)index
{
Card *card = [self cardAtIndex:index];
if (!card.isUnplayable) {
if (!card.isFaceUp) {
for (Card *otherCard in self.cards) {
if (otherCard.isFaceUp && !otherCard.isUnplayable) {
int matchScore = [card match:@[otherCard]];
if (matchScore) {
otherCard.unplayable = YES;
card.unplayable = YES;
self.score += matchScore * MATCH_BONUS;
} else {
otherCard.faceUp = NO;
self.score -= MISMATCH_PENALTY;
}
break;
}
}
self.score -= FLIP_COST;
}
card.faceUp = !card.isFaceUp;
}
}
Let's always charge a cost to flip.
Let's always charge a cost to flip.
Stanford CS193p
Winter 2013

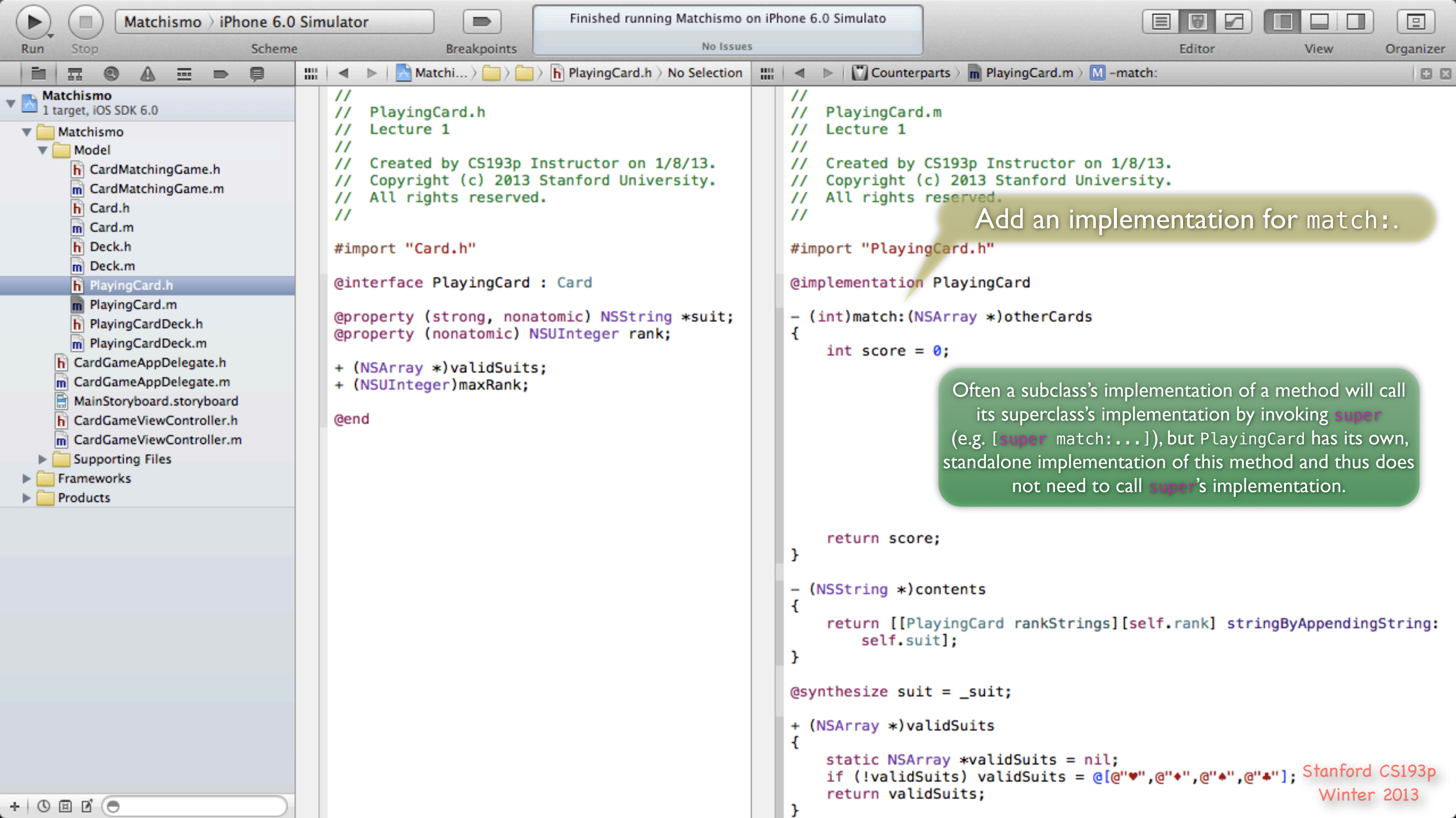

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
C 〉 @interface CardMatchingGame
//
// CardMatchingGame.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import <Foundation/Foundation.h>
#import "Deck.h"
@interface CardMatchingGame : NSObject
- (id)initWithCardCount:(NSUInteger)cardCount
usingDeck:(Deck *)deck;
- (void)flipCardAtIndex:(NSUInteger)index;
- (Card *)cardAtIndex:(NSUInteger)index;
@property (nonatomic, readonly) int score;
@end
That's it!
Pretty simple really.
But there's one more change we'll want to make to our Model.
Specifically, PlayingCard's match: algorithm.
Counterparts 〉 CardMatchingGame.m 〉 -flipCardAtIndex:
@property (nonatomic) int score;
@end
@implementation CardMatchingGame
- (NSMutableArray *)cards
- (id)initWithCardCount:(NSUInteger)count usingDeck:(Deck *)deck
{...}
- (Card *)cardAtIndex:(NSUInteger)index
{...}
#define FLIP_COST 1
#define MISMATCH_PENALTY 2
#define MATCH_BONUS 4
- (void)flipCardAtIndex:(NSUInteger)index
{
Card *card = [self cardAtIndex:index];
if (!card.isUnplayable) {
if (!card.isFaceUp) {
for (Card *otherCard in self.cards) {
if (otherCard.isFaceUp && !otherCard.isUnplayable) {
int matchScore = [card match:@[otherCard]];
if (matchScore) {
otherCard.unplayable = YES;
card.unplayable = YES;
self.score += matchScore * MATCH_BONUS;
} else {
otherCard.faceUp = NO;
self.score -= MISMATCH_PENALTY;
}
break;
}
}
self.score -= FLIP_COST;
}
card.faceUp = !card.isFaceUp;
}
}
Bring back the Navigator.
Stanford CS193p
Winter 2013

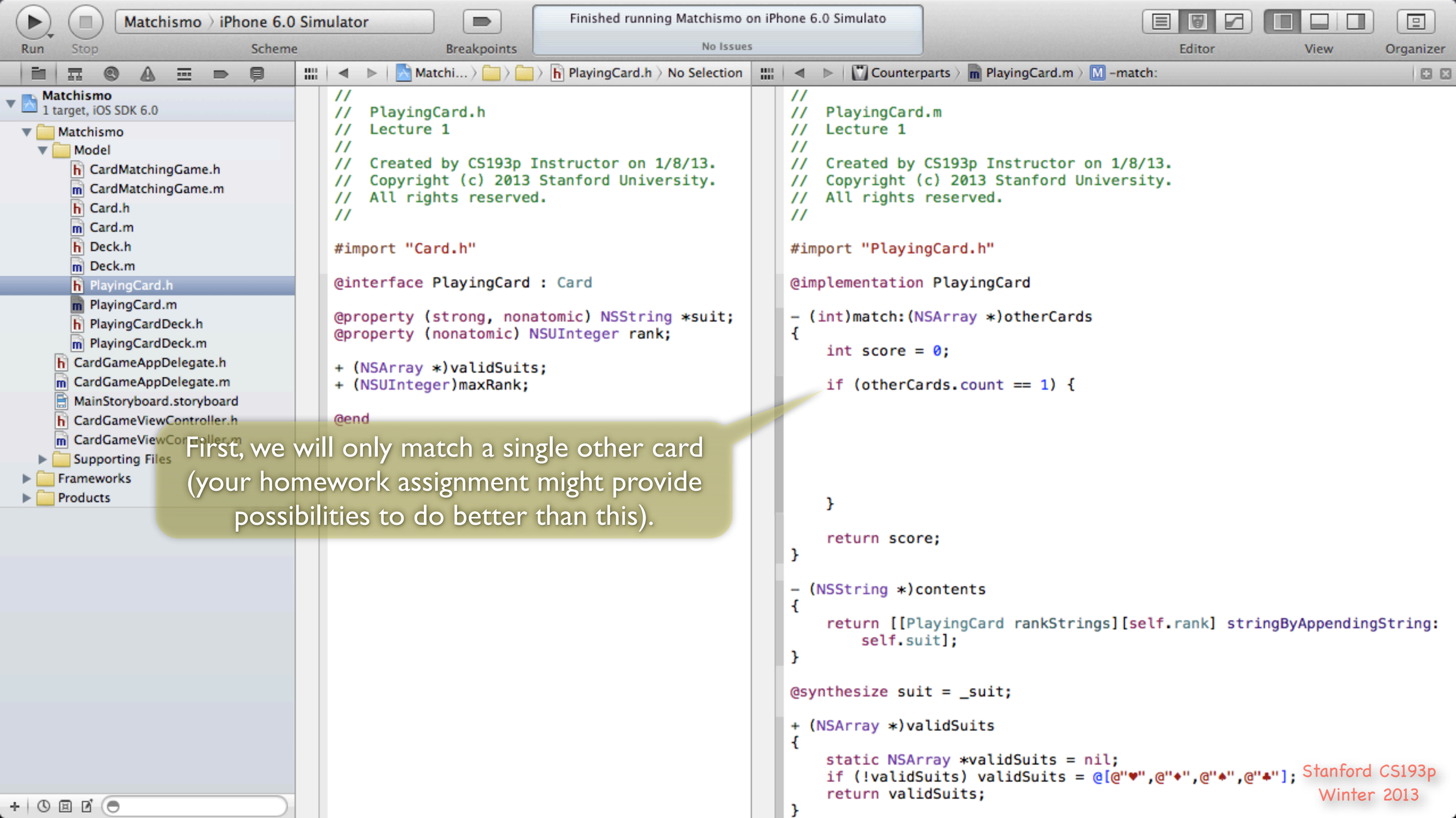
Click on Card in the Navigator
so that we remind ourselves what its match: algorithm is.

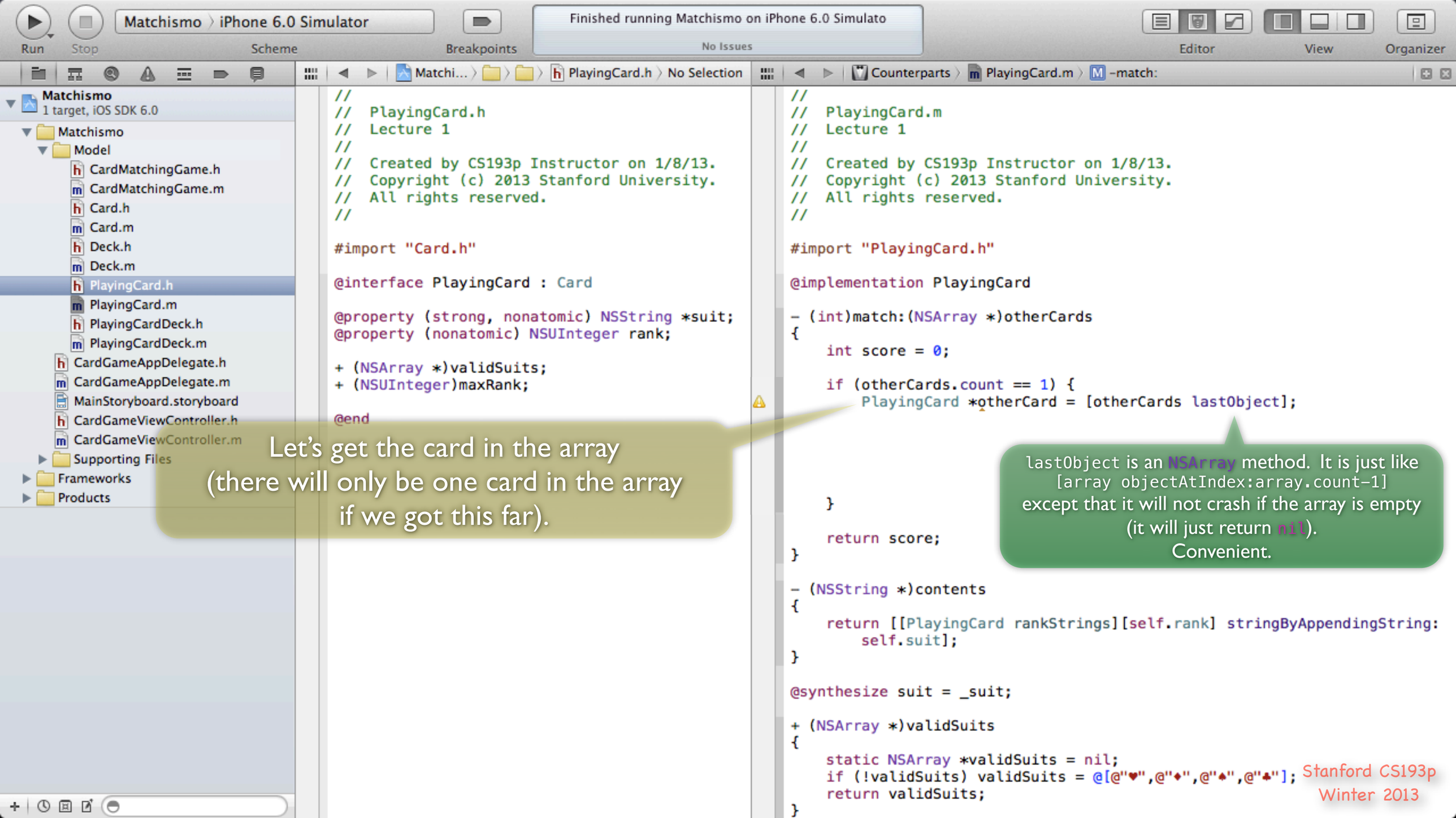
Card matches only if the cards are exactly the same
(that is to say, their contents @property values are equal).
PlayingCards should match if the suit and/or rank is the same.
Let's go to PlayingCard and override Card's implementation
of match: to make this so.

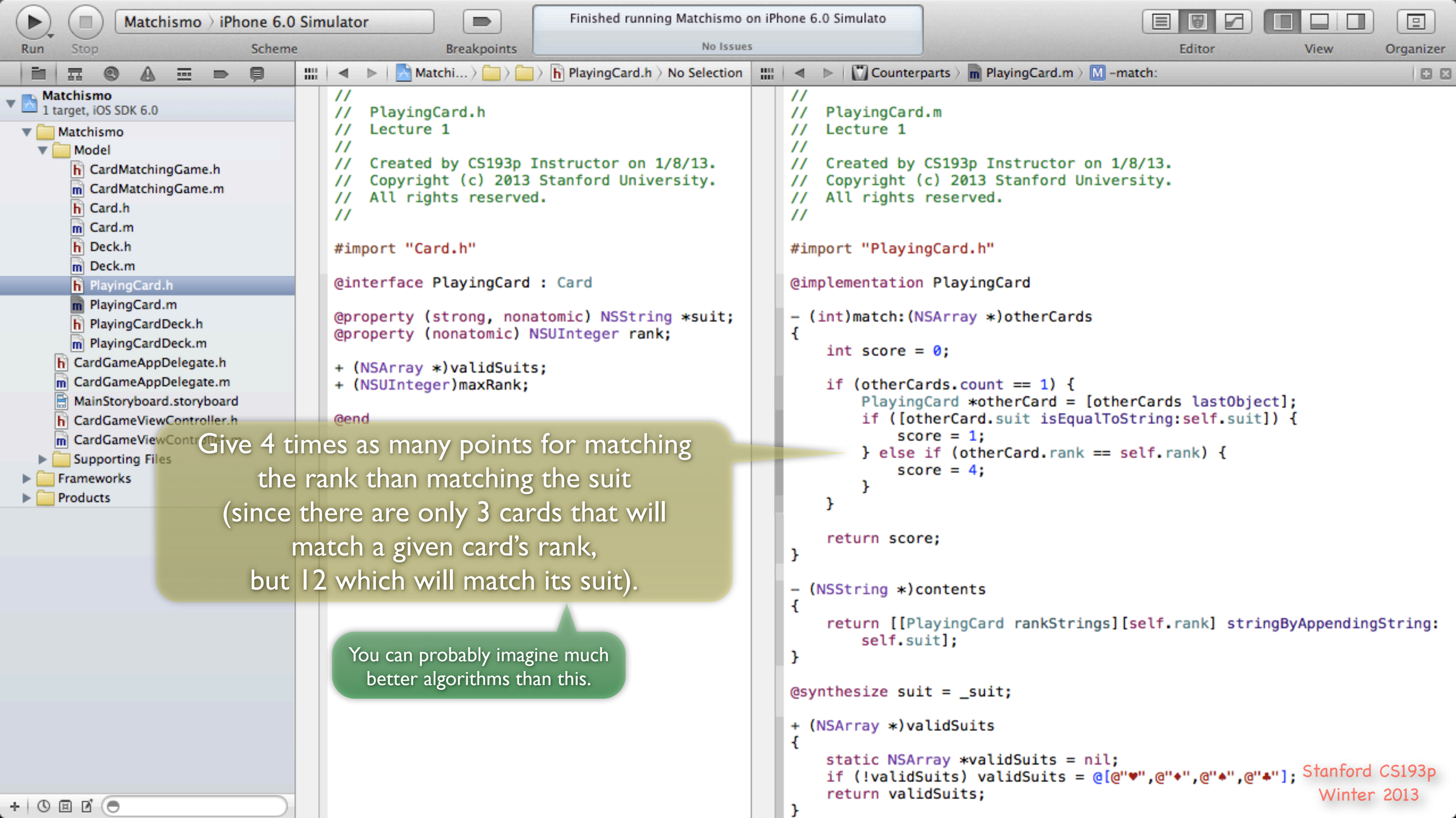
Stanford CS193p
Winter 2013

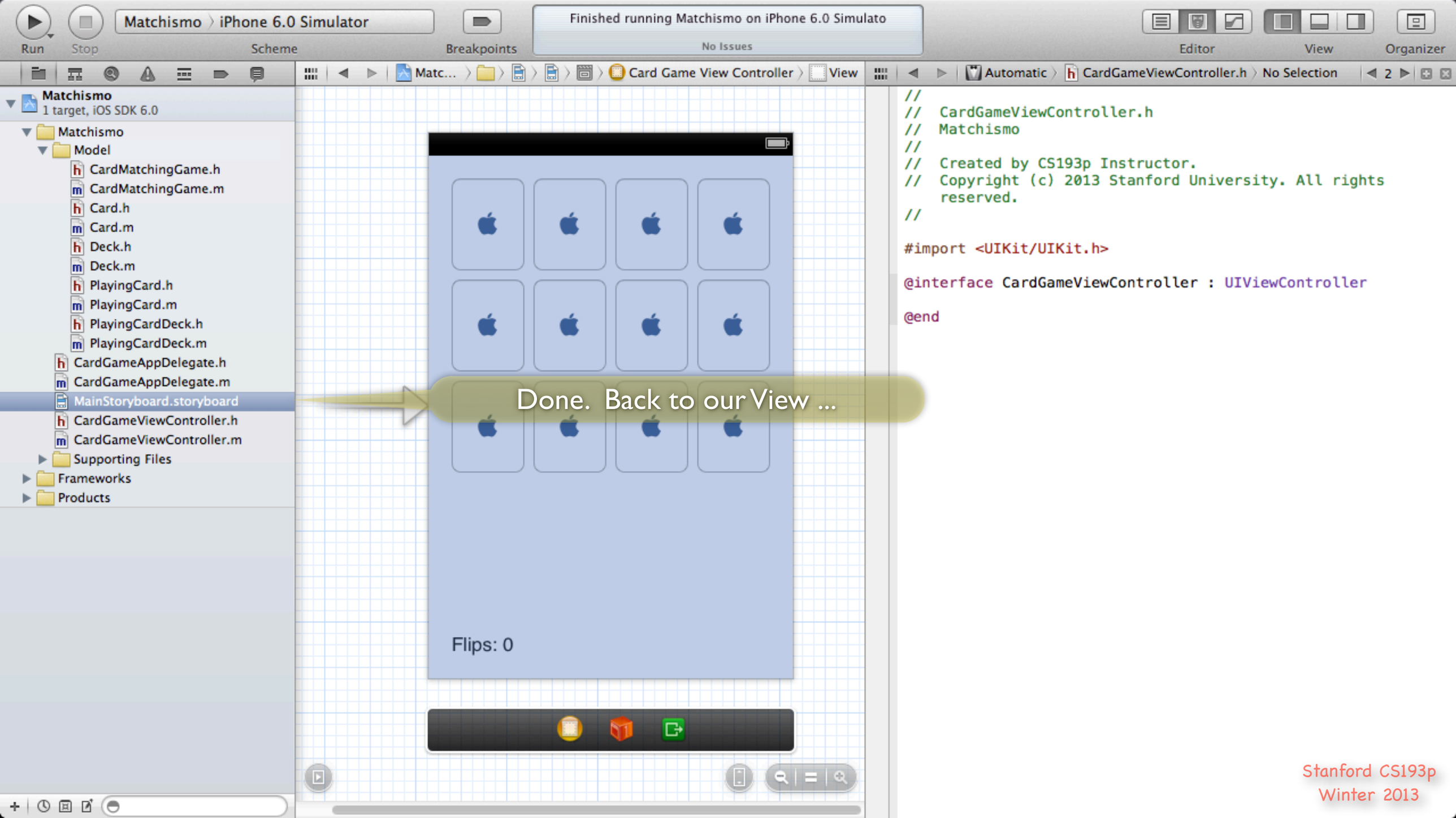

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
Matchi... 〉 PlayingCard.h 〉 No Selection
Counterparts 〉 PlayingCard.m 〉 No Selection
Click to switch to PlayingCard.
//
// PlayingCard.h
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "Card.h"
@interface PlayingCard : Card
@property (strong, nonatomic) NSString *suit;
@property (nonatomic) NSUInteger rank;
+ (NSArray *)validSuits;
+ (NSUInteger)maxRank;
@end
//
// PlayingCard.m
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "PlayingCard.h"
@implementation PlayingCard
- (NSString *)contents
{
return [[PlayingCard rankStrings][self.rank] stringByAppendingString:
self.suit];
}
@synthesize suit = _suit;
+ (NSArray *)validSuits
{
static NSArray *validSuits = nil;
if (!validSuits) validSuits = @[@"♥",@"♦",@"♠",@"♣"];
return validSuits;
}
- (void)setSuit:(NSString *)suit
{
if ([[PlayingCard validSuits] containsObject:suit]) {
_suit = suit;
}
}
- (NSString *)suit
{
return _suit ? _suit : @"?";
}
+ (NSArray *)rankStrings
{
static NSArray *rankStrings = nil;
Stanford CS193p
Winter 2013

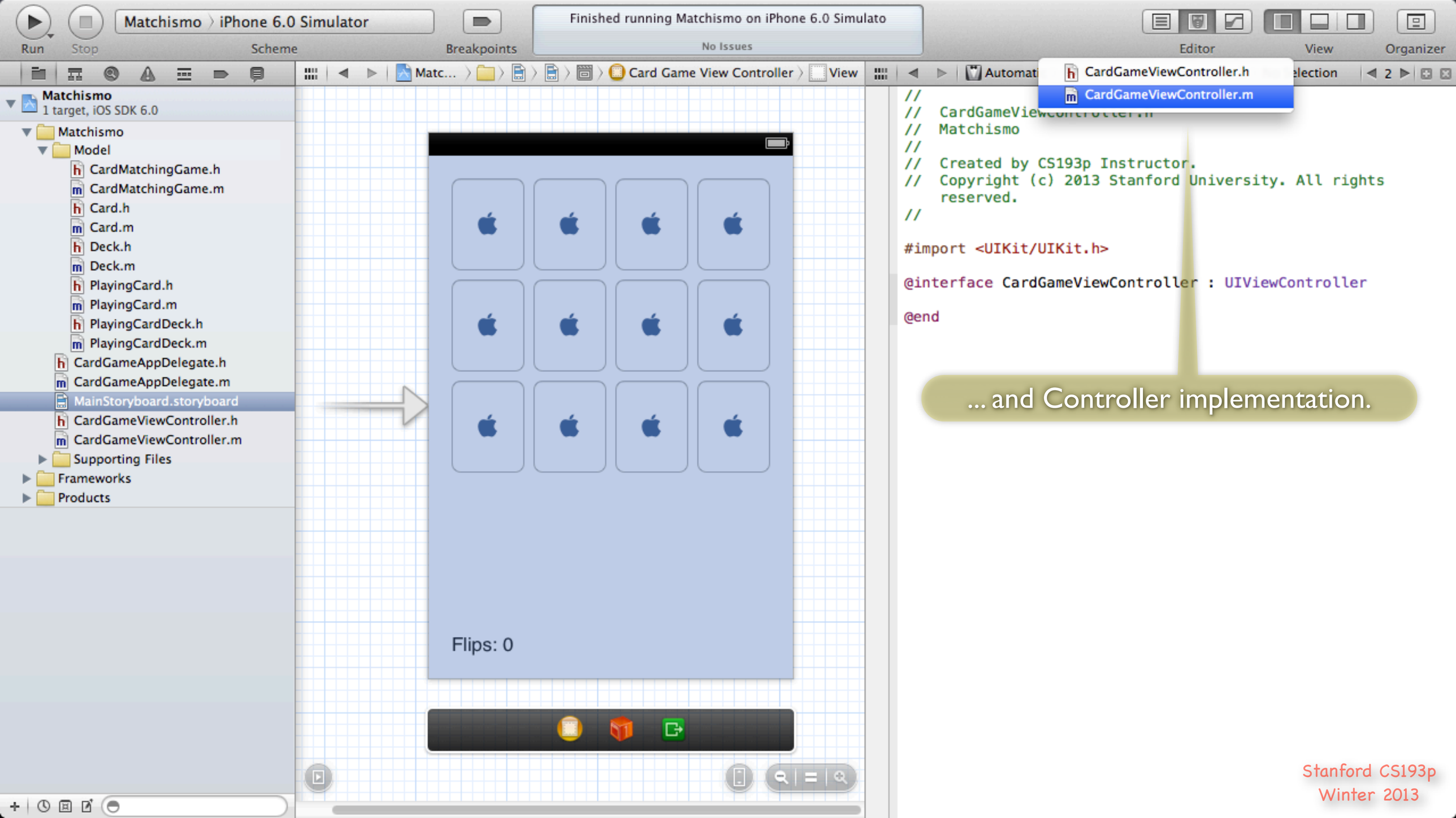

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
Matchi... 〉 PlayingCard.h 〉 No Selection
//
// PlayingCard.h
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "Card.h"
@interface PlayingCard : Card
@property (strong, nonatomic) NSString *suit;
@property (nonatomic) NSUInteger rank;
+ (NSArray *)validSuits;
+ (NSUInteger)maxRank;
@end
Counterparts 〉 PlayingCard.m 〉 -match:
//
// PlayingCard.m
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
Add an implementation for match:.
#import "PlayingCard.h"
@implementation PlayingCard
- (int)match:(NSArray *)otherCards
{
int score = 0;
Often a subclass's implementation of a method will call its superclass's implementation by invoking super (e.g. [super match:...]), but PlayingCard has its own, standalone implementation of this method and thus does not need to call super's implementation.
return score;
}
- (NSString *)contents
{
return [[PlayingCard rankStrings][self.rank] stringByAppendingString:
self.suit];
}
@synthesize suit = _suit;
+ (NSArray *)validSuits
{
static NSArray *validSuits = nil;
if (!validSuits) validSuits = @[@"♥",@"♦",@"♠",@"♣"];
return validSuits;
}
Stanford CS193p
Winter 2013

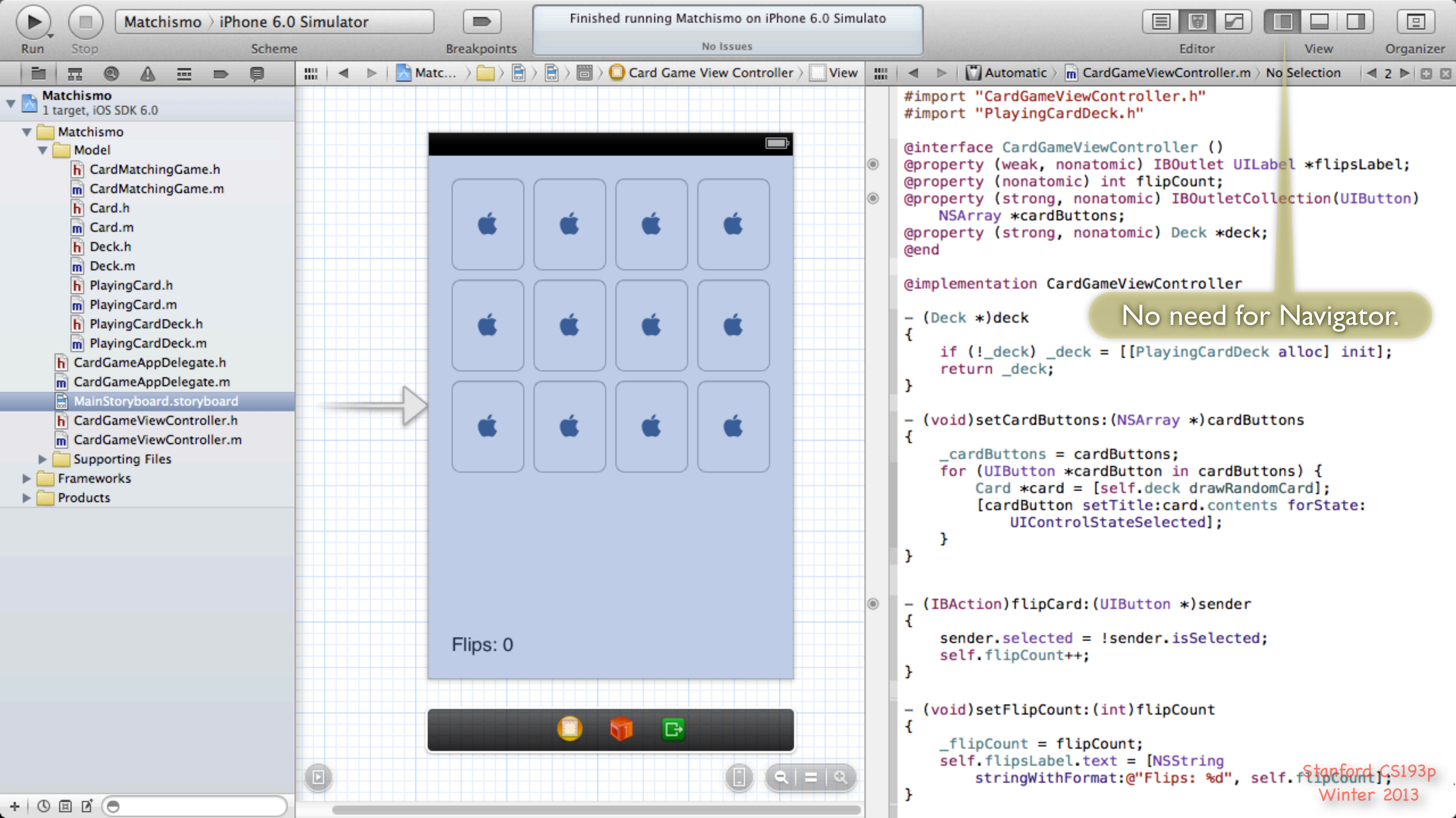

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
Supporting Files
Frameworks
Products
Matchi... 〉 PlayingCard.h 〉 No Selection
//
// PlayingCard.h
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "Card.h"
@interface PlayingCard : Card
@property (strong, nonatomic) NSString *suit;
@property (nonatomic) NSUInteger rank;
+ (NSArray *)validSuits;
+ (NSUInteger)maxRank;
@end
Counterparts 〉 PlayingCard.m 〉 -match:
//
// PlayingCard.m
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "PlayingCard.h"
@implementation PlayingCard
- (int)match:(NSArray *)otherCards
{
int score = 0;
if (otherCards.count == 1) {
}
return score;
}
- (NSString *)contents
{
return [[PlayingCard rankStrings][self.rank] stringByAppendingString:
self.suit];
}
@synthesize suit = _suit;
+ (NSArray *)validSuits
{
static NSArray *validSuits = nil;
if (!validSuits) validSuits = @[@"♥",@"♦",@"♠",@"♣"];
return validSuits;
}
First, we will only match a single other card (your homework assignment might provide possibilities to do better than this).
Stanford CS193p
Winter 2013

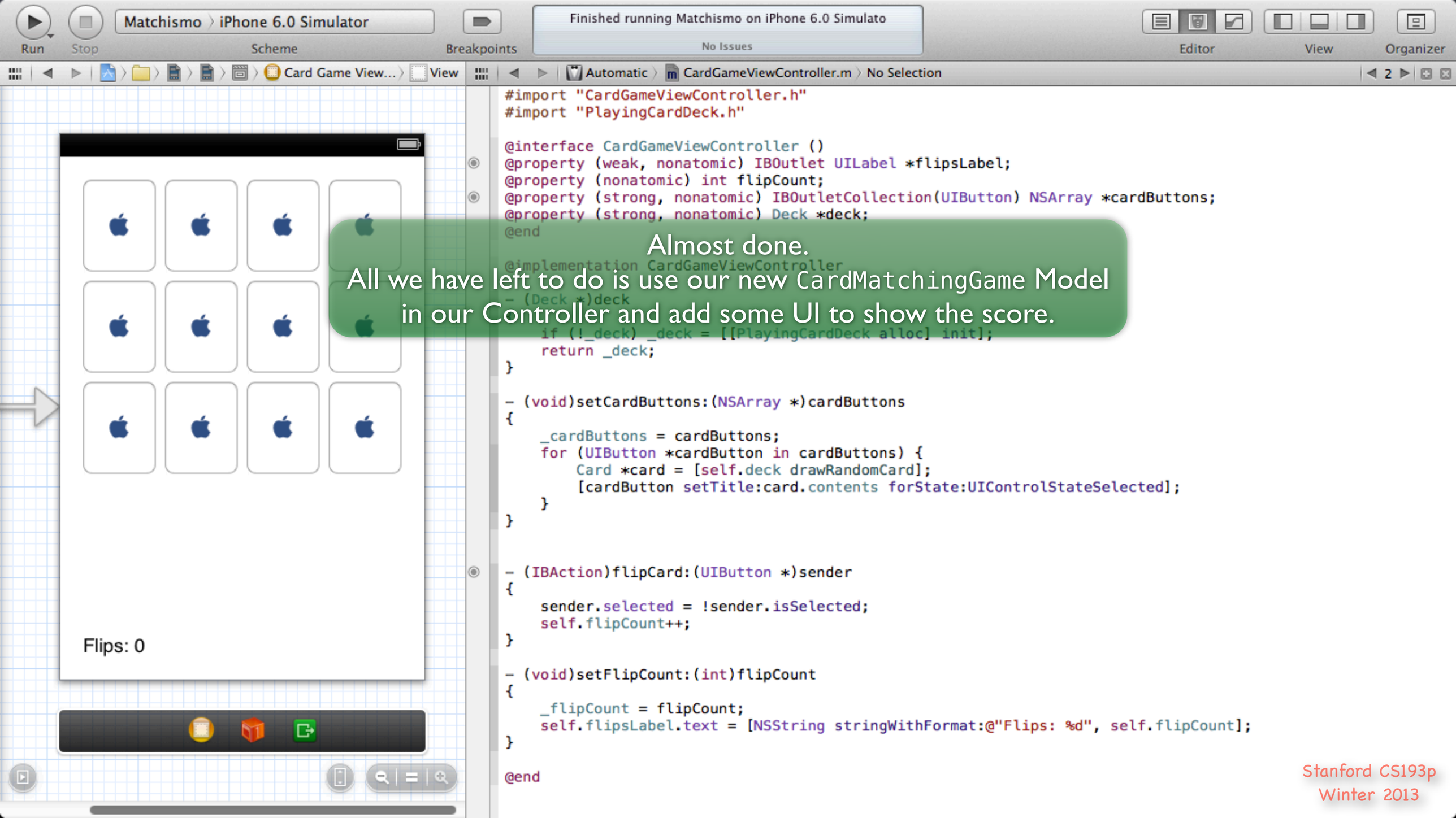

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
Matchi... 〉 PlayingCard.h 〉 No Selection
//
// PlayingCard.h
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "Card.h"
@interface PlayingCard : Card
@property (strong, nonatomic) NSString *suit;
@property (nonatomic) NSUInteger rank;
+ (NSArray *)validSuits;
+ (NSUInteger)maxRank;
@end
Let's get the card in the array
(there will only be one card in the array
if we got this far).
Counterparts 〉 PlayingCard.m 〉 -match:
//
// PlayingCard.m
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "PlayingCard.h"
@implementation PlayingCard
- (int)match:(NSArray *)otherCards
{
int score = 0;
if (otherCards.count == 1) {
PlayingCard *otherCard = [otherCards lastObject];
}
return score;
}
- (NSString *)contents
{
return [[PlayingCard rankStrings][self.rank] stringByAppendingString:
self.suit];
}
@synthesize suit = _suit;
+ (NSArray *)validSuits
{
static NSArray *validSuits = nil;
if (!validSuits) validSuits = @[@"♥",@"♦",@"♠",@"♣"];
return validSuits;
}
lastObject is an NSArray method. It is just like
[array objectAtIndex:array.count-1]
except that it will not crash if the array is empty
(it will just return nil).
Convenient.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
Matchi... 〉 PlayingCard.h 〉 No Selection
//
// PlayingCard.h
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "Card.h"
@interface PlayingCard : Card
@property (strong, nonatomic) NSString *suit;
@property (nonatomic) NSUInteger rank;
+ (NSArray *)validSuits;
+ (NSUInteger)maxRank;
@end
Counterparts 〉 PlayingCard.m 〉 -match:
//
// PlayingCard.m
// Lecture 1
//
// Created by CS193p Instructor on 1/8/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "PlayingCard.h"
@implementation PlayingCard
- (int)match:(NSArray *)otherCards
{
int score = 0;
if (otherCards.count == 1) {
PlayingCard *otherCard = [otherCards lastObject];
if ([otherCard.suit isEqualToString:self.suit]) {
score = 1;
} else if (otherCard.rank == self.rank) {
score = 4;
}
}
return score;
}
- (NSString *)contents
{
return [[PlayingCard rankStrings][self.rank] stringByAppendingString:
self.suit];
}
@synthesize suit = _suit;
+ (NSArray *)validSuits
{
static NSArray *validSuits = nil;
if (!validSuits) validSuits = @[@"♥",@"♦",@"♠",@"♣"];
return validSuits;
}
Give 4 times as many points for matching the rank than matching the suit (since there are only 3 cards that will match a given card's rank, but 12 which will match its suit).
You can probably imagine much better algorithms than this.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
Matc...
Card Game View Controller
View
Done. Back to our View ...
Flips: 0
Automatic
CardGameViewController.h
No Selection
2
//
// CardGameViewController.h
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface CardGameViewController : UIViewController
@end
Stanford CS193p
Winter 2013

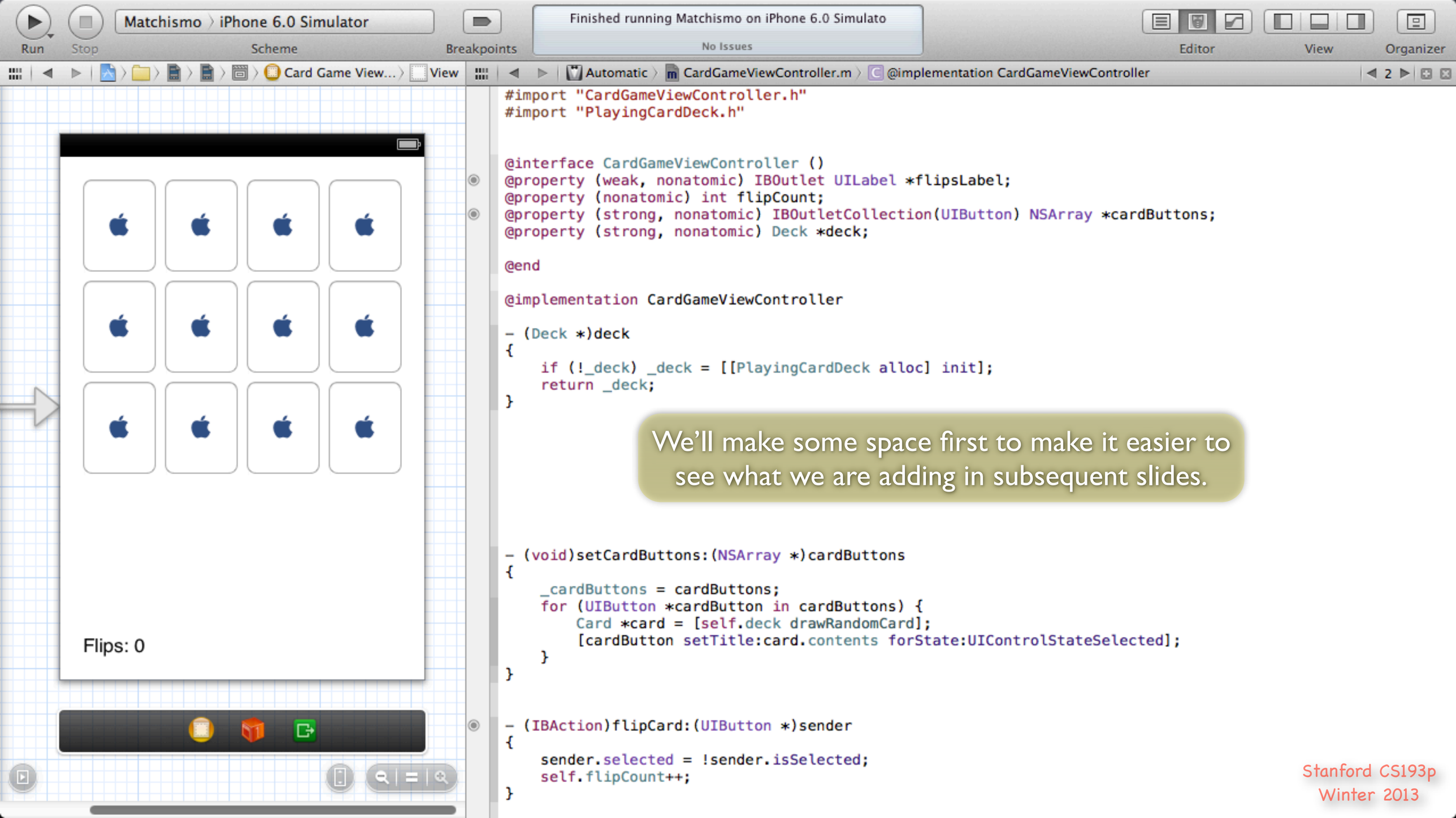

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
Matc...
Card Game View Controller
View
Flips: 0
Automati
CardGameViewController.h
CardGameViewController.m
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import <UIKit/UIKit.h>
@interface CardGameViewController : UIViewController
@end
... and Controller implementation.
Stanford CS193p
Winter 2013

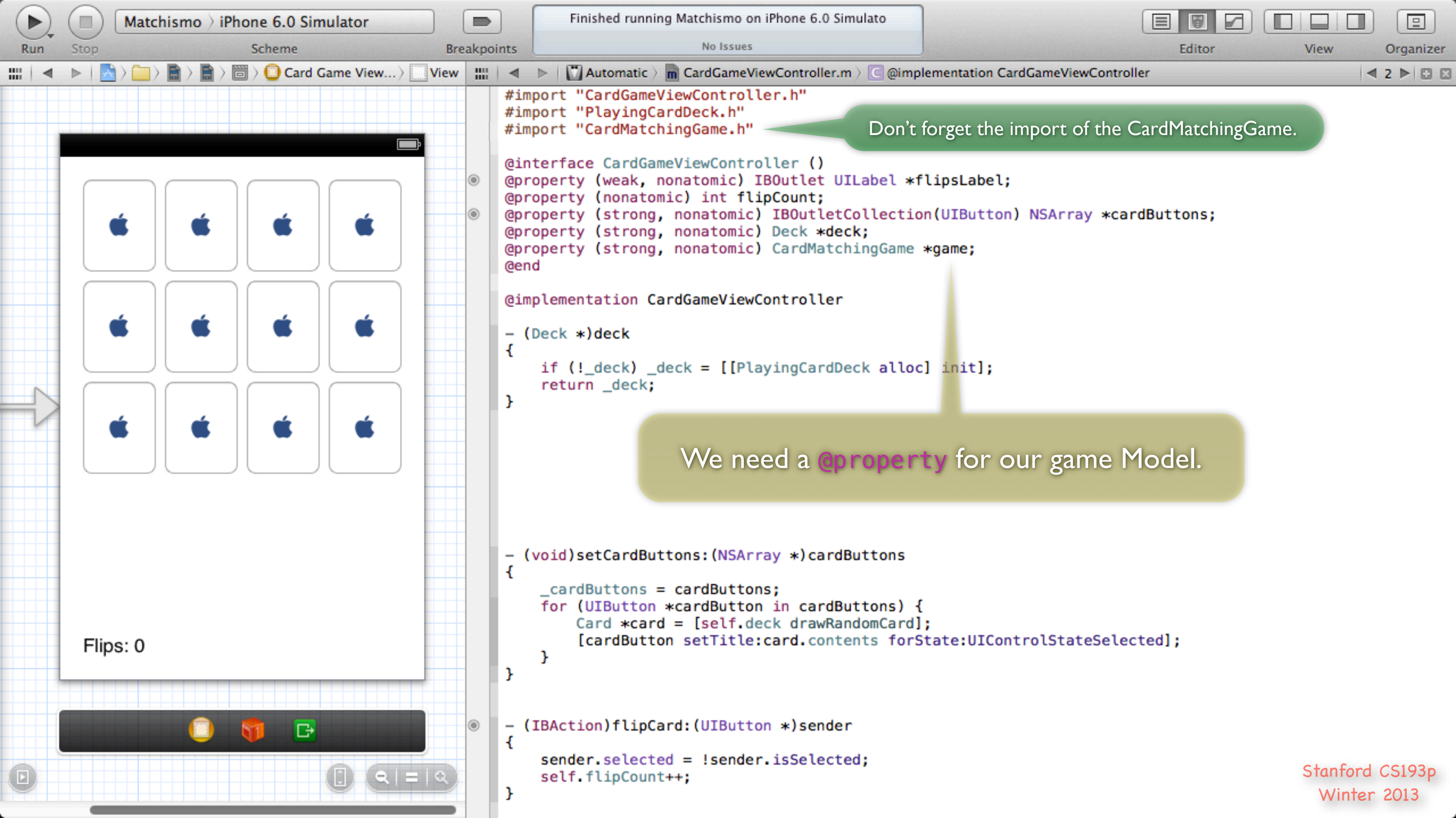

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Matchismo
1 target, iOS SDK 6.0
Matchismo
Model
CardMatchingGame.h
CardMatchingGame.m
Card.h
Card.m
Deck.h
Deck.m
PlayingCard.h
PlayingCard.m
PlayingCardDeck.h
PlayingCardDeck.m
CardGameAppDelegate.h
CardGameAppDelegate.m
MainStoryboard.storyboard
CardGameViewController.h
CardGameViewController.m
Supporting Files
Frameworks
Products
Matc...
Card Game View Controller
View
Flips: 0
Automatic
CardGameViewController.m
No Selection
2
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton)
NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButtons) {
Card *card = [self.deck drawRandomCard];
[cardButton setTitle:card.contents forState:
UIControlStateSelected];
}
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
- (void)setFlipCount:(int)flipCount
{
_flipCount = flipCount;
self.flipsLabel.text = [NSString
stringWithFormat:@"Flips: %d", self.flipCount];
}
No need for Navigator.
Stanford CS193p
Winter 2013

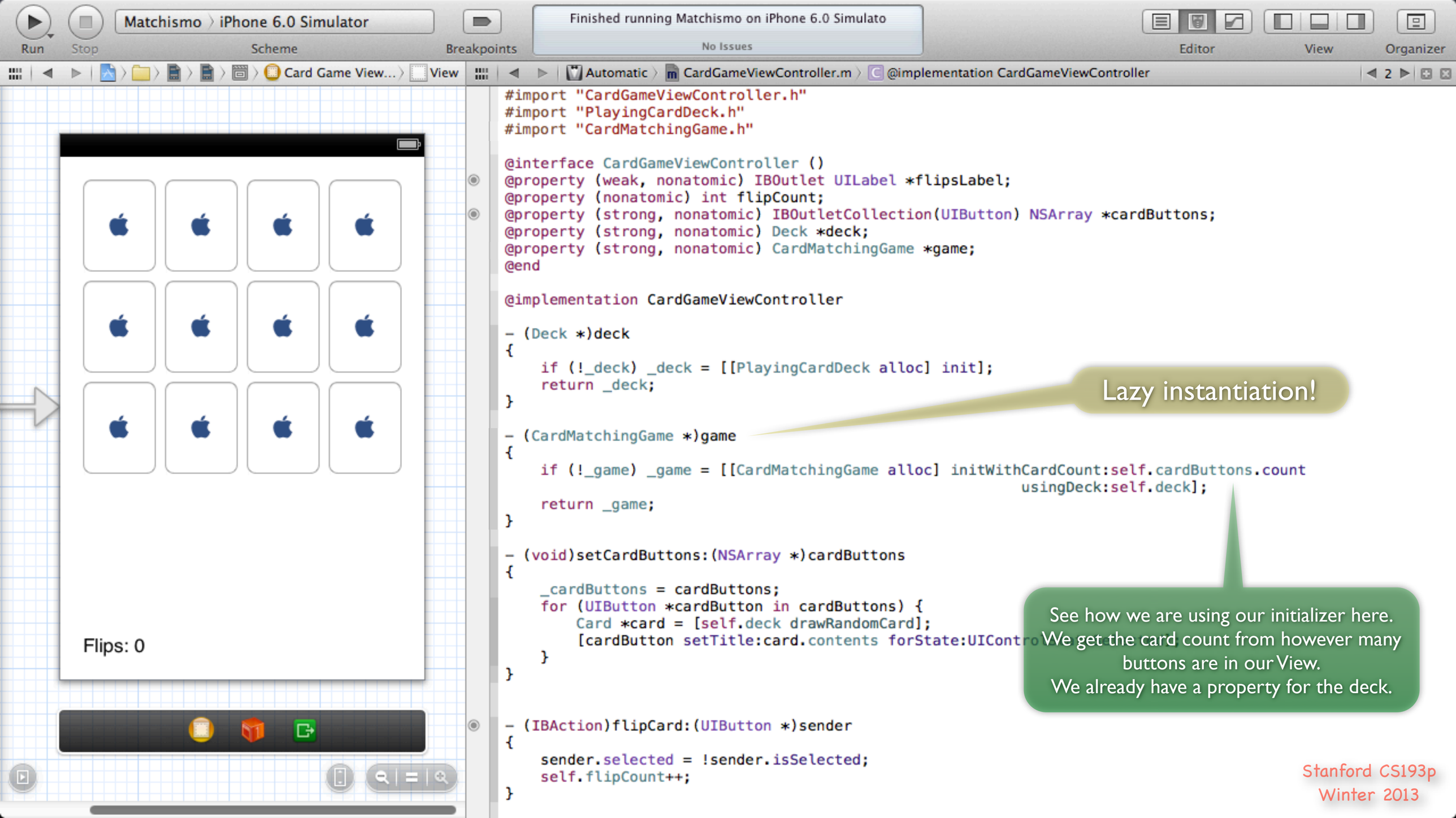


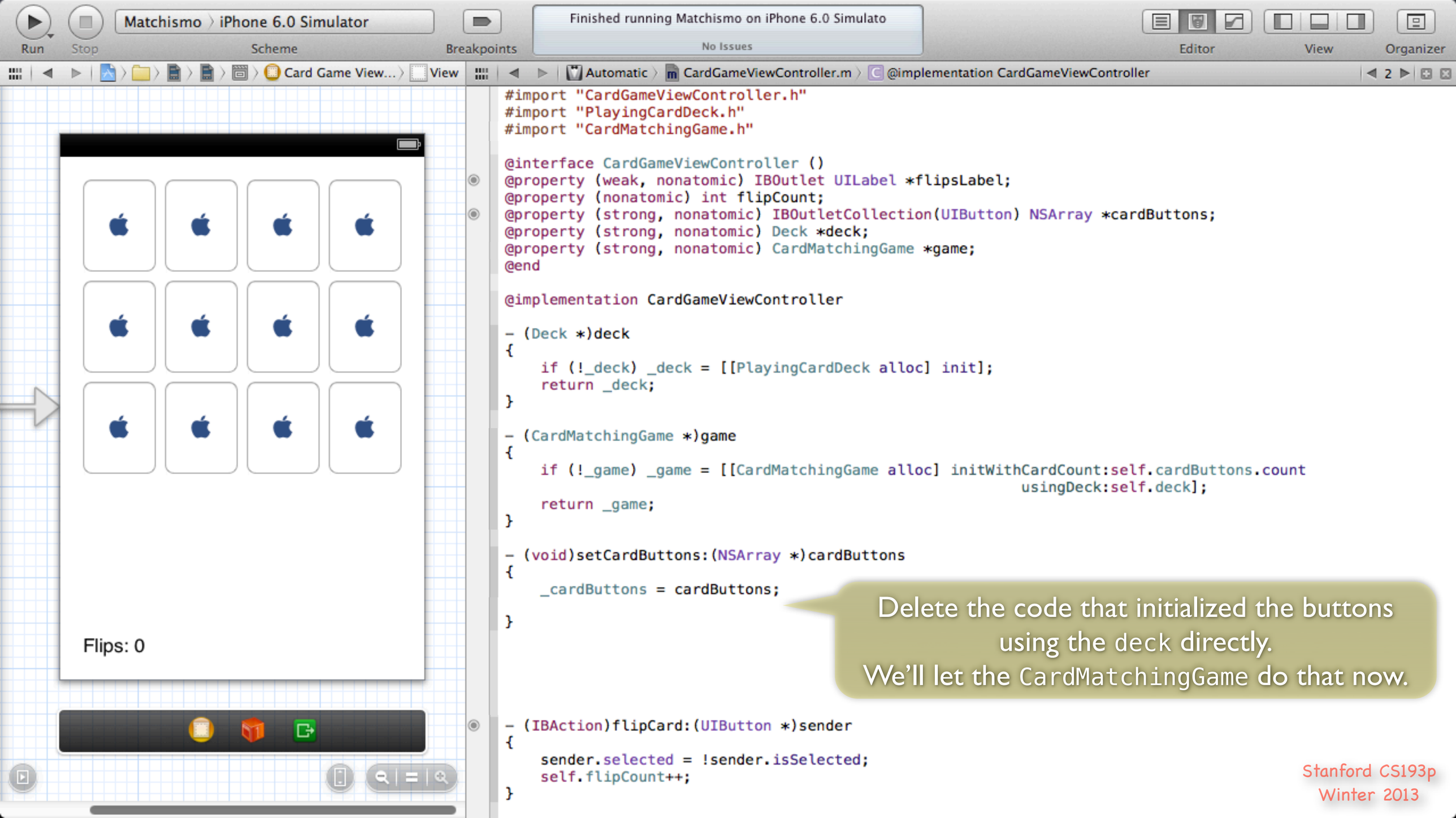
Almost done.

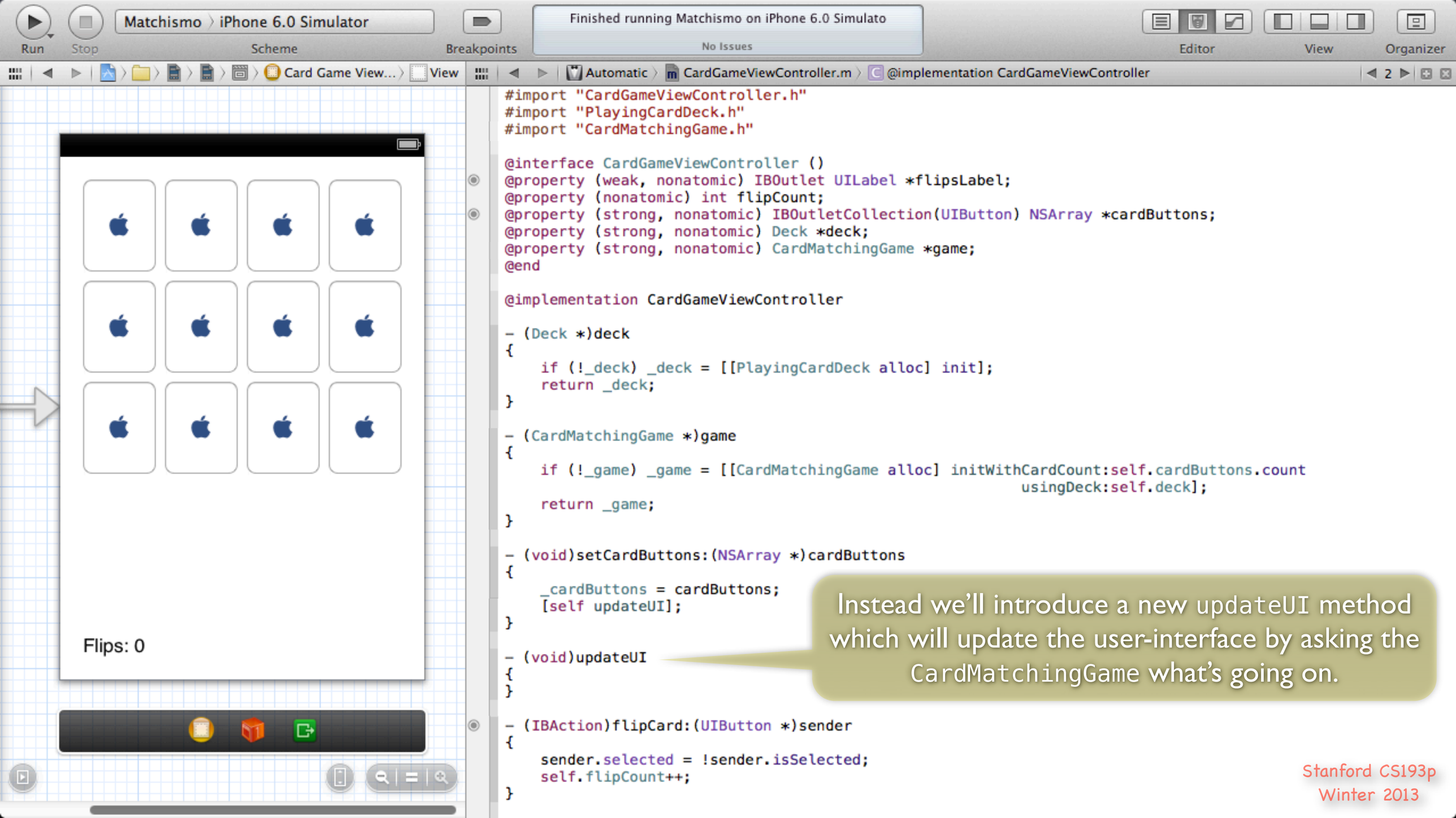
All we have left to do is use our new CardMatchingGame Model in our Controller and add some UI to show the score.

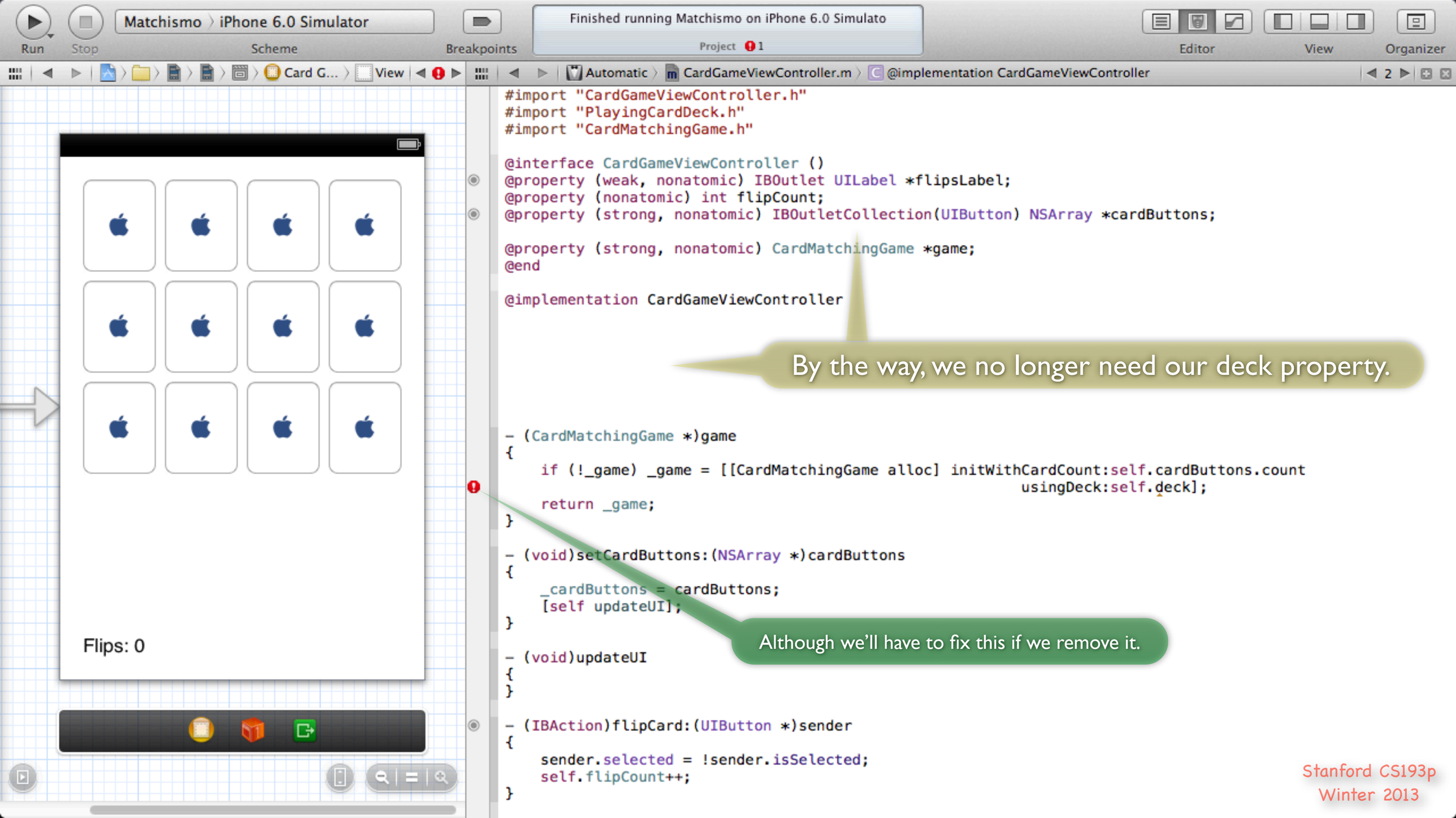

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 @implementation CardGameViewController
2
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
We'll make some space first to make it easier to see what we are adding in subsequent slides.
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButtons) {
Card *card = [self.deck drawRandomCard];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
}
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Flips: 0
Stanford CS193p
Winter 2013

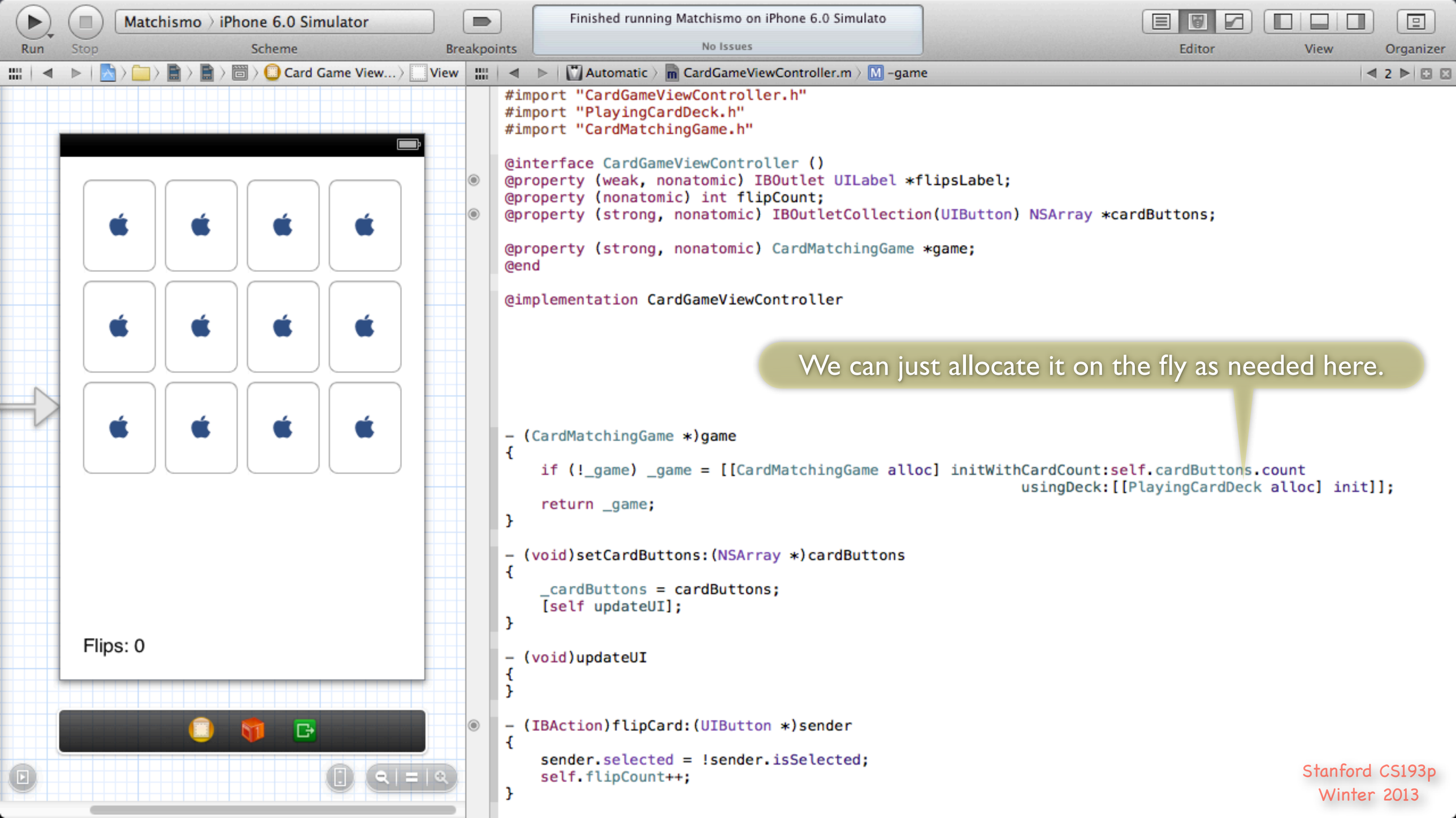

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
Don't forget the import of the CardMatchingGame.
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
We need a @property for our game Model.
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButtons) {
Card *card = [self.deck drawRandomCard];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
}
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Stanford CS193p
Winter 2013

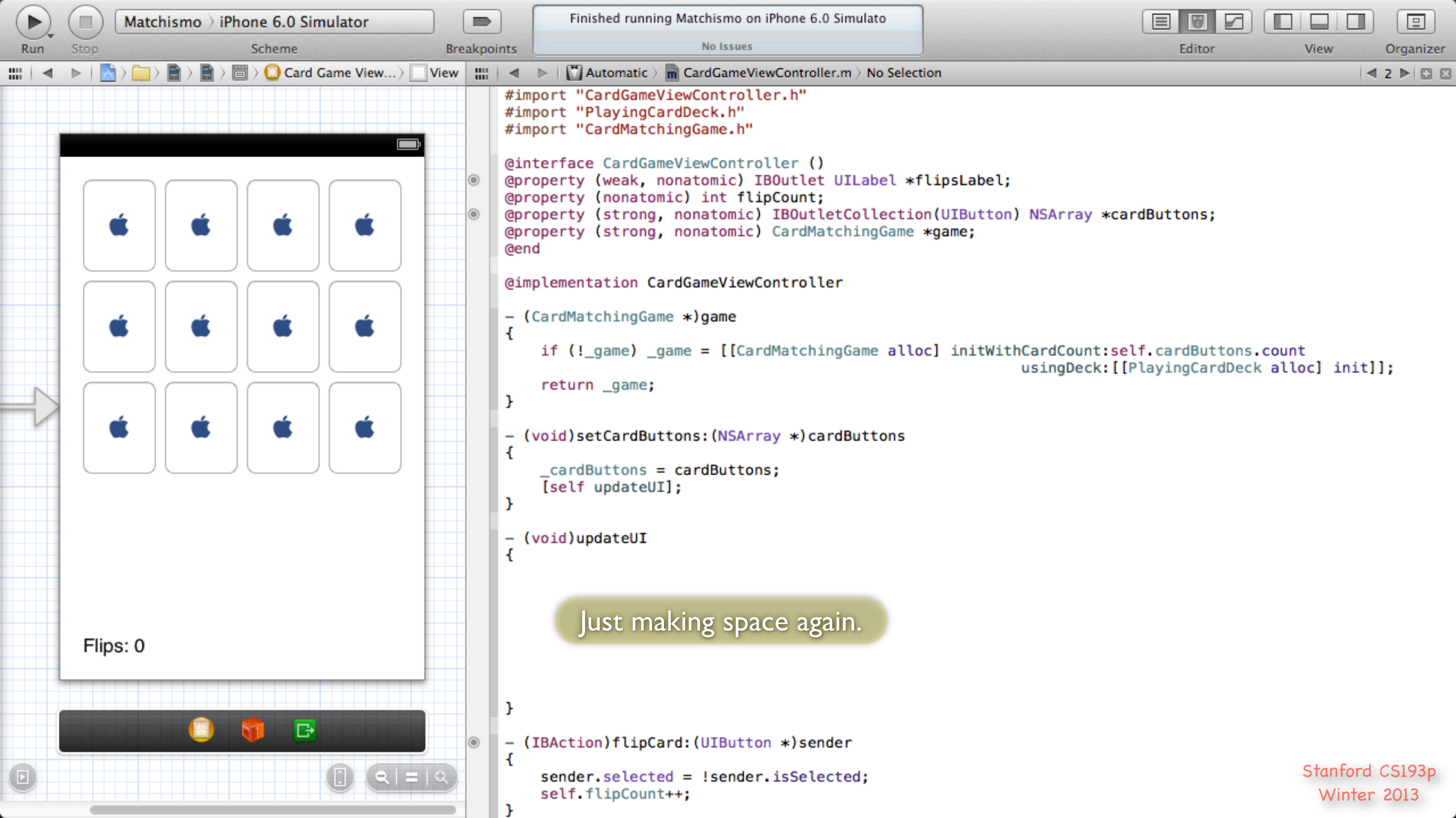

Matchismo 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:self.deck];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
for (UIButton *cardButton in cardButtons) {
Card *card = [self.deck drawRandomCard];
[cardButton setTitle:card.contents forState:UIContro
}
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Lazy instantiation!
See how we are using our initializer here.
We get the card count from however many buttons are in our View.
We already have a property for the deck.
Stanford CS193p
Winter 2013

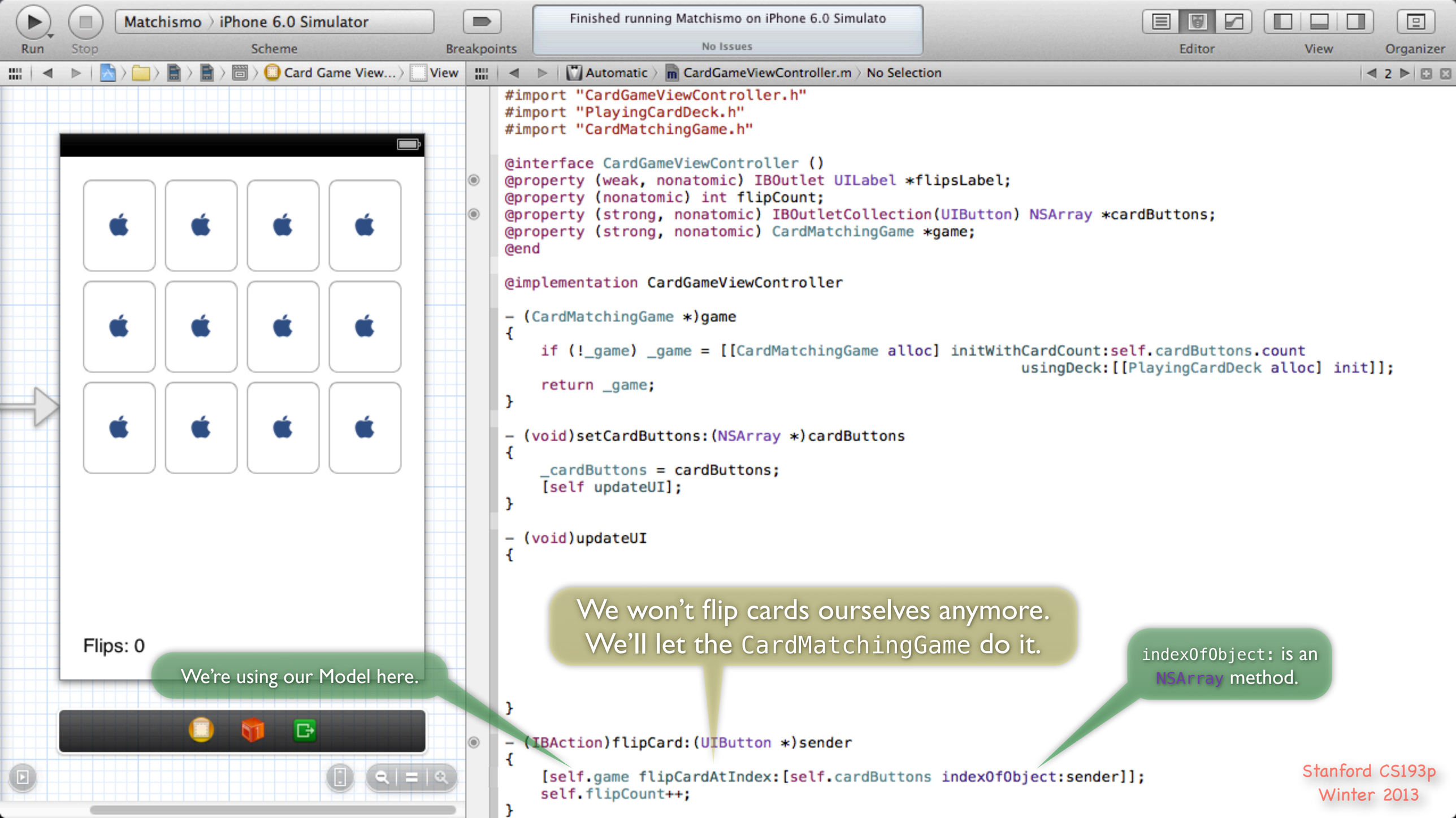

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 @implementation CardGameViewController
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:self.deck];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
}
Delete the code that initialized the buttons using the deck directly.
We'll let the CardMatchingGame do that now.
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Stanford CS193p
Winter 2013

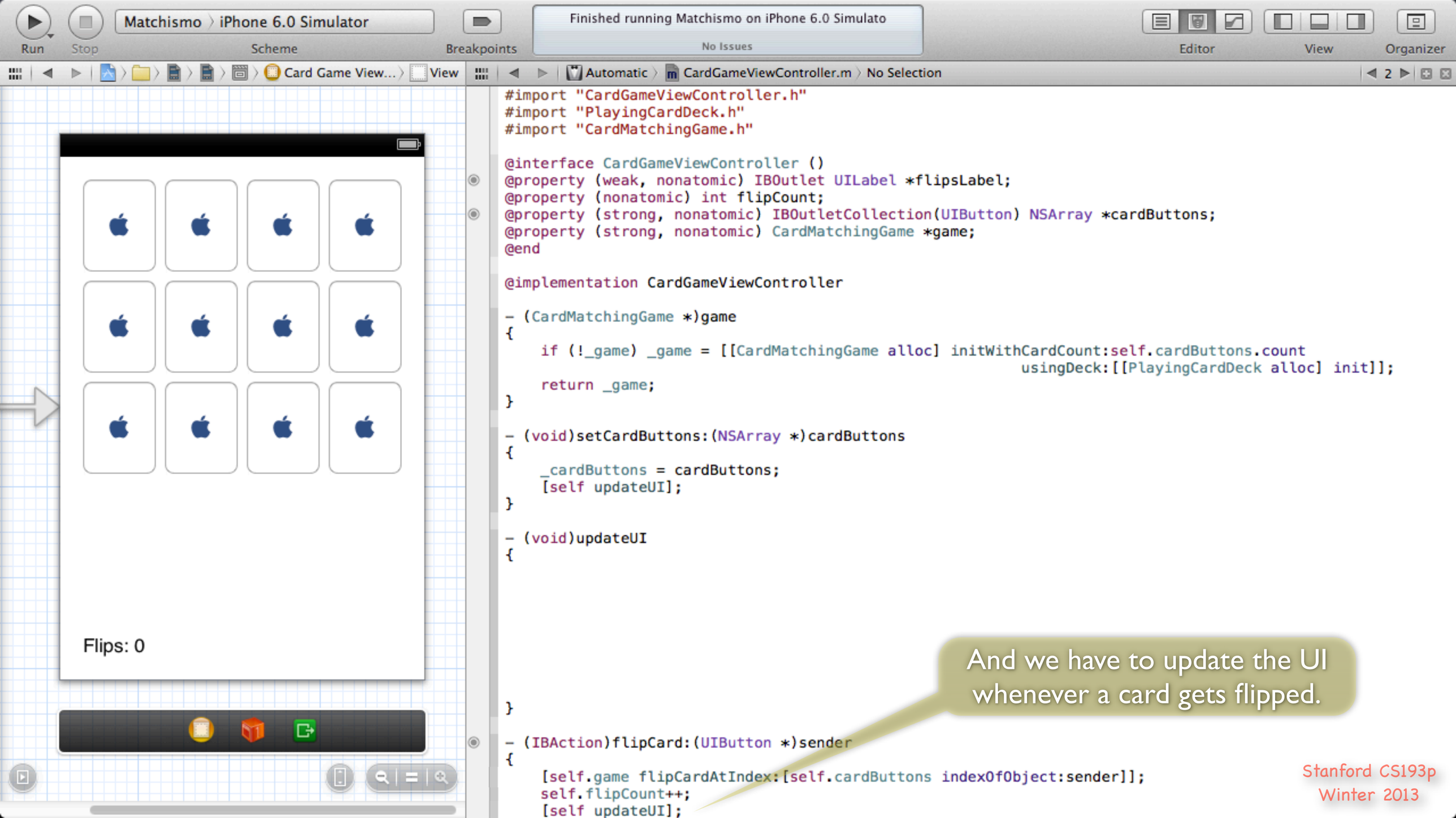

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
2
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) Deck *deck;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (Deck *)deck
{
if (!_deck) _deck = [[PlayingCardDeck alloc] init];
return _deck;
}
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:self.deck];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Instead we'll introduce a new updateUI method which will update the user-interface by asking the CardMatchingGame what's going on.
Stanford CS193p
Winter 2013

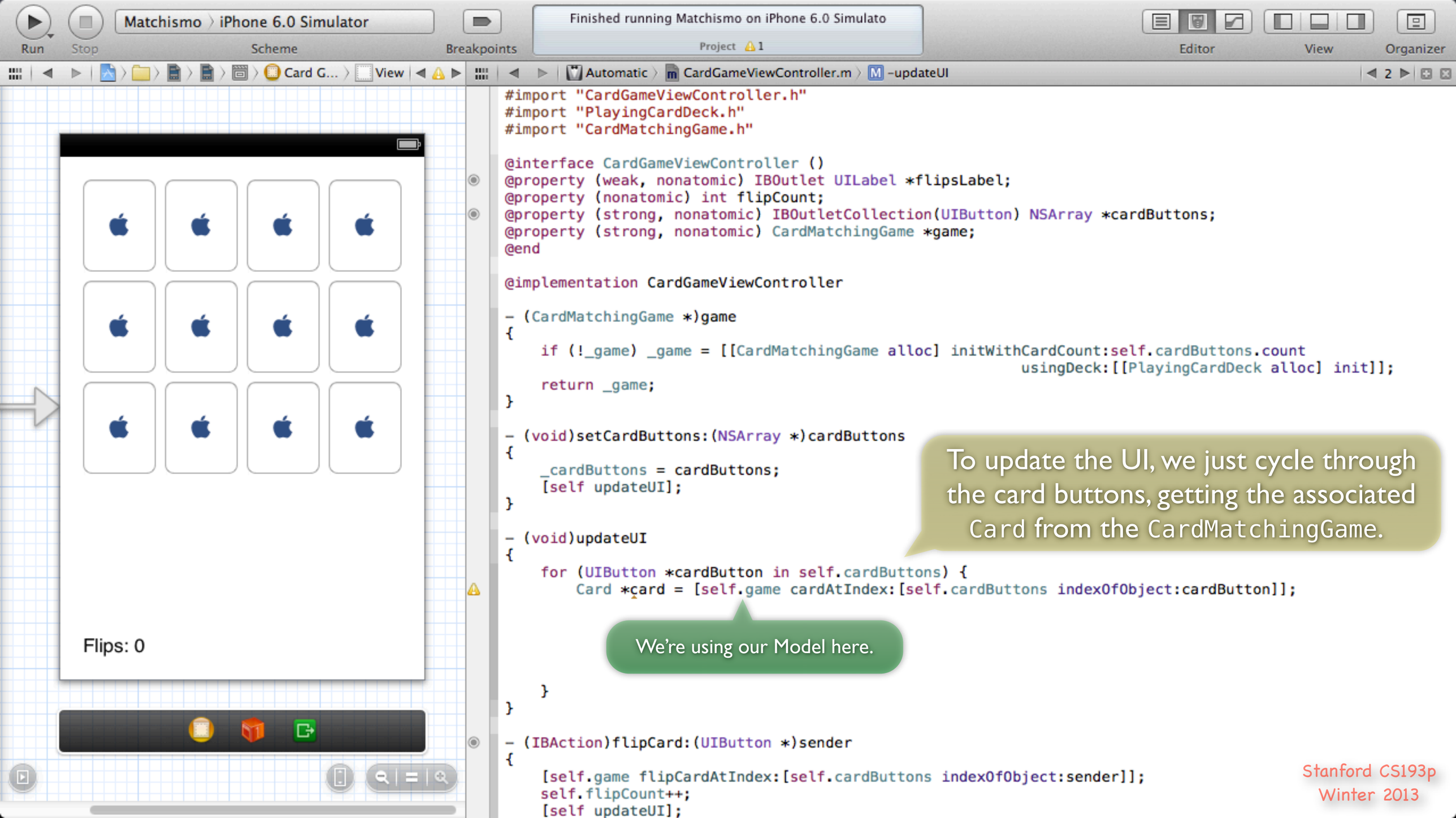

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
Card G...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
2
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:self.deck];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Flips: 0
By the way, we no longer need our deck property.
Although we'll have to fix this if we remove it.
Stanford CS193p
Winter 2013

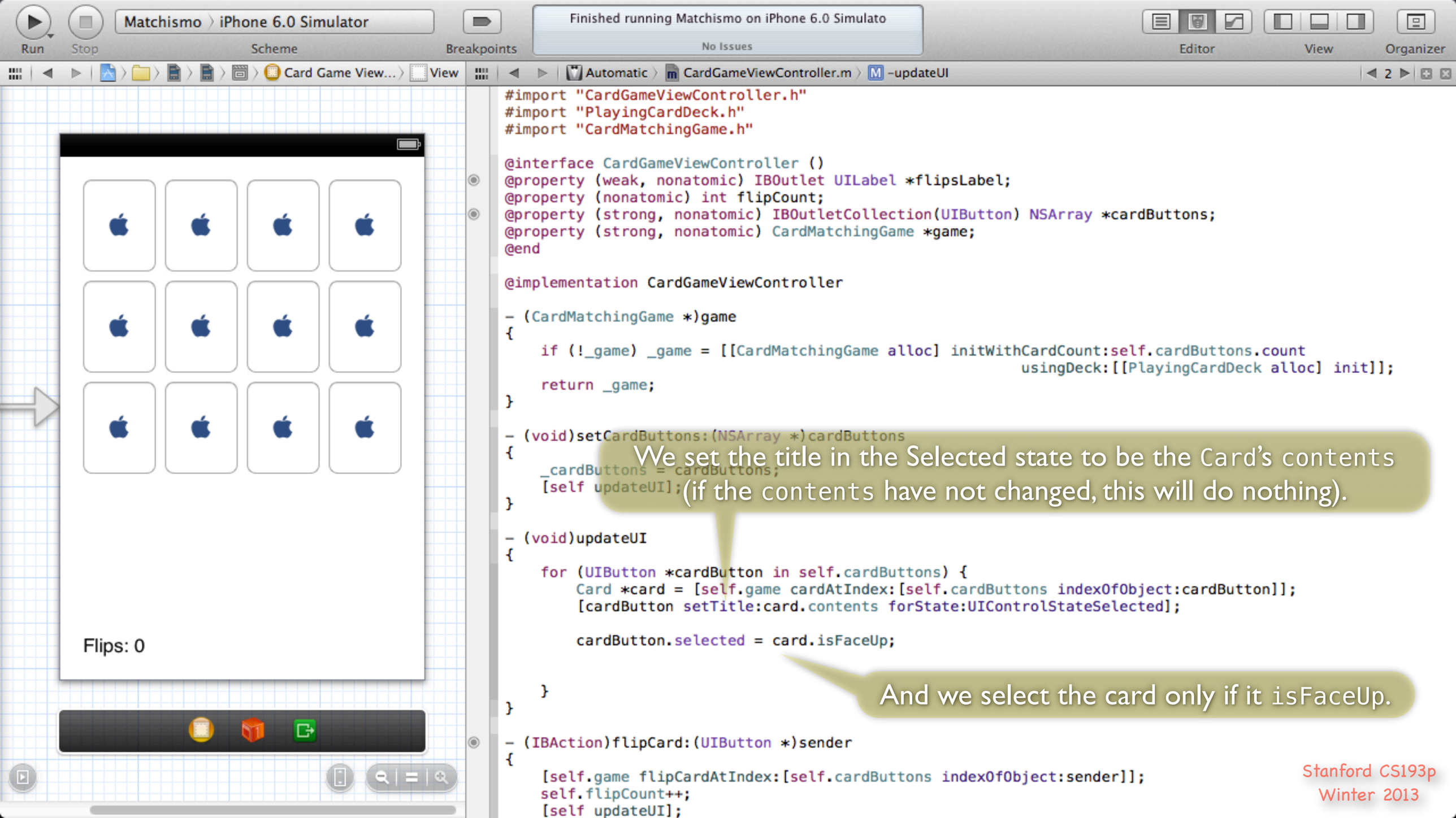

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 -game
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
We can just allocate it on the fly as needed here.
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Flips: 0
Stanford CS193p
Winter 2013

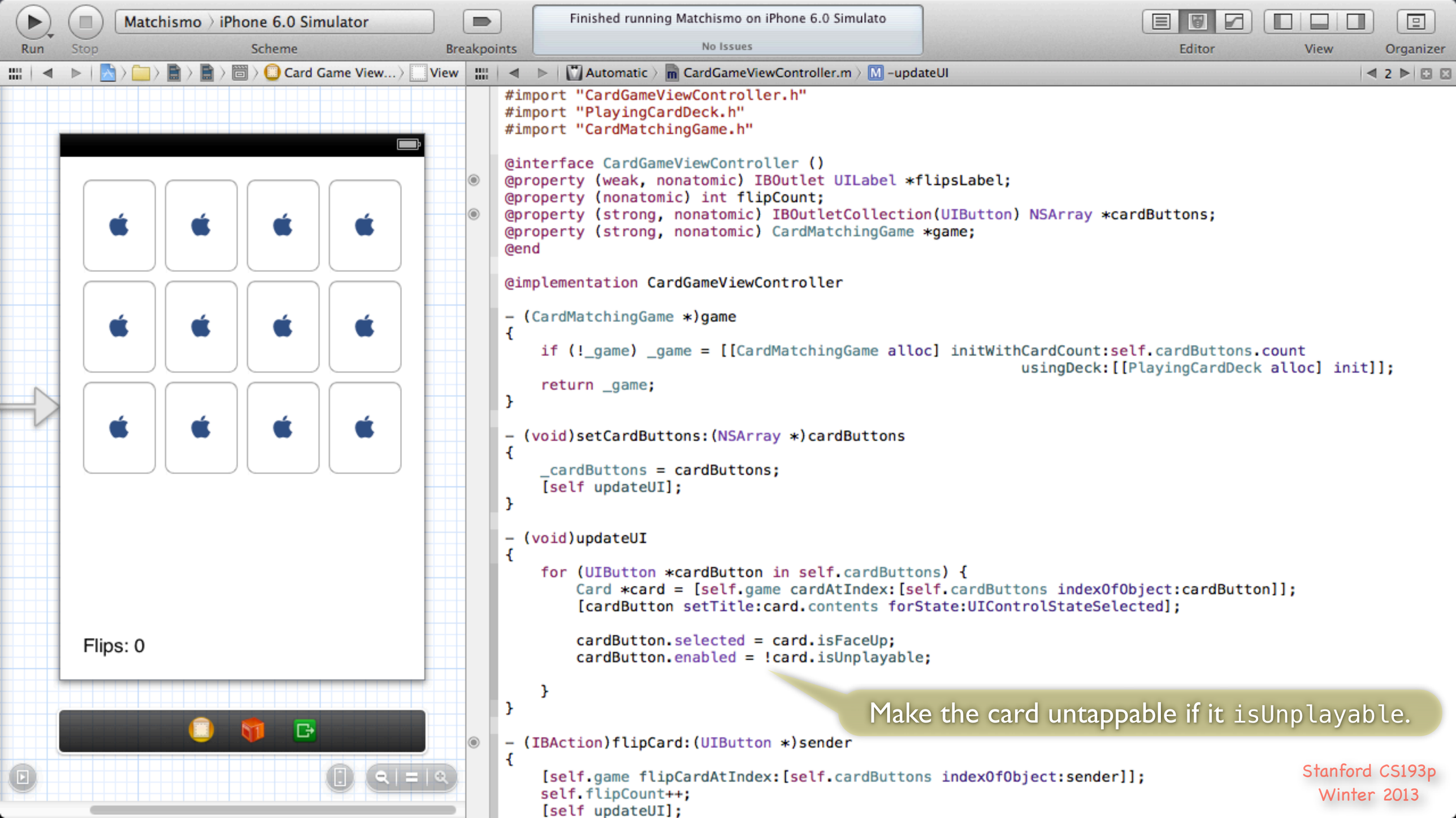

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
No Selection
2
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
Just making space again.
}
- (IBAction)flipCard:(UIButton *)sender
{
sender.selected = !sender.isSelected;
self.flipCount++;
}
Stanford CS193p
Winter 2013

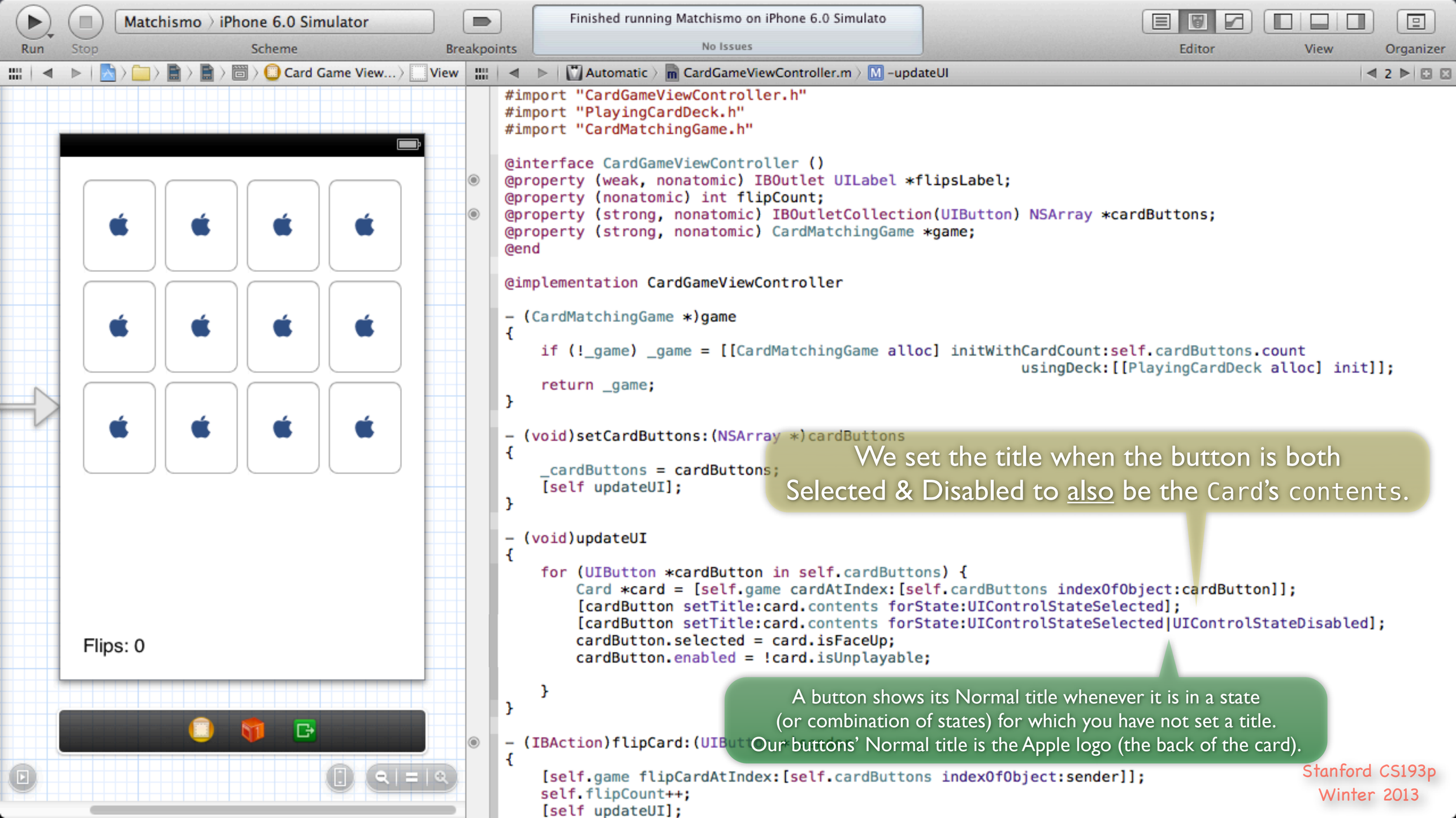

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 No Selection
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
}
Flips: 0
We're using our Model here.
We won't flip cards ourselves anymore.
We'll let the CardMatchingGame do it.
indexOfObject: is an
NSArray method.
Stanford CS193p
Winter 2013

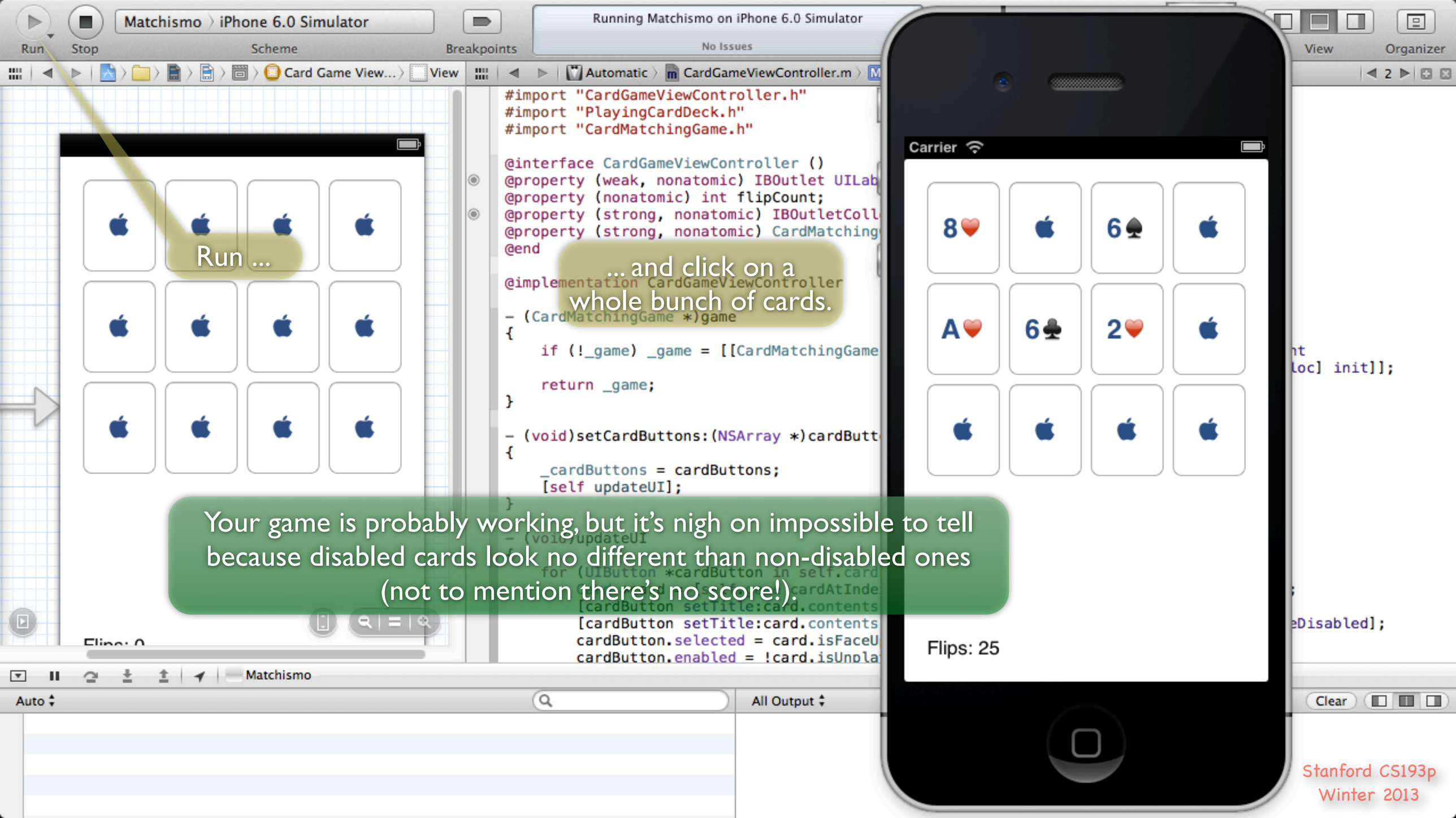

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 No Selection
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
And we have to update the UI whenever a card gets flipped.
Stanford CS193p
Winter 2013

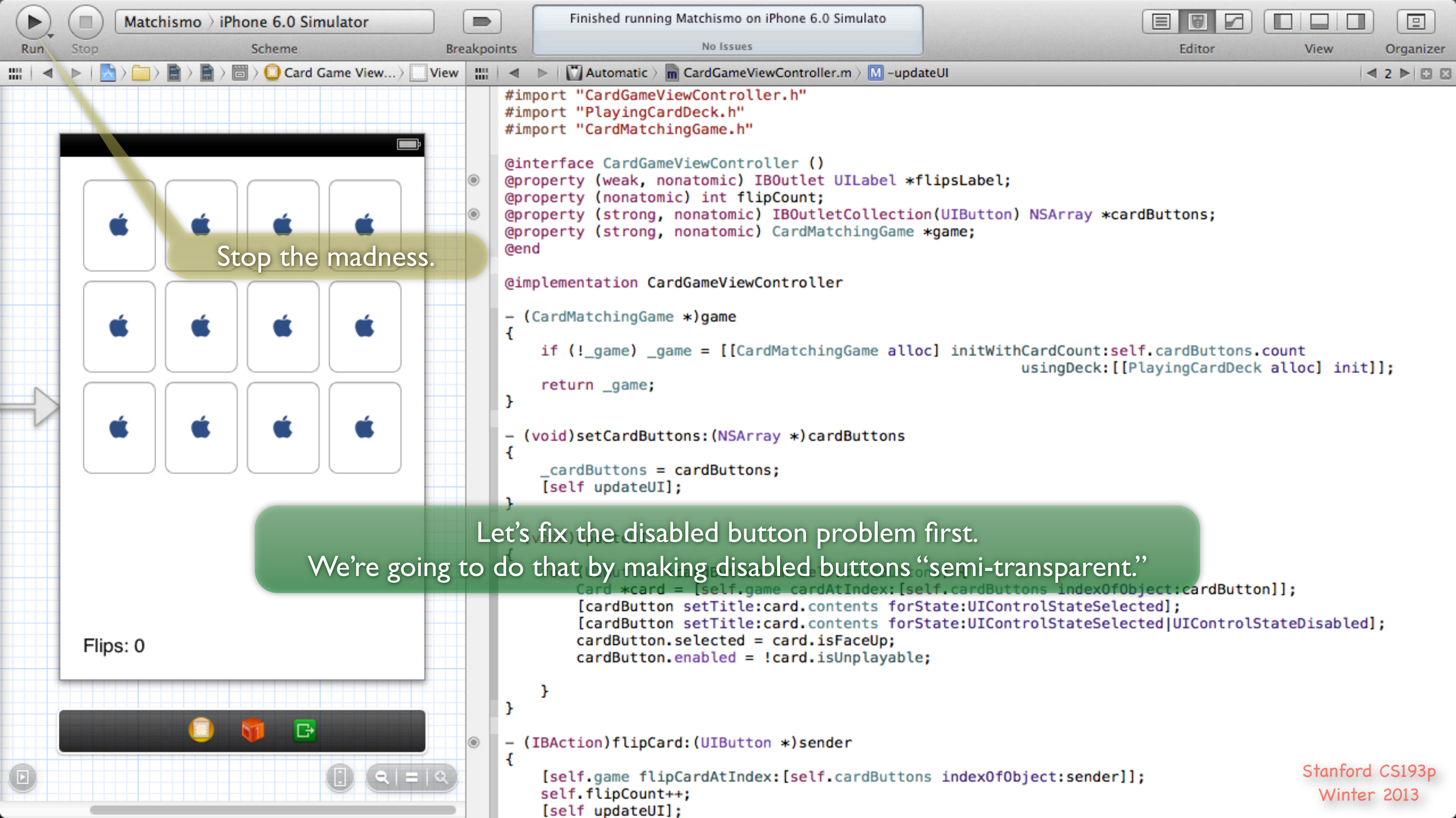

Matchismo 〉 iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
Project 1
Editor
View
Organizer
Card G...
View
Automatic
CardGameViewController.m
-updateUI
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
Flips: 0
To update the UI, we just cycle through the card buttons, getting the associated Card from the CardMatchingGame.
We're using our Model here.
Stanford CS193p
Winter 2013

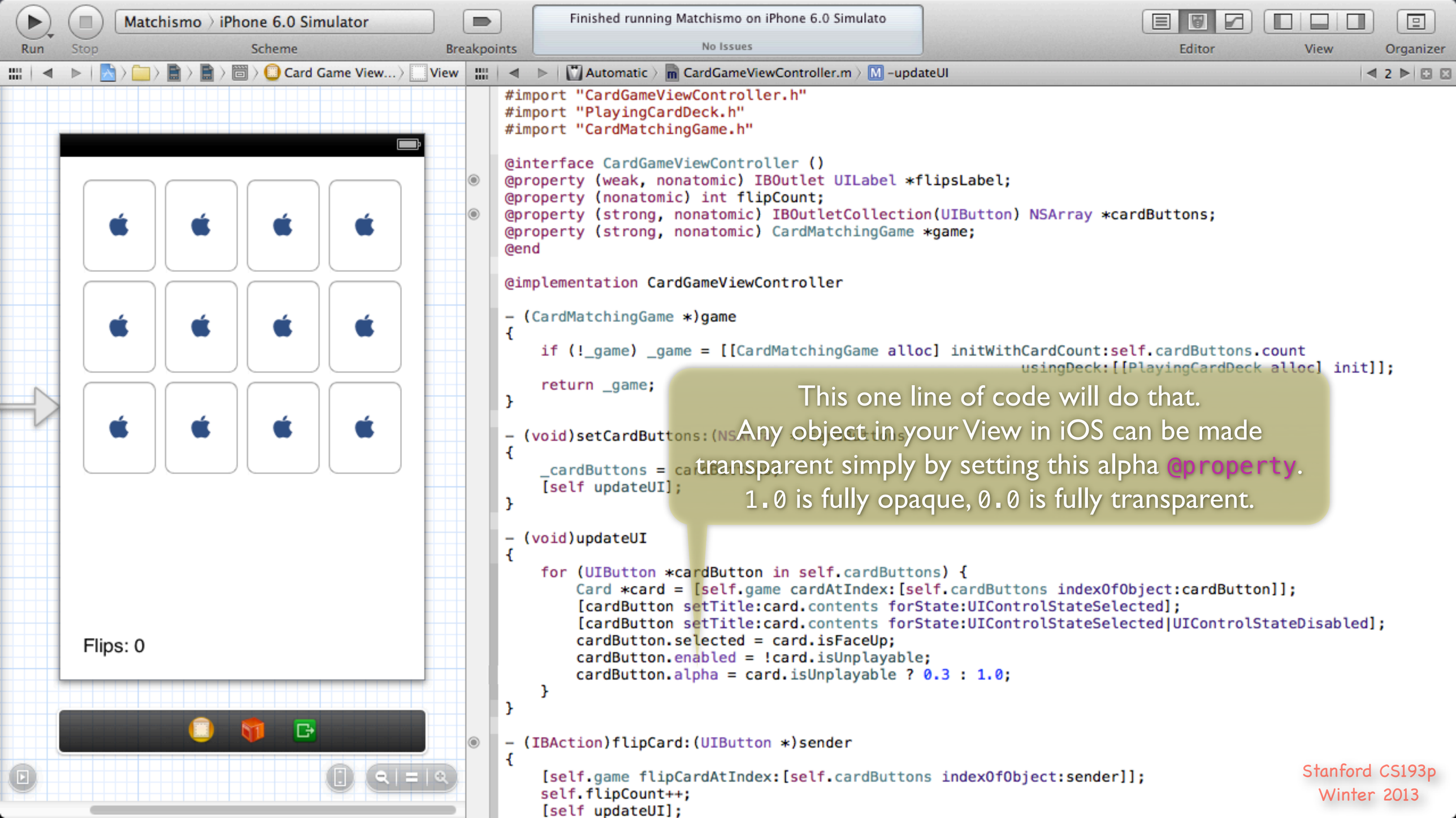

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
-updateUI
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
cardButton.selected = card.isFaceUp;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
We set the title in the Selected state to be the Card's contents (if the contents have not changed, this will do nothing).
And we select the card only if it isFaceUp.
Stanford CS193p
Winter 2013

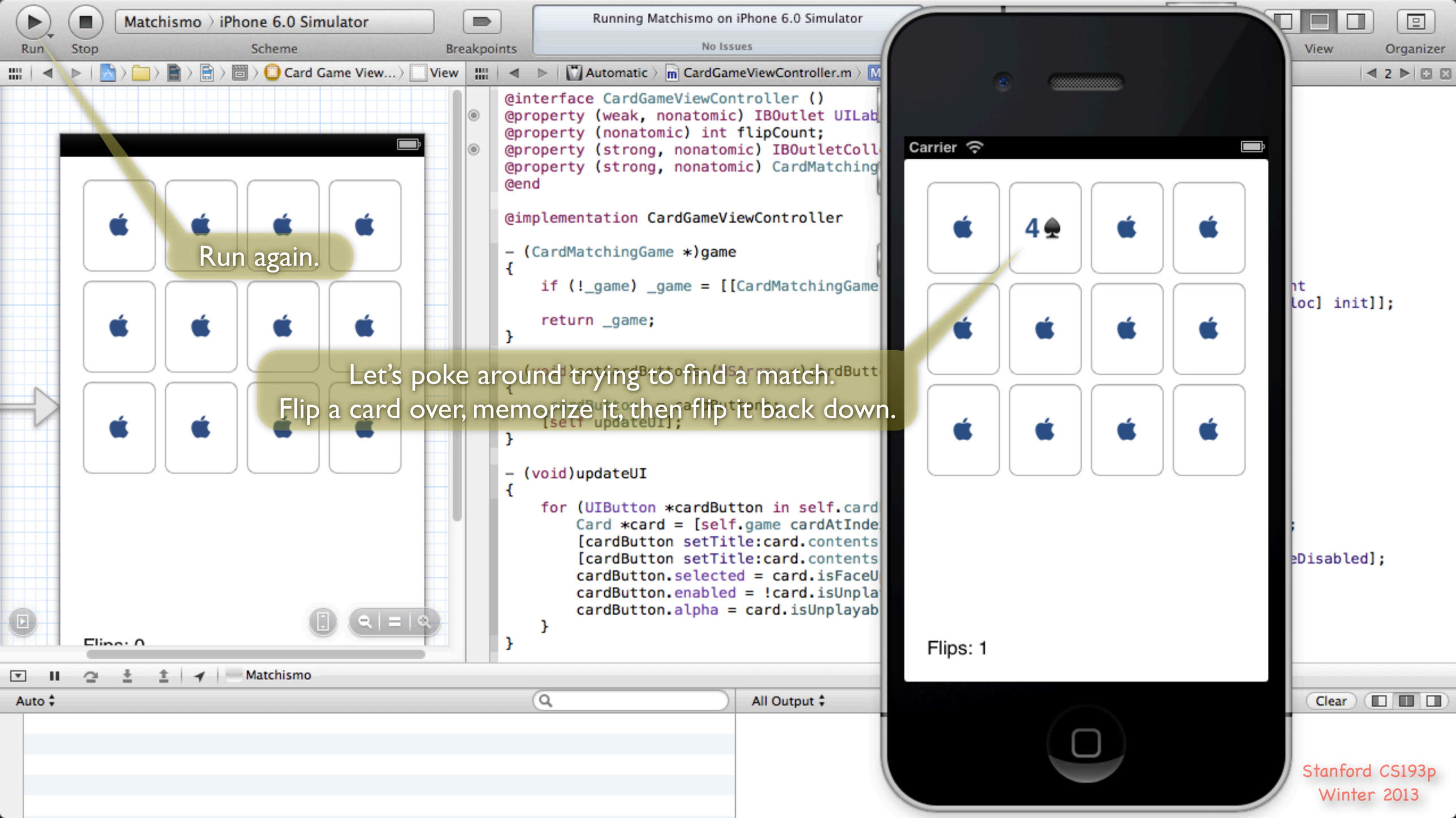

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
-updateUI
2
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
Flips: 0
Make the card untappable if it isUnplayable.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 -updateUI
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
}
}
- (IBAction)flipCard:(UIBut
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
Flips: 0
We set the title when the button is both Selected & Disabled to also be the Card's contents.
A button shows its Normal title whenever it is in a state (or combination of states) for which you have not set a title. Our buttons' Normal title is the Apple logo (the back of the card).
Stanford CS193p
Winter 2013


Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
for (UIButton *cardButton in self.card
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
Run ...
... and click on a whole bunch of cards.
Your game is probably working, but it's nigh on impossible to tell because disabled cards look no different than non-disabled ones (not to mention there's no score!).
Flips: 0
Carrier
8♥
6♠
A♥
6♣
2♥
Flips: 25
Matchismo
Auto
All Output
Clear
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 -updateUI
Stop the madness.
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
Let's fix the disabled button problem first.
We're going to do that by making disabled buttons "semi-transparent."
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
-updateUI
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
This one line of code will do that.
Any object in your View in iOS can be made transparent simply by setting this alpha @property.
1.0 is fully opaque, 0.0 is fully transparent.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
Run again.
Let's poke around trying to find a match.
Flip a card over, memorize it, then flip it back down.
Carrier
4♠
Flips: 1
Matchismo
Auto
All Output
Clear
Stanford CS193p
Winter 2013

Run Stop Matchismo › iPhone 6.0 Simulator Scheme Breakpoints

Running Matchismo on iPhone 6.0 Simulator

No Issues

View Organizer

Card Game View... › View

Automatic › CardGameViewController.m

```
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end

@implementation CardGameViewController

- (CardMatchingGame *)game
{
    if (!_game) _game = [[CardMatchingGame

    return _game;
}

- (void)setC
{
    _cardButtons = cardButtons;
    [self updateUI];
}

- (void)updateUI
{
    for (UIButton *cardButton in self.card
        Card *card = [self.game cardAtInde
        [cardButton setTitle:card.contents
        [cardButton setTitle:card.contents
        cardButton.selected = card.isFaceU
        cardButton.enabled = !card.isUnpla
        cardButton.alpha = card.isUnplayab
    }
}
```

4♠ flipped back down:

Carrier

Flips: 2

Flips: 0

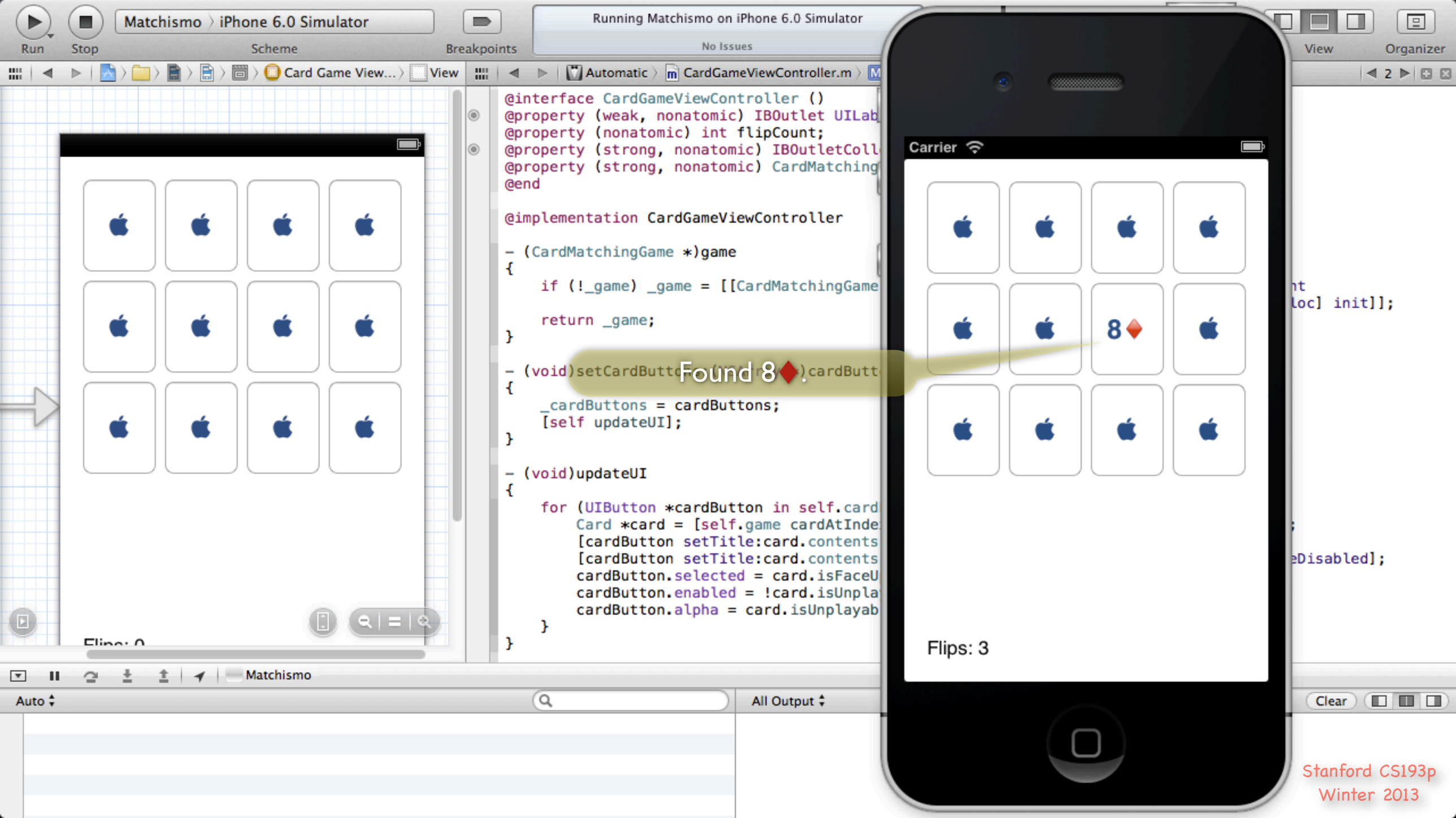
Matchismo

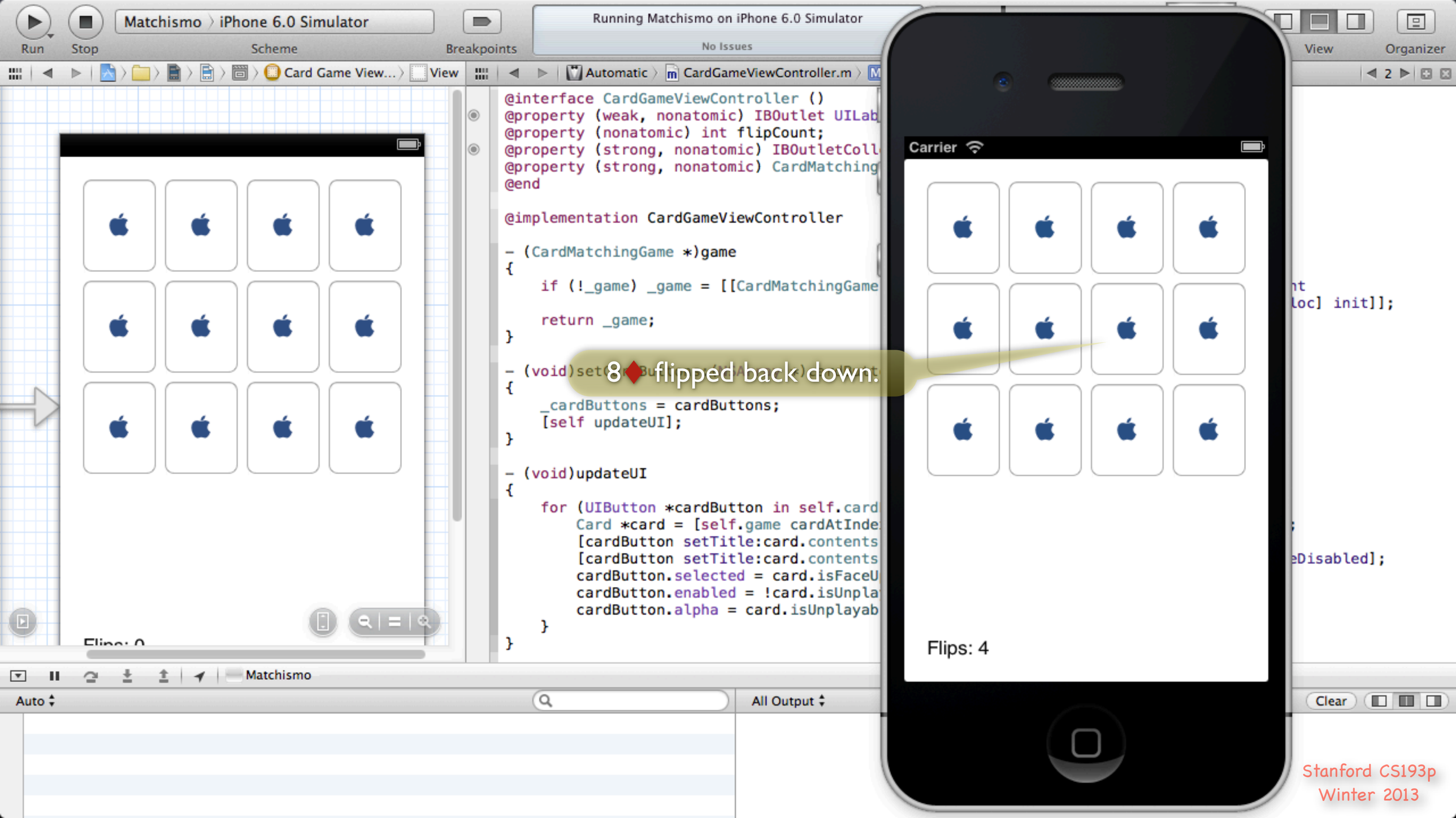
Auto

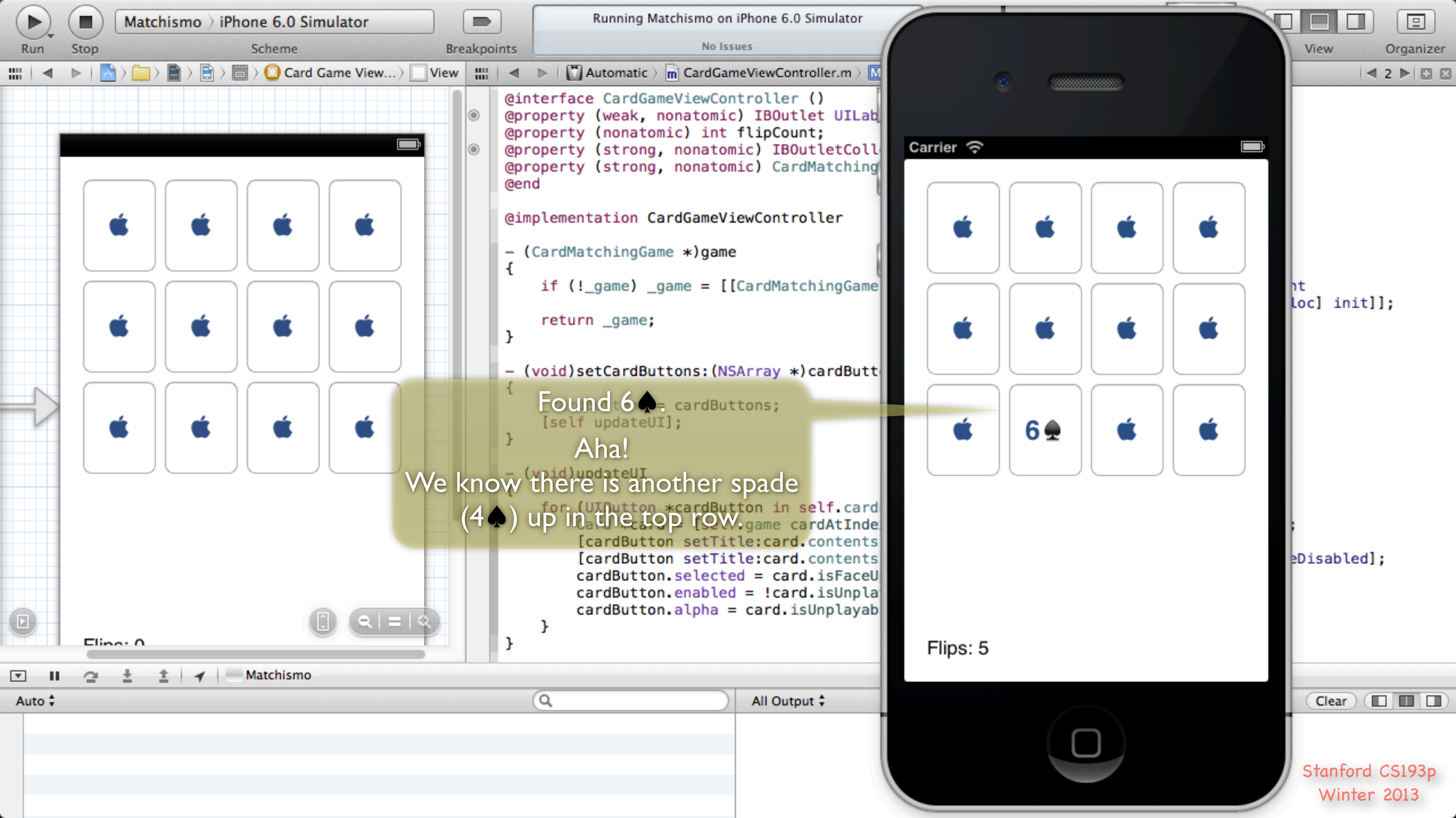
All Output

Clear

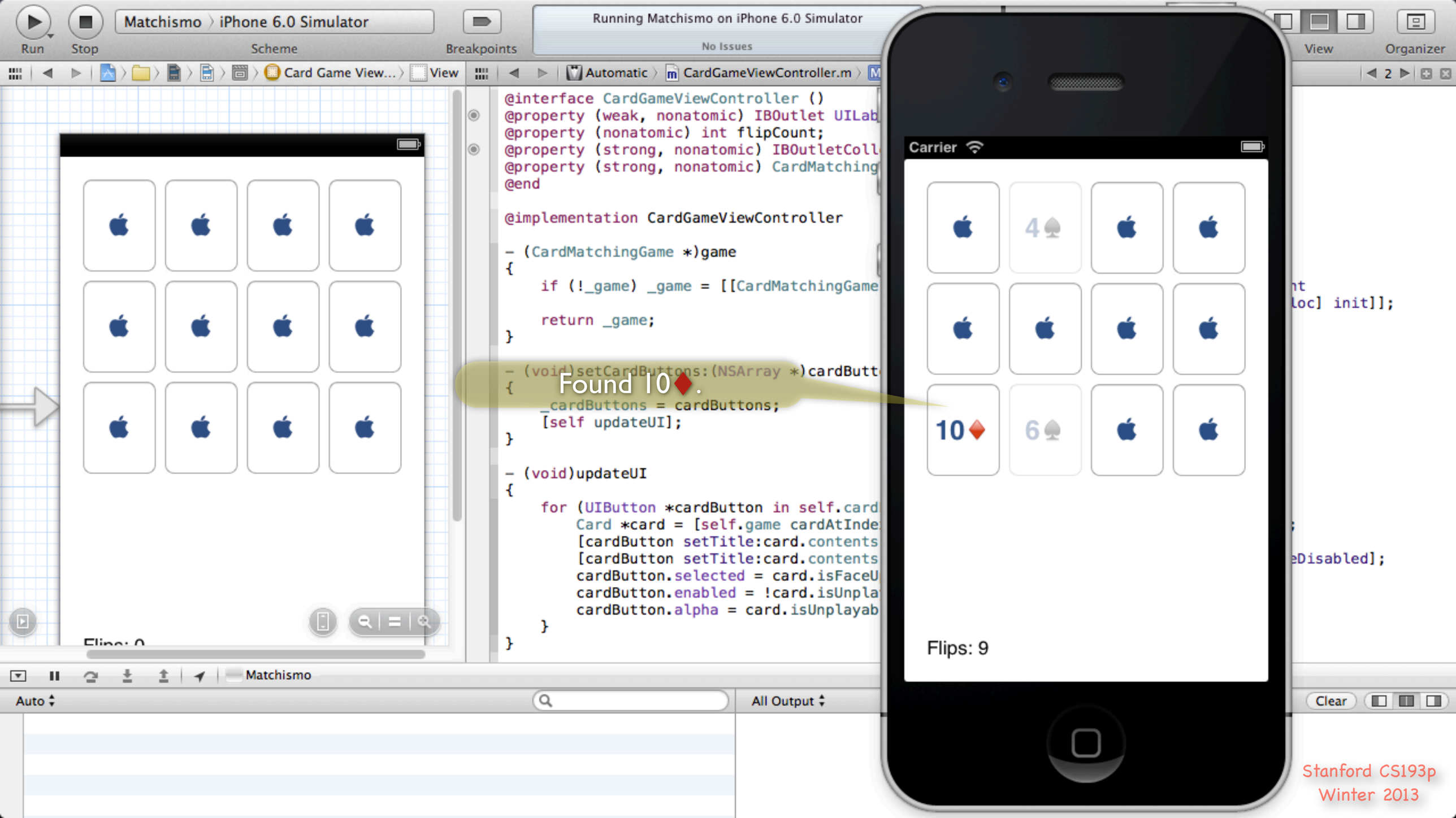
Stanford CS193p
Winter 2013

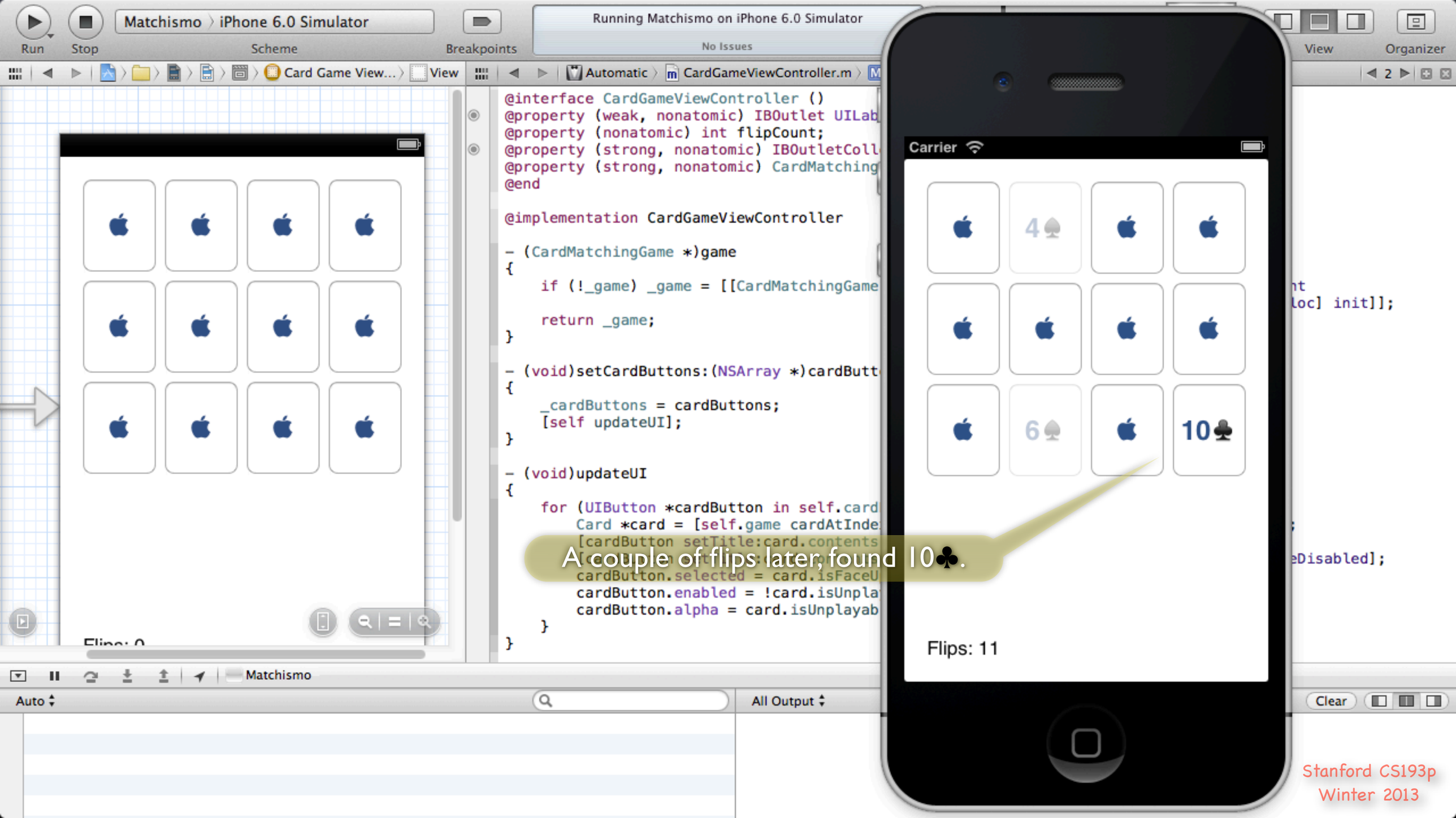

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
Found 8♦.
Carrier
8♦
Flips: 3
Matchismo
Auto
All Output
Clear
Stanford CS193p
Winter 2013

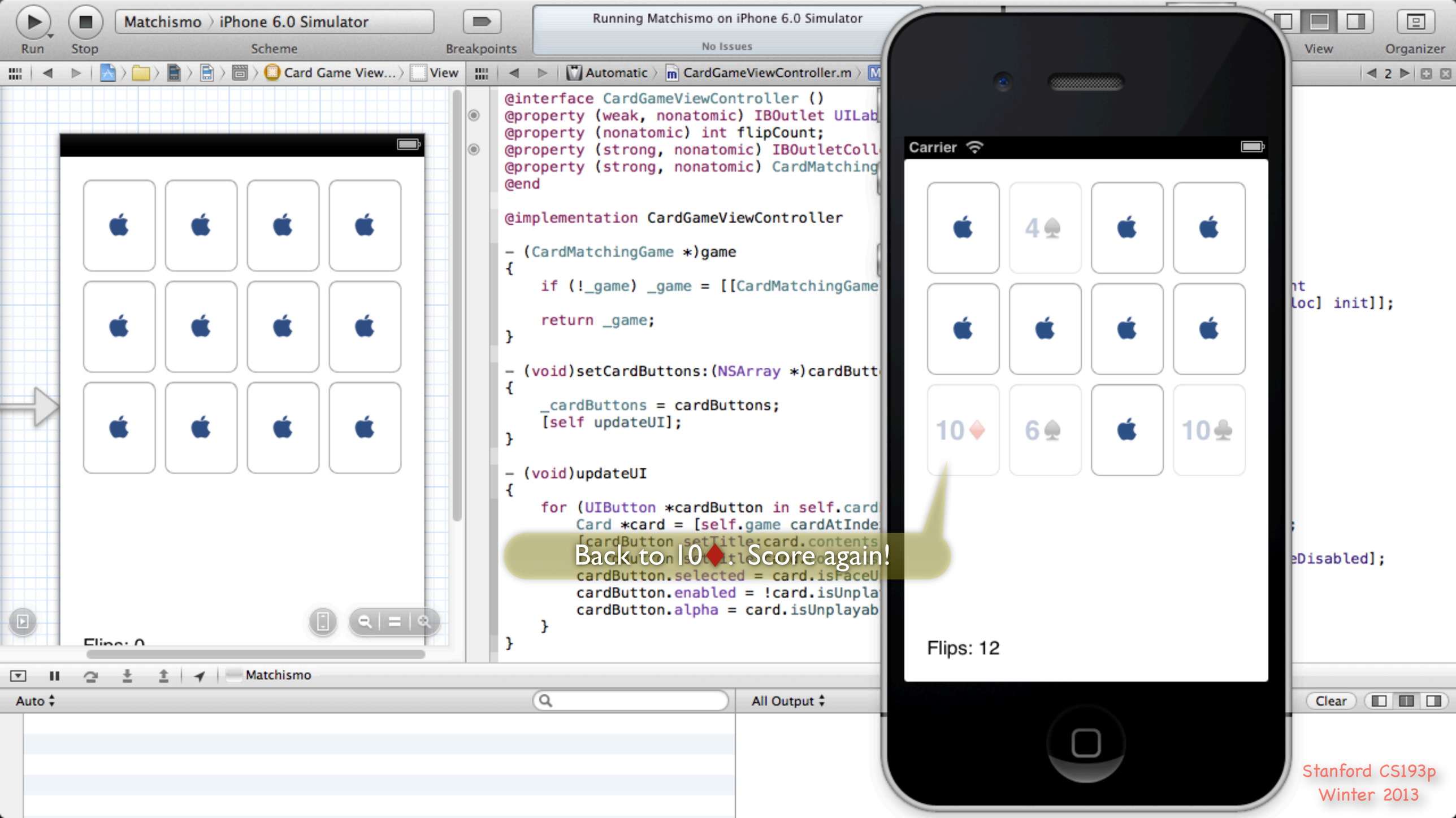

Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)set
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
nt
loc] init]];
eDisabled];
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
8 ♦ flipped back down.
Flips: 4
Stanford CS193p
Winter 2013

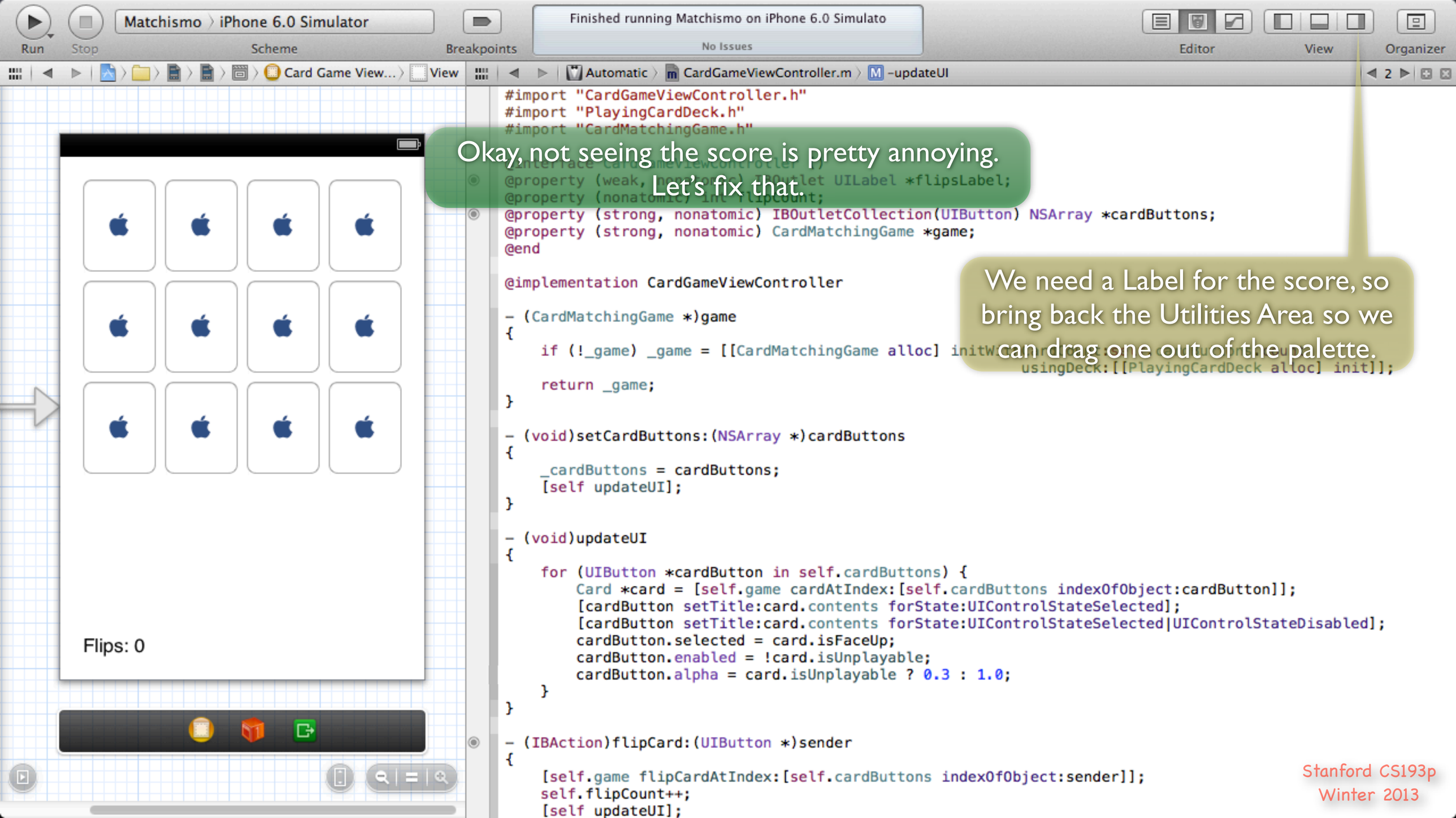

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
Found 6♠.
Aha!
We know there is another spade
(4♠) up in the top row.
Carrier
6♠
Flips: 5
Matchismo
Auto
All Output
Clear
Stanford CS193p
Winter 2013

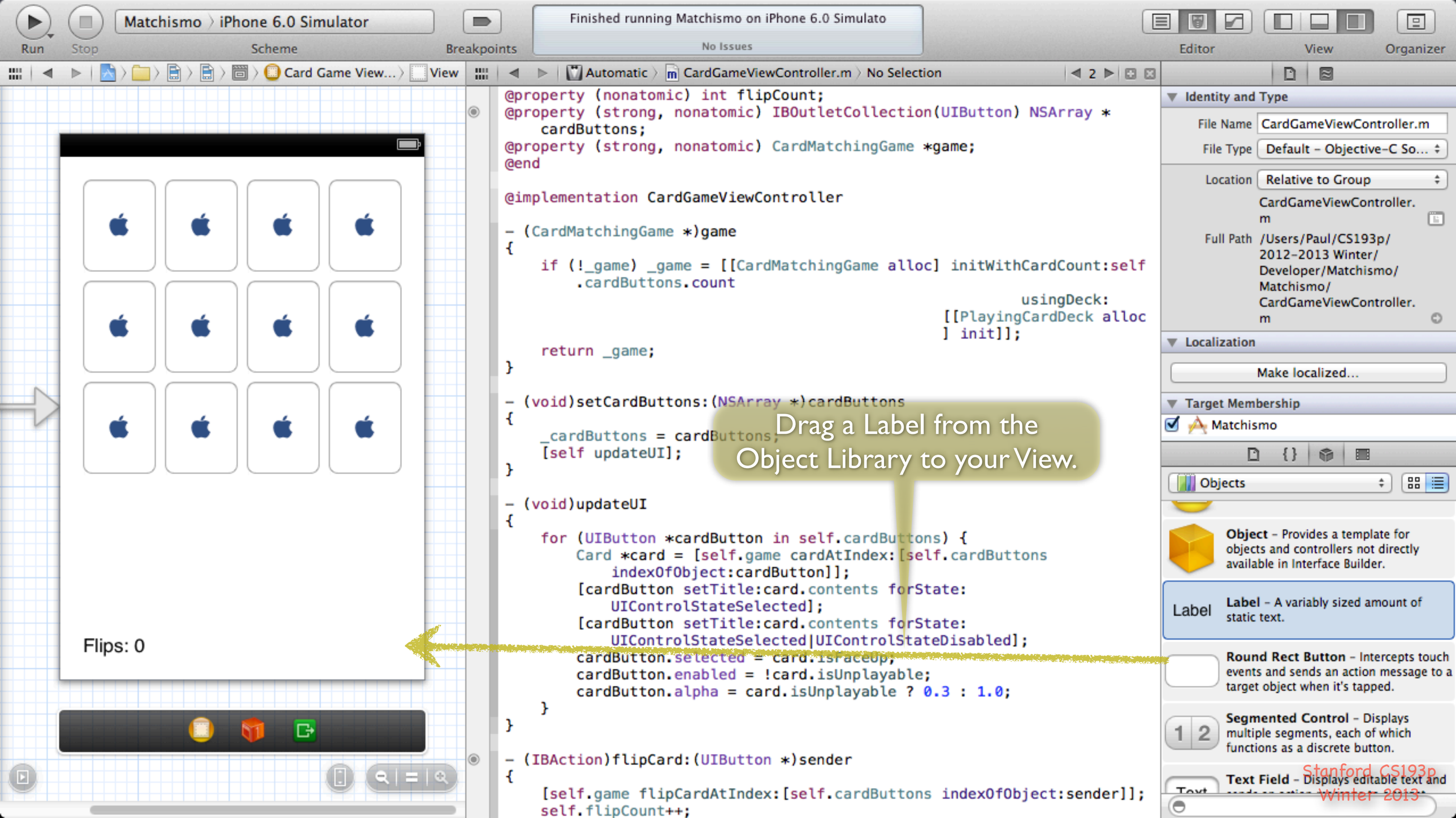

Matchismo › iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
Flip over the matching card.
Score!
Er, well, we can't see the score.
Flips: 0
Carrier
4♠
6♠
Flips: 6
Matchismo
Auto
All Output
Clear
Stanford CS193p
Winter 2013

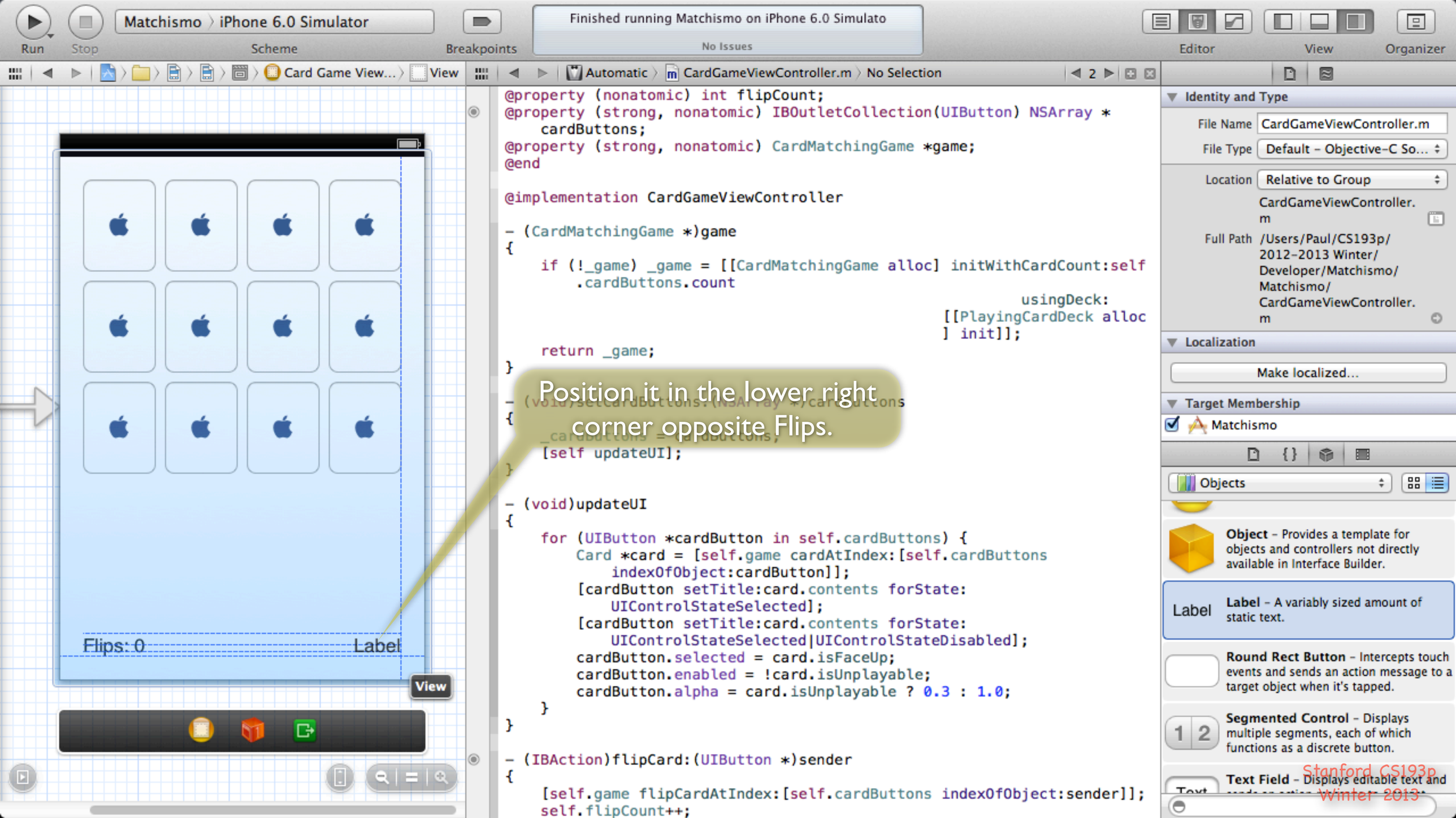

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
Found 10♦.
Carrier
4♠
10♦
6♠
Flips: 9
Matchismo
Auto
All Output
Clear
Stanford CS193p
Winter 2013

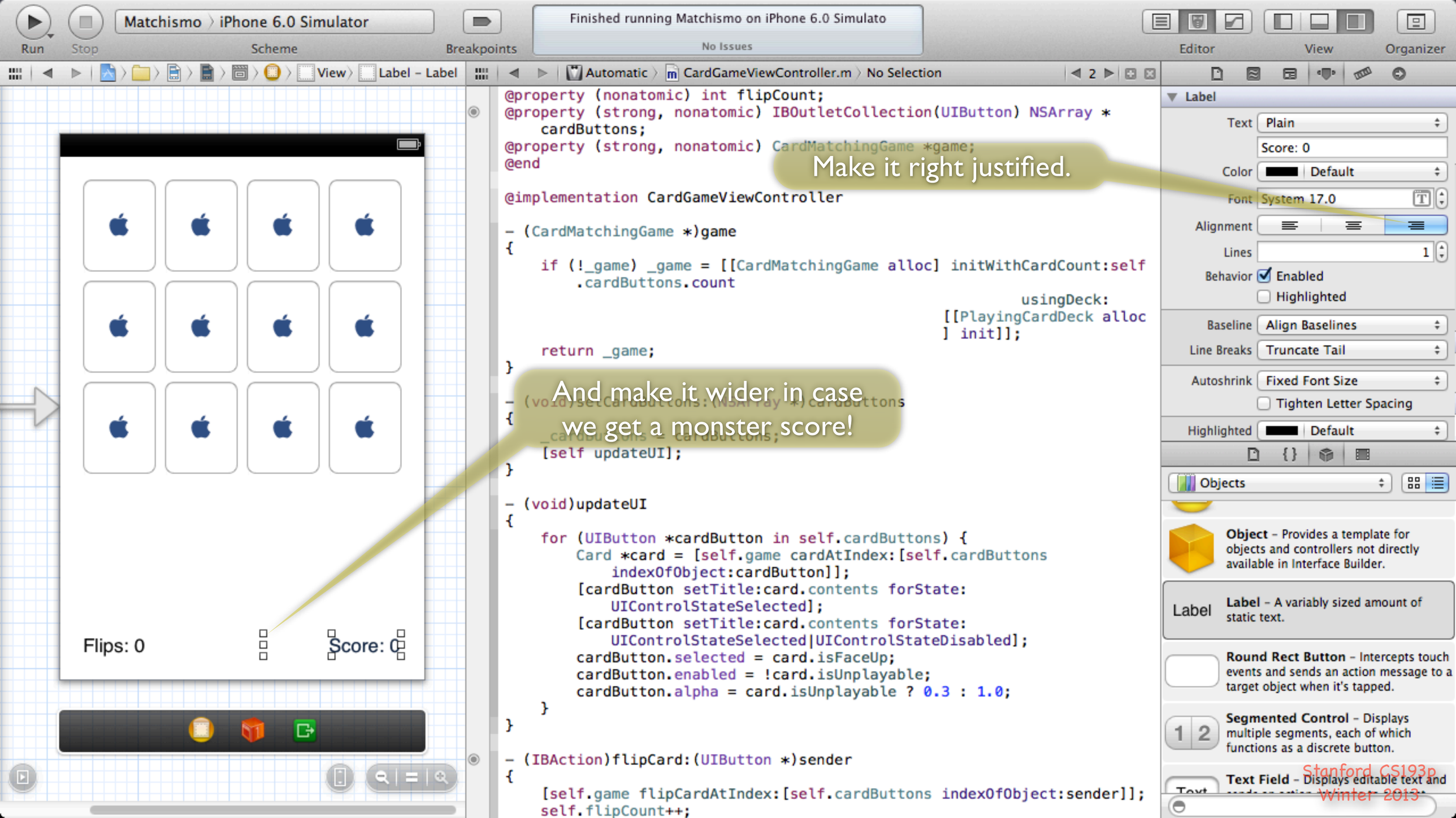

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
4♠
6♠
10♣
A couple of flips later, found 10♣.
Flips: 11
Stanford CS193p
Winter 2013

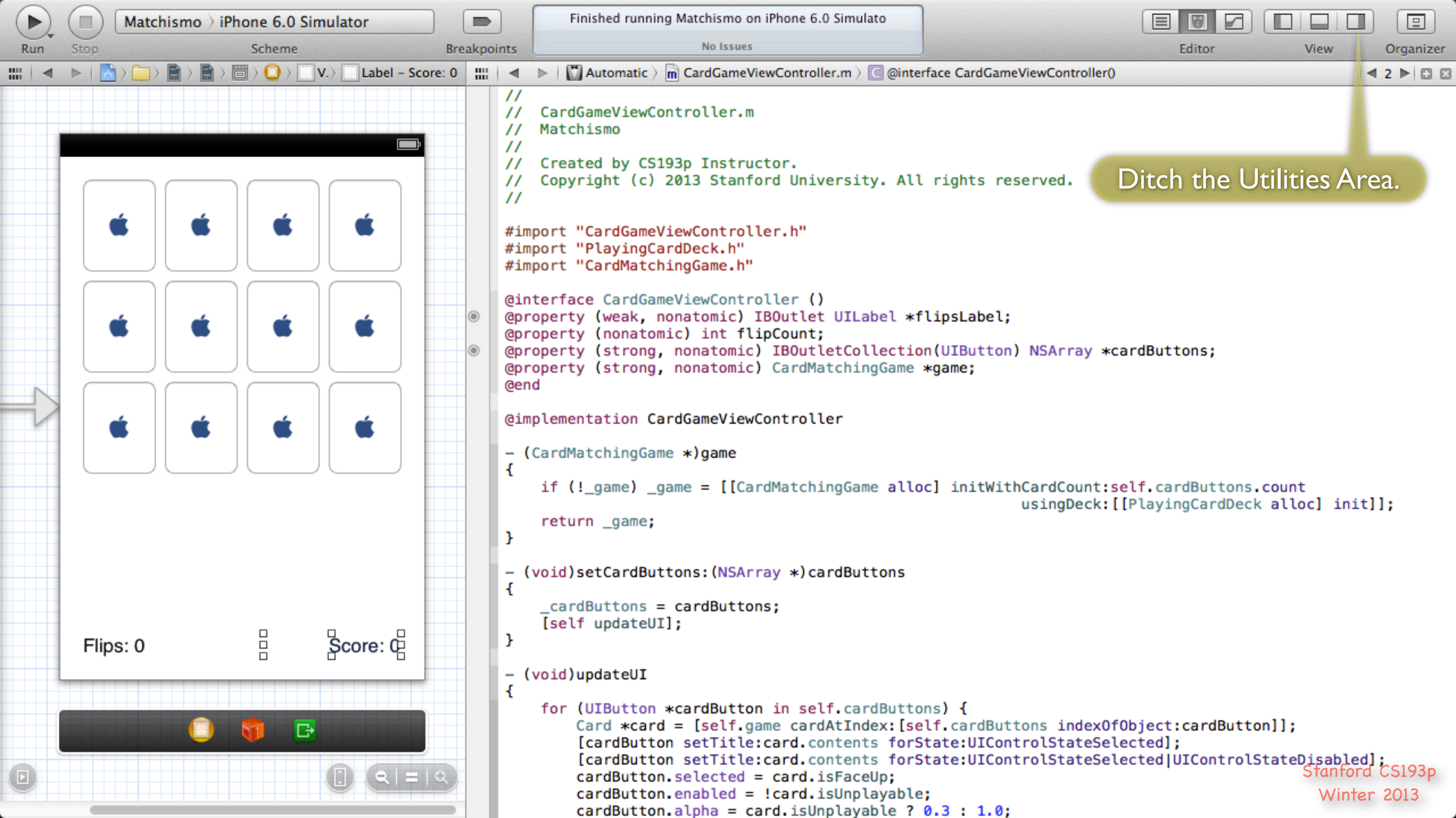

Matchismo › iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILab
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletColl
@property (strong, nonatomic) CardMatching
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
Flips: 0
Matchismo
Auto
All Output
Clear
Carrier
4♠
10♦
6♠
10♣
Back to 10♦. Score again!
Flips: 12
Stanford CS193p
Winter 2013

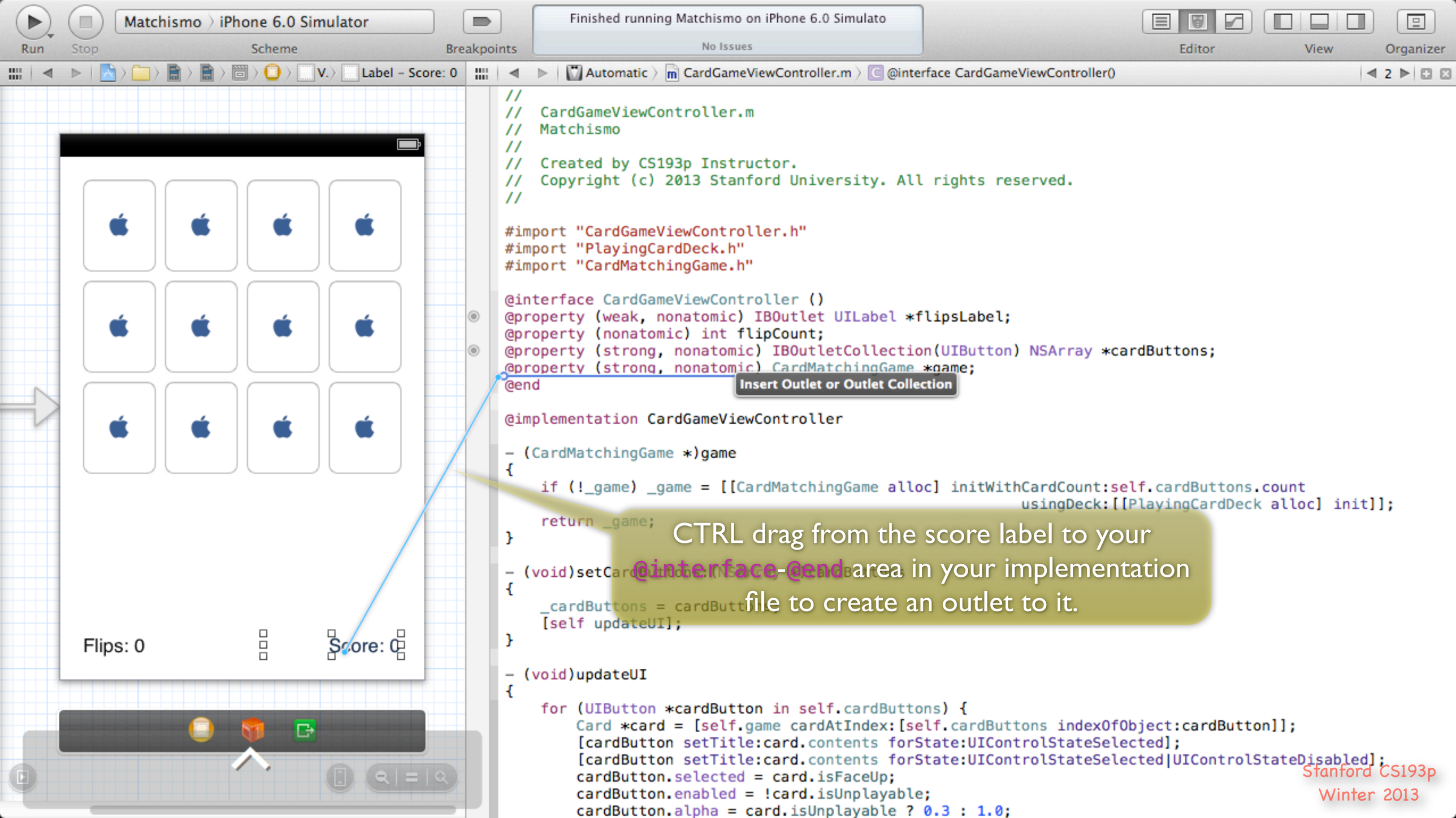

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
-updateUI
Flips: 0
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:[self.cardButtons count]
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
[self updateUI];
Okay, not seeing the score is pretty annoying.
Let's fix that.
We need a Label for the score, so bring back the Utilities Area so we can drag one out of the palette.
Stanford CS193p
Winter 2013

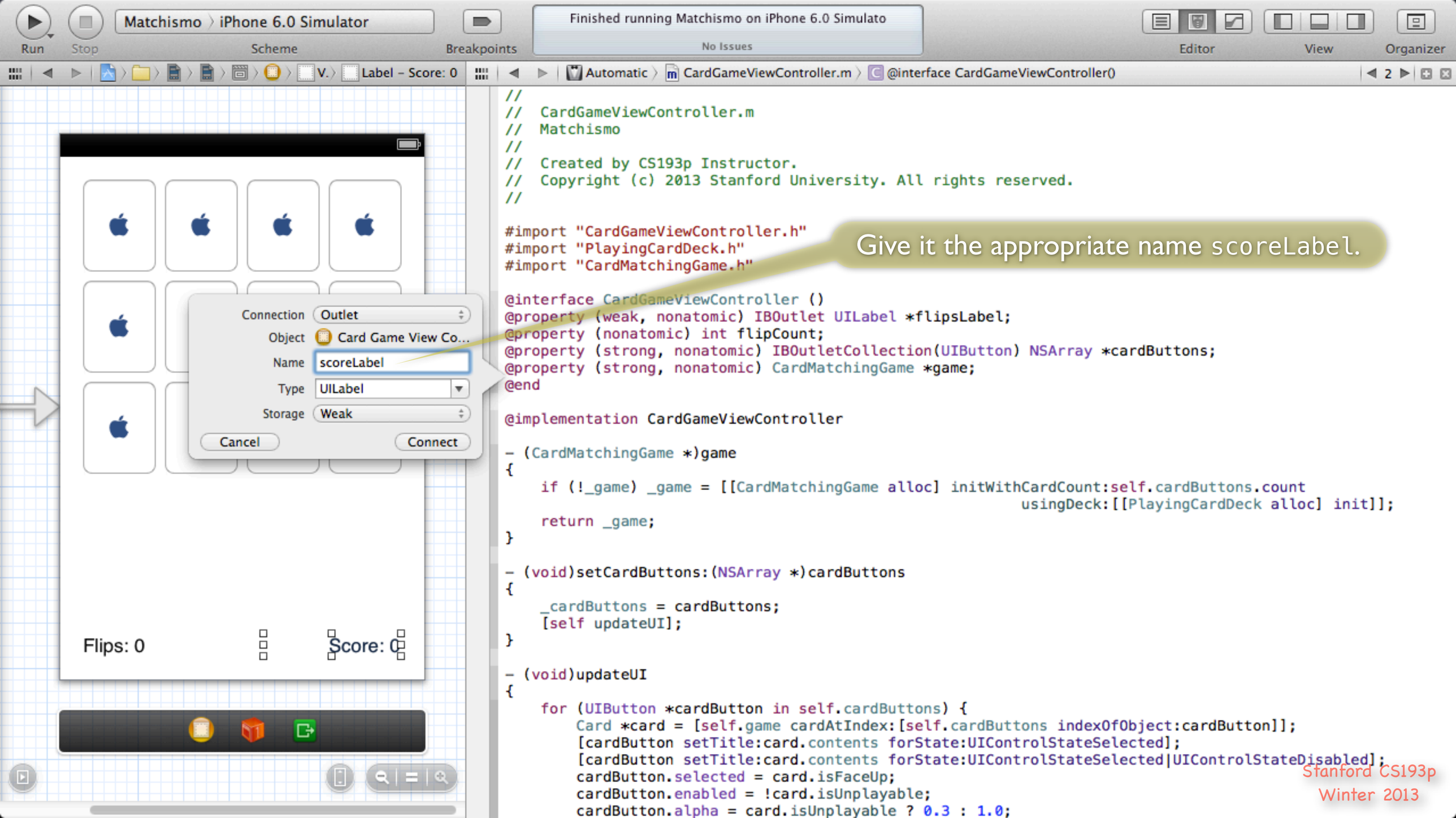

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 No Selection
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *
cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self
.cardButtons.count
usingDeck:
[[PlayingCardDeck alloc
] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons
indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:
UIControlStateSelected];
[cardButton setTitle:card.contents forState:
UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
Flips: 0
Drag a Label from the
Object Library to your View.
Identity and Type
File Name CardGameViewController.m
File Type Default – Objective-C So...
Location Relative to Group
CardGameViewController.
m
Full Path /Users/Paul/CS193p/
2012-2013 Winter/
Developer/Matchismo/
Matchismo/
CardGameViewController.
m
Localization
Make localized...
Target Membership
Matchismo
Objects
Object – Provides a template for objects and controllers not directly available in Interface Builder.
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
1 2
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
Text Field – Displays editable text and
Stanford CS193p
Winter 2013

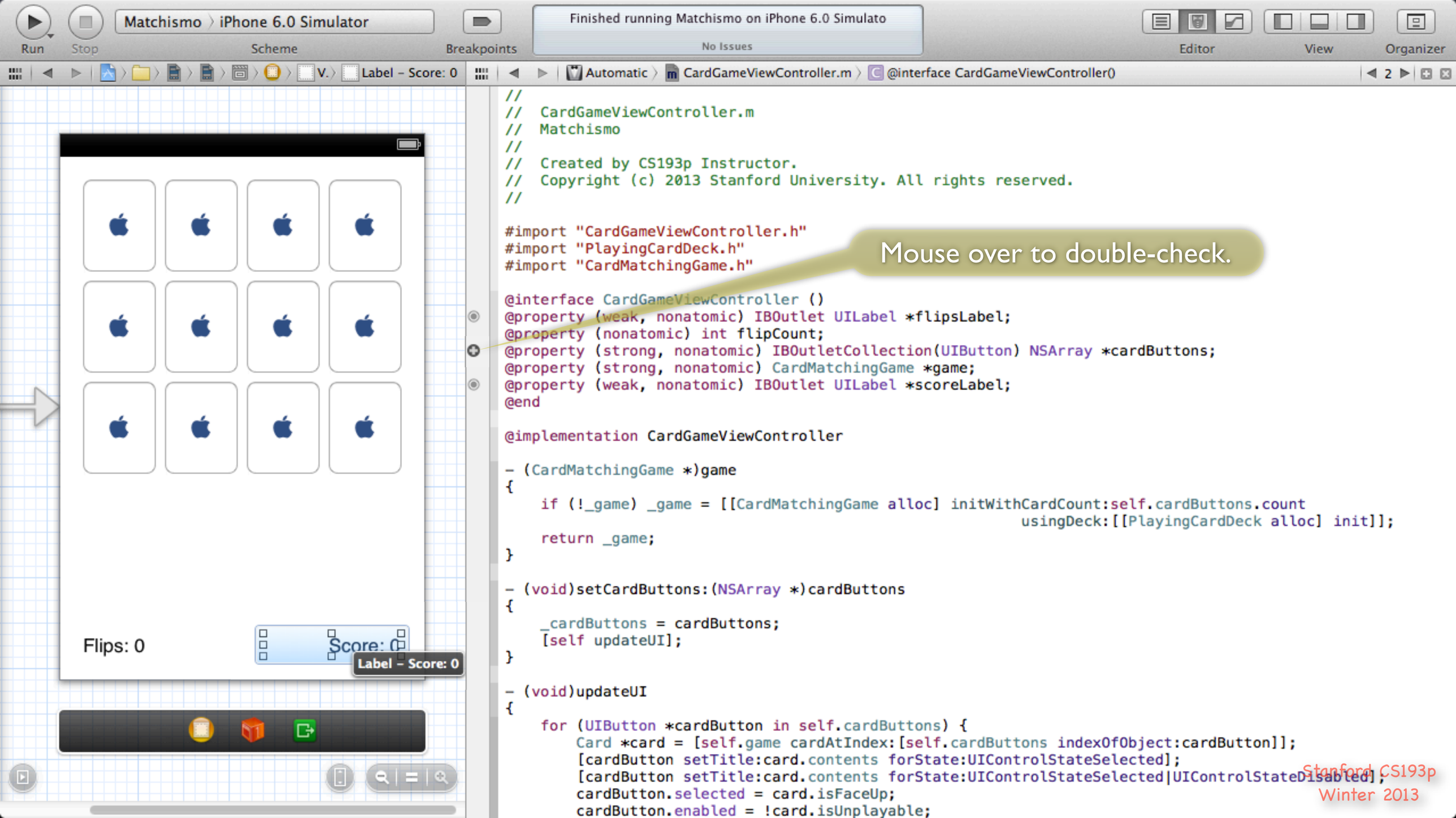

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m 〉 No Selection
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *
cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self
.cardButtons.count
usingDeck:
[[PlayingCardDeck alloc
] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons
indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:
UIControlStateSelected];
[cardButton setTitle:card.contents forState:
UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
Flips: 0
Label
View
Position it in the lower right corner opposite Flips.
Identity and Type
File Name
CardGameViewController.m
File Type
Default – Objective-C So...
Location
Relative to Group
CardGameViewController.m
Full Path
/Users/Paul/CS193p/2012-2013 Winter/Developer/Matchismo/Matchismo/CardGameViewController.m
Localization
Make localized...
Target Membership
Matchismo
Objects
Object – Provides a template for objects and controllers not directly available in Interface Builder.
Label
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
1 2
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
Text Field – Displays editable text and
Stanford CS193p
Winter 2013

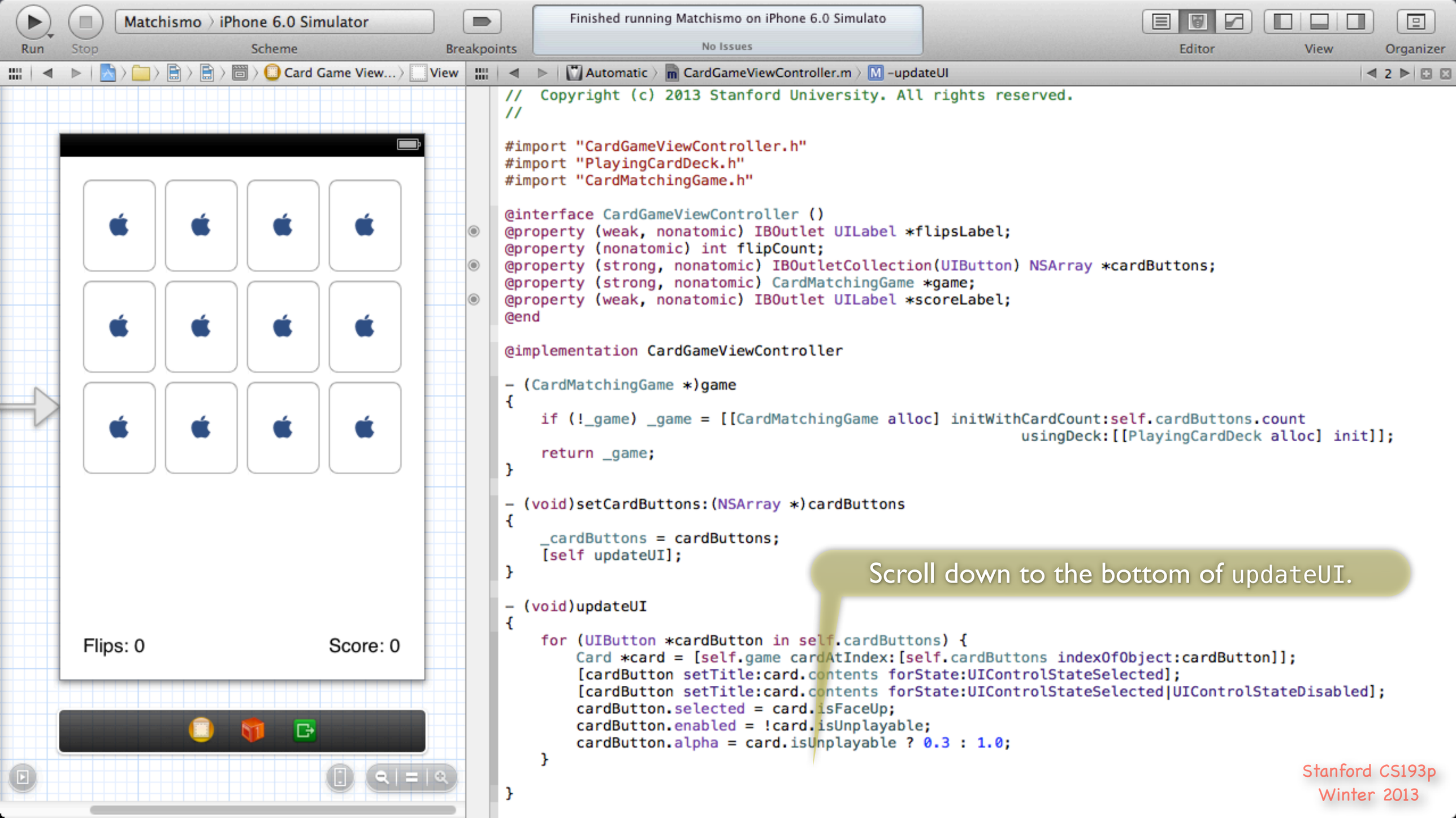

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
View 〉 Label – Label
Automatic 〉 CardGameViewController.m 〉 No Selection
Flips: 0
Score: 0
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *
cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self
.cardButtons.count
usingDeck:
[[PlayingCardDeck alloc
] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons
indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:
UIControlStateSelected];
[cardButton setTitle:card.contents forState:
UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardButtons indexOfObject:sender]];
self.flipCount++;
Make it right justified.
And make it wider in case
we get a monster score!
Label
Text Plain
Score: 0
Color Default
Font System 17.0
Alignment
Lines 1
Behavior Enabled
Highlighted
Baseline Align Baselines
Line Breaks Truncate Tail
Autoshrink Fixed Font Size
Tighten Letter Spacing
Highlighted Default
Objects
Object – Provides a template for objects and controllers not directly available in Interface Builder.
Label – A variably sized amount of static text.
Round Rect Button – Intercepts touch events and sends an action message to a target object when it's tapped.
Segmented Control – Displays multiple segments, each of which functions as a discrete button.
Text Field – Displays editable text and
Stanford CS193p
Winter 2013

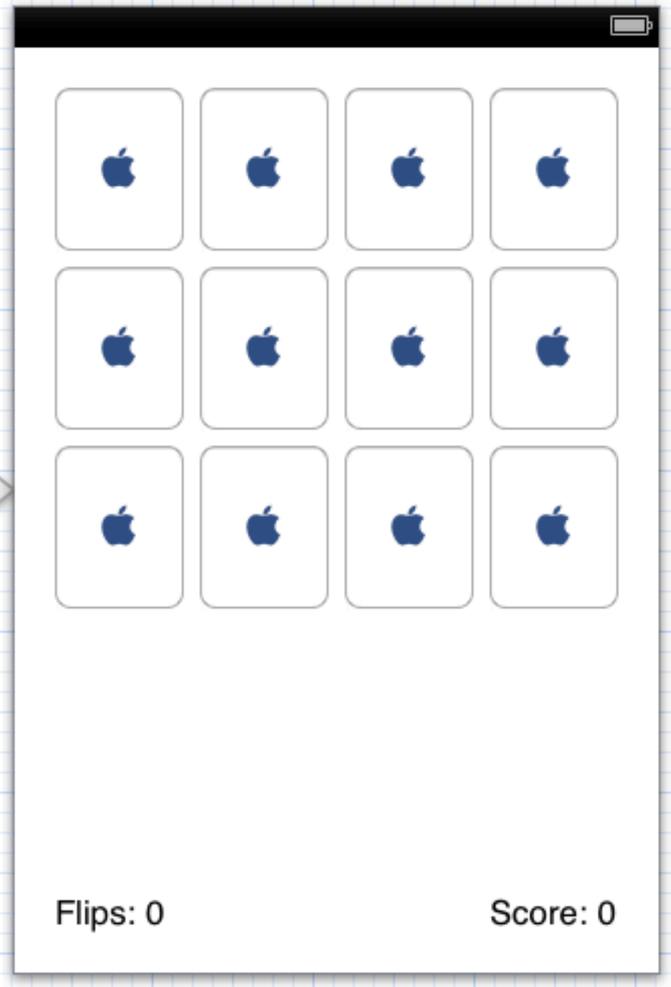

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Label – Score: 0
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
Flips: 0
Score: 0
Ditch the Utilities Area.
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Label – Score: 0
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
Flips: 0
Score: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
Insert Outlet or Outlet Collection
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
CTRL drag from the score label to your @interface-@end area in your implementation file to create an outlet to it.
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Label – Score: 0
Automatic 〉 CardGameViewController.m 〉 @interface CardGameViewController()
Connection Outlet
Object Card Game View Co...
Name scoreLabel
Type UILabel
Storage Weak
Cancel
Connect
Flips: 0
Score: 0
Give it the appropriate name scoreLabel.
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
Stanford CS193p
Winter 2013

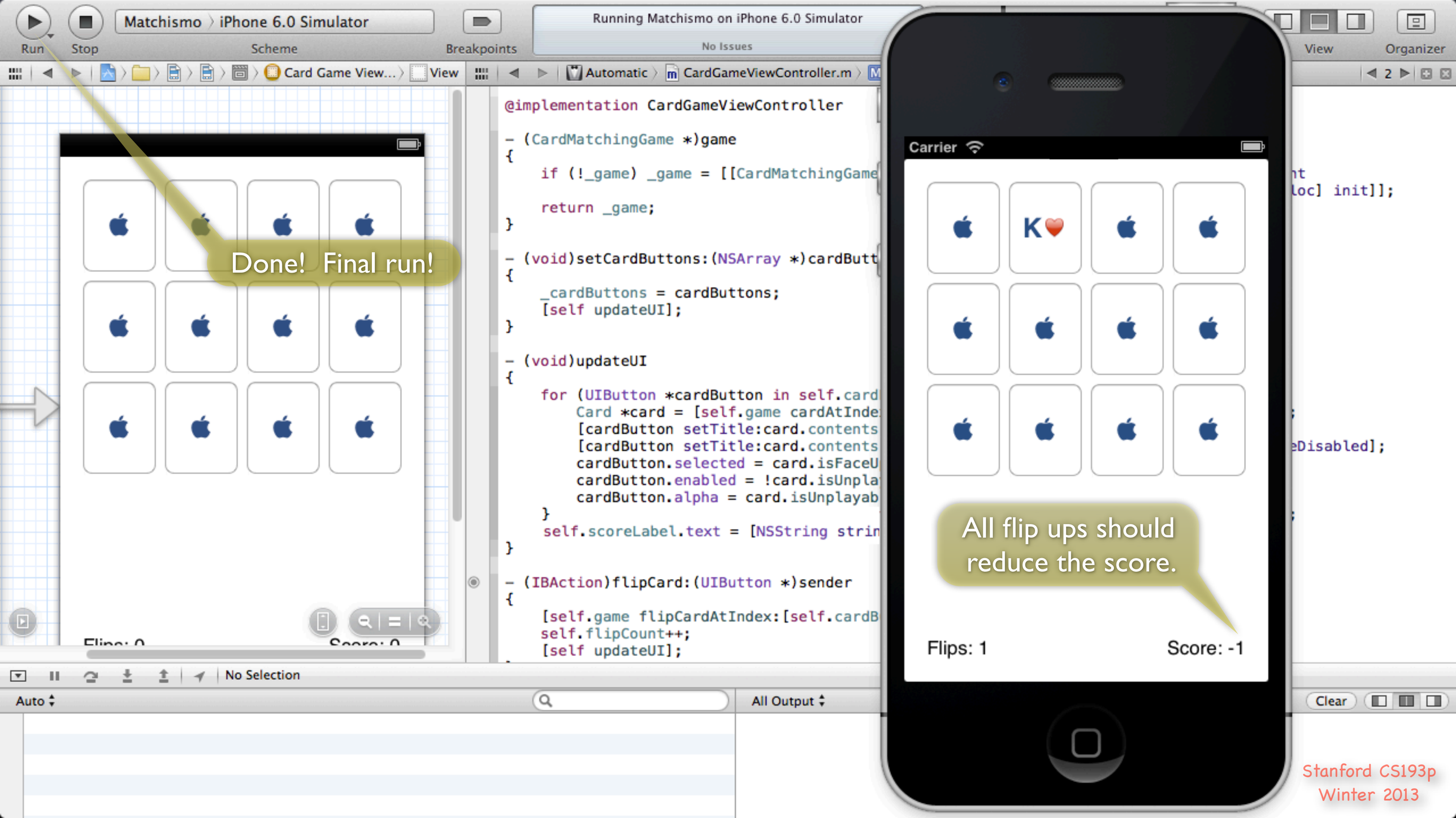

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Label – Score: 0
Automatic
CardGameViewController.m
@interface CardGameViewController()
Flips: 0
Score: 0
Label – Score: 0
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
Mouse over to double-check.
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@property (weak, nonatomic) IBOutlet UILabel *scoreLabel;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
Stanford CS193p
Winter 2013

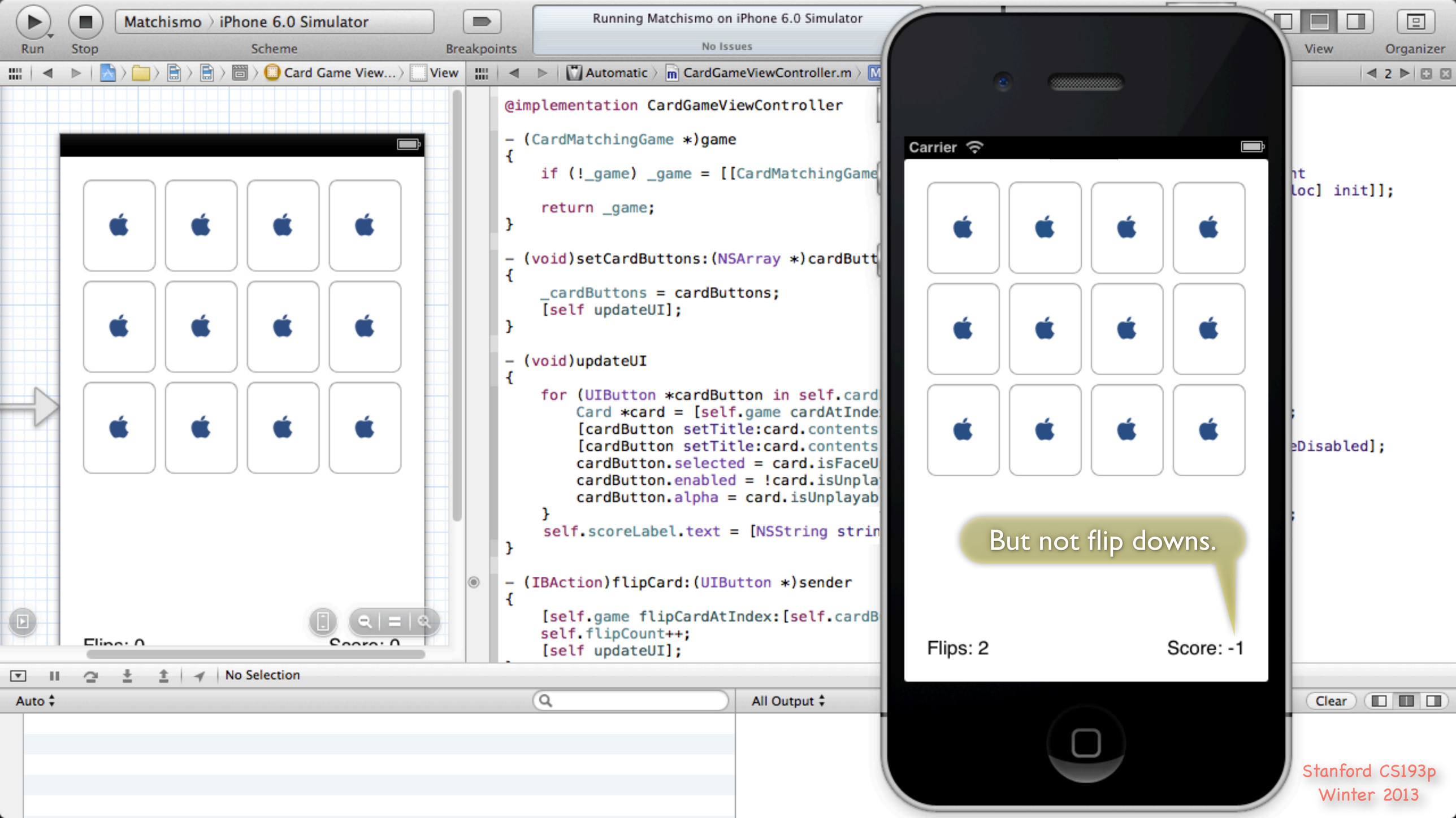

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
-updateUI
Flips: 0
Score: 0
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@property (weak, nonatomic) IBOutlet UILabel *scoreLabel;
@end
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
usingDeck:[[PlayingCardDeck alloc] init]];
return _game;
}
- (void)setCardButtons:(NSArray *)cardButtons
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.cardButtons) {
Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
[cardButton setTitle:card.contents forState:UIControlStateSelected];
[cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
cardButton.selected = card.isFaceUp;
cardButton.enabled = !card.isUnplayable;
cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
}
}
Scroll down to the bottom of updateUI.
Stanford CS193p
Winter 2013

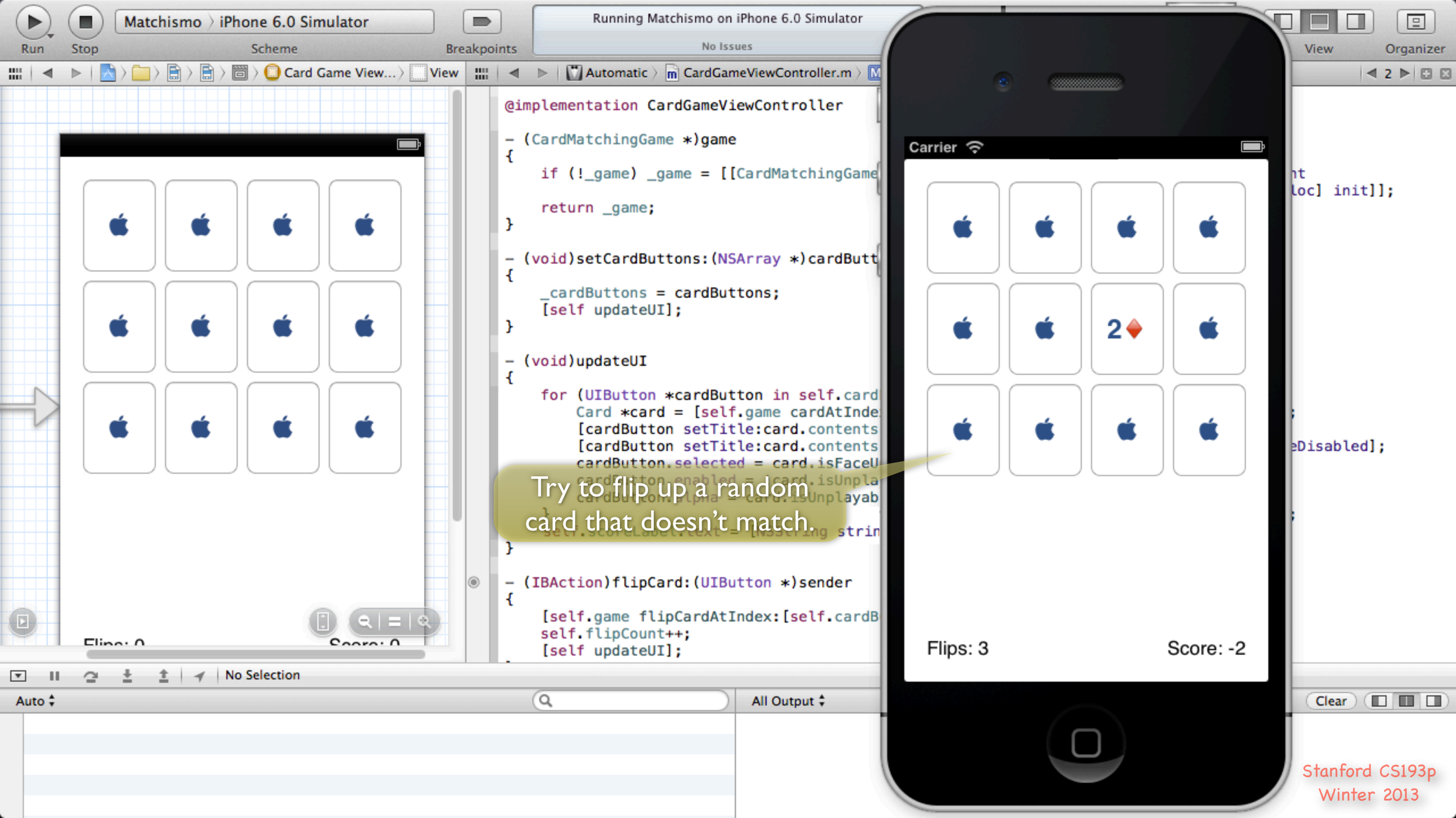


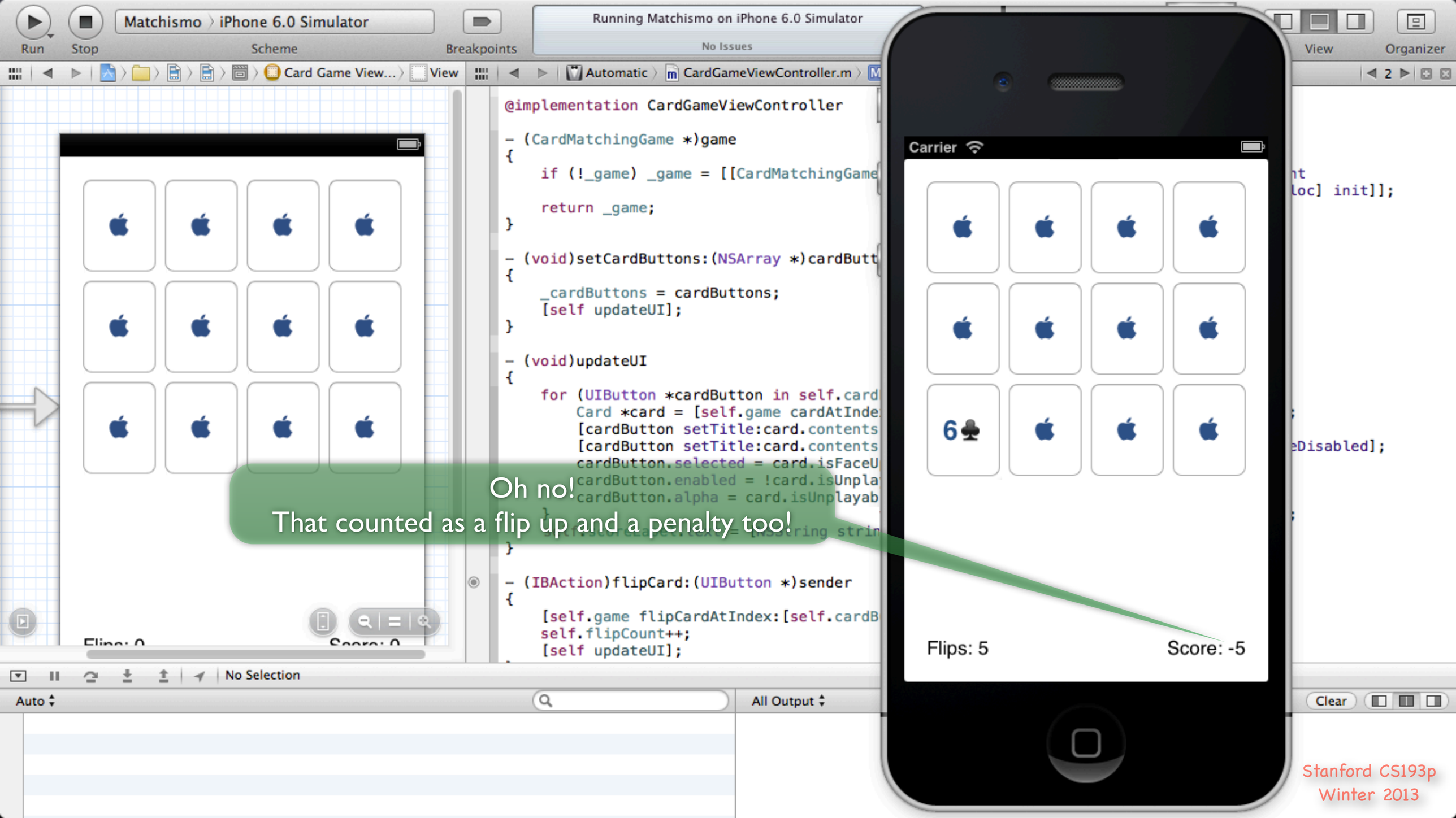

```
//  Copyright (c) 2013 Stanford University. All rights reserved.
//

#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"

@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@property (weak, nonatomic) IBOutlet UILabel *scoreLabel;
@end

@implementation CardGameViewController

- (CardMatchingGame *)game
{
    if (!_game) _game = [[CardMatchingGame alloc] initWithCardCount:self.cardButtons.count
                                                          usingDeck:[[PlayingCardDeck alloc] init]];
    return _game;
}

- (void)setCardButtons:(NSArray *)cardButtons
{
    _cardButtons = cardButtons;
    [self updateUI];
}

- (void)updateUI
{
    for (UIButton *cardButton in self.cardButtons) {
        Card *card = [self.game cardAtIndex:[self.cardButtons indexOfObject:cardButton]];
        [cardButton setTitle:card.contents forState:UIControlStateSelected];
        [cardButton setTitle:card.contents forState:UIControlStateSelected|UIControlStateDisabled];
        cardButton.selected = card.isFaceUp;
        cardButton.enabled = !card.isUnplayable;
        cardButton.alpha = card.isUnplayable ? 0.3 : 1.0;
    }
    self.scoreLabel.text = [NSString stringWithFormat:@"Score: %d", self.game.score];
}
```

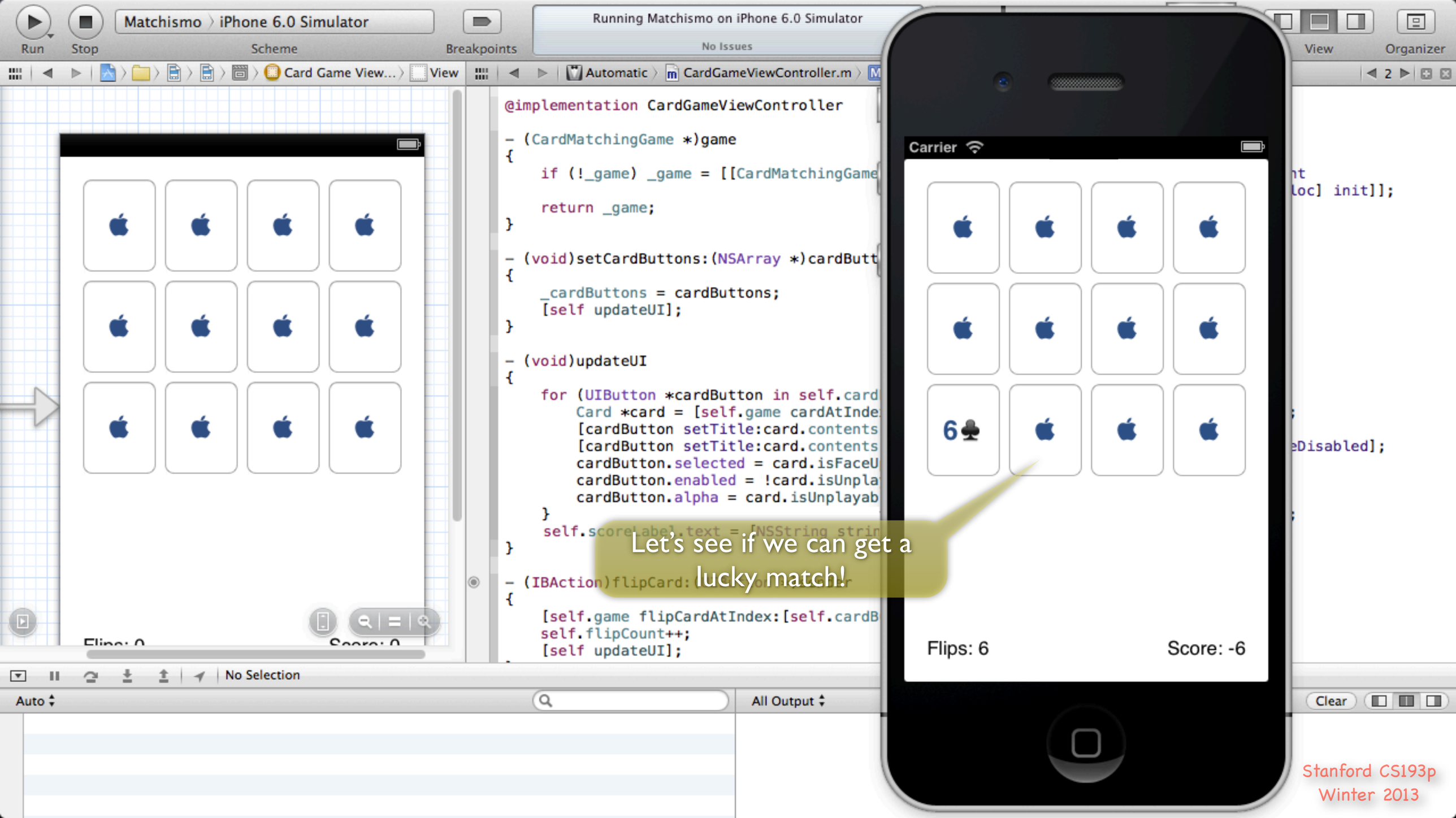
And update the score (just like we did for flips).

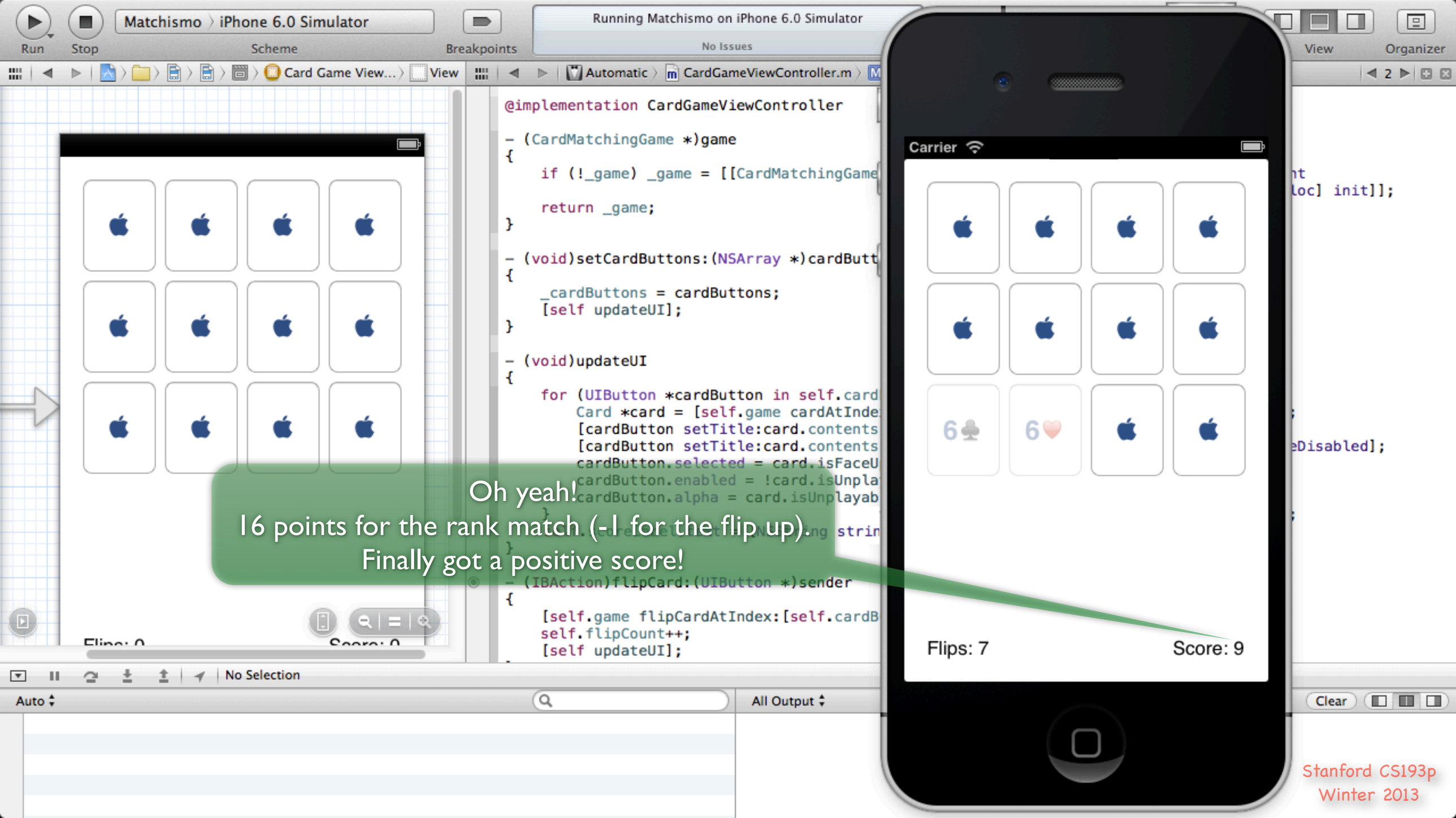

Run
Stop
Matchismo › iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
self.scoreLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
Done! Final run!
Flips: 0
Score: 0
No Selection
Auto
All Output
Clear
Carrier
All flip ups should reduce the score.
Flips: 1
Score: -1
Stanford CS193p
Winter 2013

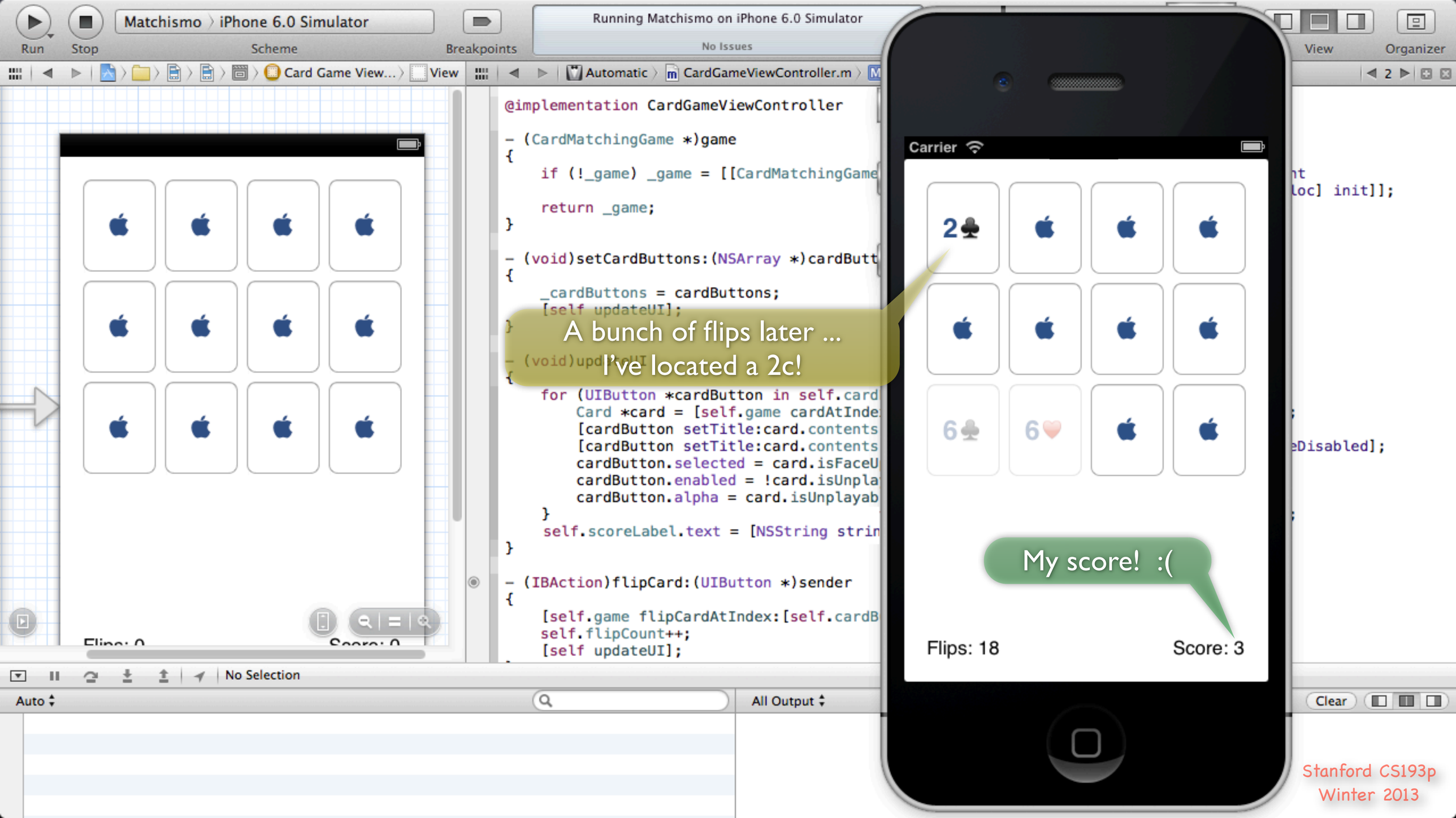

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
self.scoreLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
Carrier
But not flip downs.
Flips: 2
Score: -1
No Selection
Auto
All Output
Clear
Stanford CS193p
Winter 2013

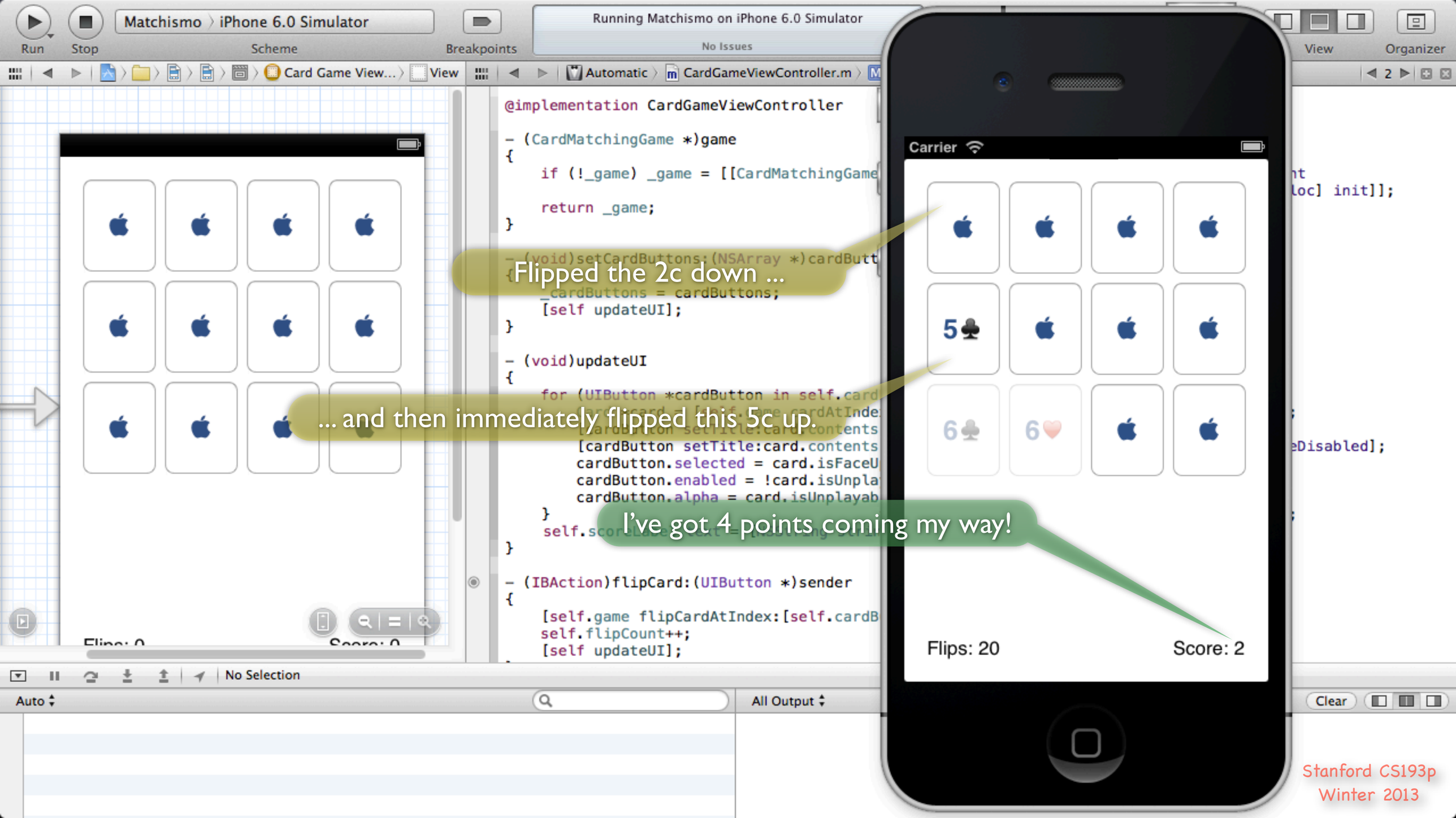

Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
}
Try to flip up a random card that doesn't match.
Carrier
2
Flips: 3
Score: -2
Flips: 0
Score: 0
No Selection
Auto
All Output
Clear
Stanford CS193p
Winter 2013

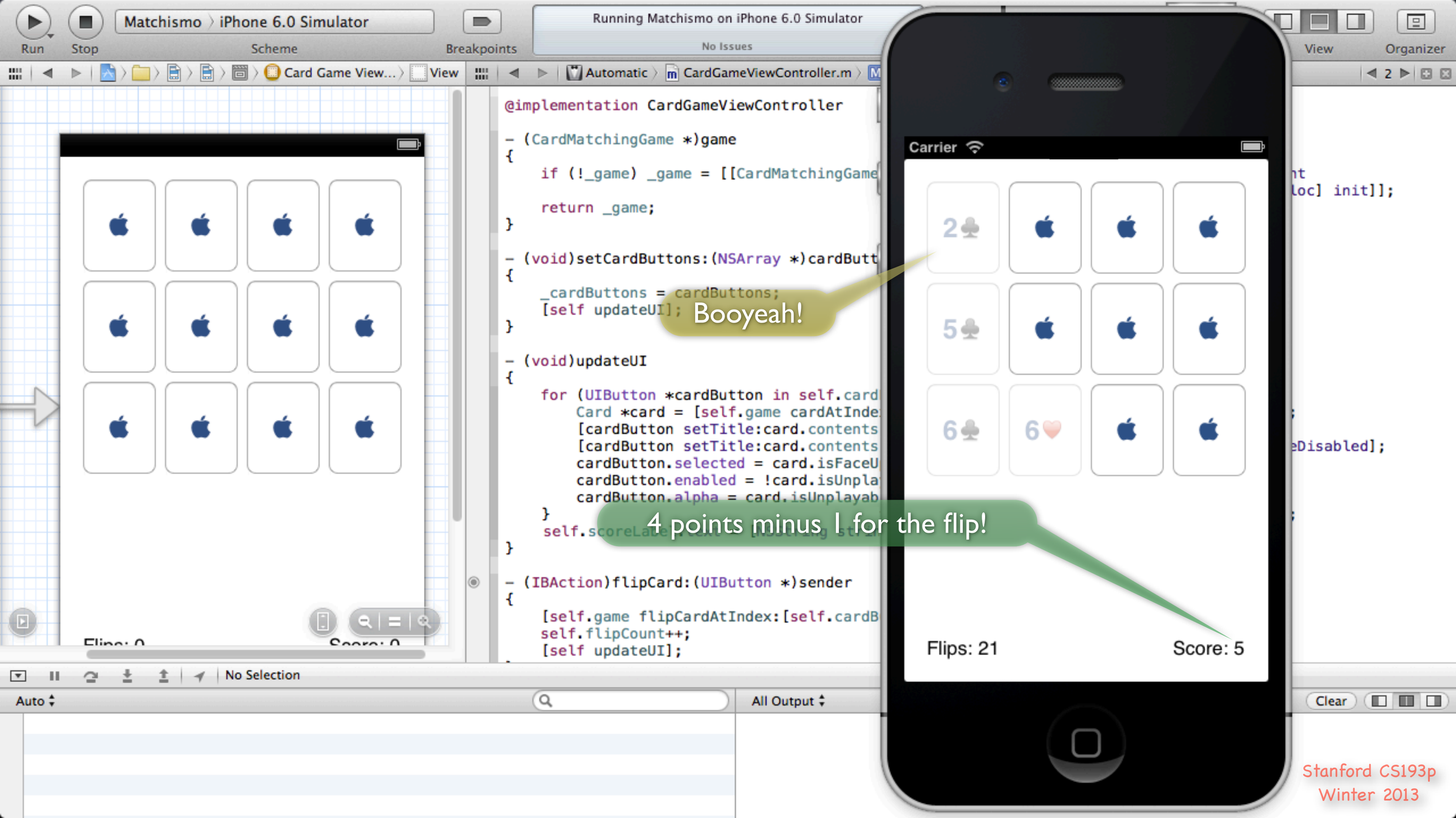

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
Flips: 0
Score: 0
Oh no!
That counted as a flip up and a penalty too!
Carrier
6
Flips: 5
Score: -5
No Selection
Auto
All Output
Clear
Stanford CS193p
Winter 2013

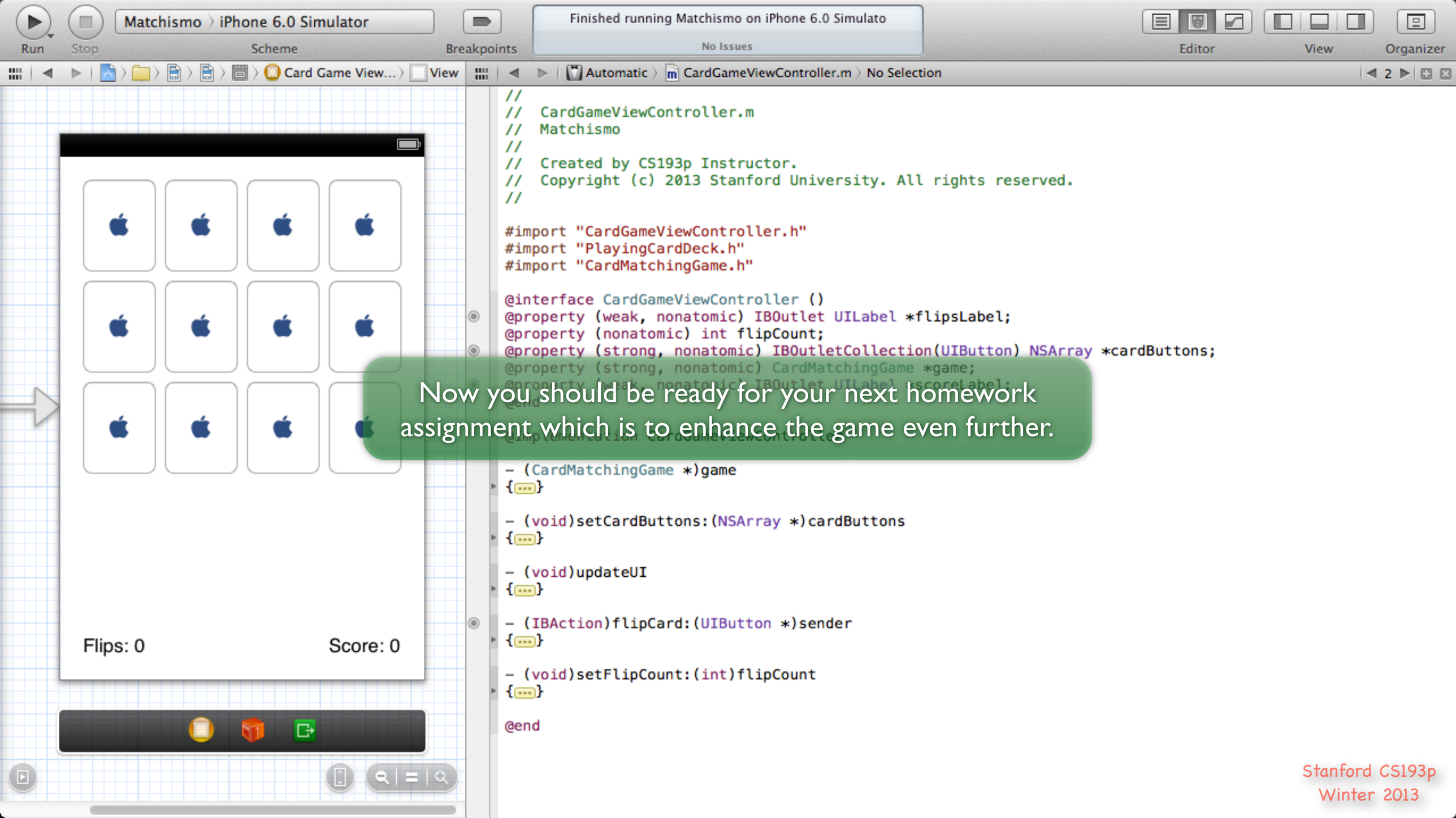

Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
self.scoreLabel.text = [NSString strin
}
- (IBAction)flipCard:
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
Flips: 0
Score: 0
No Selection
Auto
All Output
Clear
Carrier
6
Flips: 6
Score: -6
Let's see if we can get a lucky match!
Stanford CS193p
Winter 2013


Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
Oh yeah!
16 points for the rank match (-1 for the flip up).
Finally got a positive score!
Carrier
Flips: 7
Score: 9
No Selection
Auto
All Output
Clear
Stanford CS193p
Winter 2013


Matchismo
iPhone 6.0 Simulator
Run
Stop
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
self.scoreLabel.text = [NSString strin
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
A bunch of flips later ...
I've located a 2c!
Carrier
2♣
6♣
6♥
My score! :(
Flips: 18
Score: 3
No Selection
Auto
All Output
Clear
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
Flipped the 2c down ...
... and then immediately flipped this 5c up.
I've got 4 points coming my way!
Carrier
5♣
6♣
6♥
Flips: 20
Score: 2
No Selection
Auto
All Output
Clear
Stanford CS193p
Winter 2013


Run
Stop
Matchismo 〉 iPhone 6.0 Simulator
Scheme
Breakpoints
Running Matchismo on iPhone 6.0 Simulator
No Issues
View
Organizer
Card Game View... 〉 View
Automatic 〉 CardGameViewController.m
@implementation CardGameViewController
- (CardMatchingGame *)game
{
if (!_game) _game = [[CardMatchingGame
return _game;
}
- (void)setCardButtons:(NSArray *)cardButt
{
_cardButtons = cardButtons;
[self updateUI];
}
- (void)updateUI
{
for (UIButton *cardButton in self.card
Card *card = [self.game cardAtInde
[cardButton setTitle:card.contents
[cardButton setTitle:card.contents
cardButton.selected = card.isFaceU
cardButton.enabled = !card.isUnpla
cardButton.alpha = card.isUnplayab
}
self.scoreLa
}
- (IBAction)flipCard:(UIButton *)sender
{
[self.game flipCardAtIndex:[self.cardB
self.flipCount++;
[self updateUI];
Booyeah!
4 points minus 1 for the flip!
Carrier
2♣
5♣
6♣
6♥
Flips: 21
Score: 5
Flips: 0
Score: 0
No Selection
Auto
All Output
Clear
Stanford CS193p
Winter 2013


Run
Stop
Matchismo
iPhone 6.0 Simulator
Scheme
Breakpoints
Finished running Matchismo on iPhone 6.0 Simulato
No Issues
Editor
View
Organizer
Card Game View...
View
Automatic
CardGameViewController.m
No Selection
2
//
// CardGameViewController.m
// Matchismo
//
// Created by CS193p Instructor.
// Copyright (c) 2013 Stanford University. All rights reserved.
//
#import "CardGameViewController.h"
#import "PlayingCardDeck.h"
#import "CardMatchingGame.h"
@interface CardGameViewController ()
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@property (nonatomic) int flipCount;
@property (strong, nonatomic) IBOutletCollection(UIButton) NSArray *cardButtons;
@property (strong, nonatomic) CardMatchingGame *game;
@property (weak, nonatomic) IBOutlet UILabel *scoreLabel;
@end
- (CardMatchingGame *)game
{...}
- (void)setCardButtons:(NSArray *)cardButtons
{...}
- (void)updateUI
{...}
- (IBAction)flipCard:(UIButton *)sender
{...}
- (void)setFlipCount:(int)flipCount
{...}
@end
Flips: 0
Score: 0
Now you should be ready for your next homework assignment which is to enhance the game even further.
Stanford CS193p
Winter 2013

Review

- Things you should know by now ...
  MVC
  Xcode
  Basic Objective-C

MVC

- Model is UI-independent
  Cards and Decks, not UIButtons and UILabels
- View is (so far) completely generic UI elements
  UIButton
  UILabel
- Controller interprets Model for View (and vice-versa)
  Example: converting isFaceUp to selected state of a button
  Example: converting isUnplayable to enabled state of a button
  Example: taking a button touch and turning it into a flipCard in the Model
  Target/Action and Outlets (so far)

Xcode

- Create a Project and maneuver through Xcode's UI
  Hide/Show Navigator, Utilities, Assistant Editor, etc., etc. Also how to run in the Simulator.
- Edit
  Not just code, but also your storyboard, use Attributes Inspector to edit buttons, labels, et. al.
  Ctrl-drag to make connections (actions and outlets).
  Right click on buttons, etc., to find out about and disconnect connections.
  Look at warnings and errors (and get rid of them hopefully!). Debugger on Friday this week.
- Add classes to your project
  e.g. you added the `Card`, etc., Model classes in your Homework assignment.
- Use the documentation
  Many ways to get to documentation, but ALT-clicking on a keyword is one of the coolest.
  Once there, search and click on links to find what you want.
  Crucial to being a good iOS programming to become familiar with all the documentation.

Basic Objective-C

- Classes

  Header .h (public) versus Implementation .m (private)

  `@interface MyClass : MySuperclass` … `@end` (only in header file)

  `@interface MyClass()` … `@end` (only in implementation file)

  `@implementation` … `@end` (only in implementation file)

  `#import`

- Properties

  `@property (nonatomic)` <type> <property name> (always `nonatomic` in this course)

  It's just setter and getter methods. Default ones automatically generated for you by compiler.

  Better than instance variables alone (lazy instantiation, consistency checking, UI updating, etc.).

  `@property (strong` or `weak)` <type which is a pointer to an object> <property name>

  `@property (getter=`<getter name>`)` …

  `@property (readonly)` … & `@property (readwrite)` …

  Invoking setter and getter using dot notation, e.g., `self.cards = …` or `if (rank > self.rank)` …

  `@synthesize` <prop name> `= _`<prop name> (only if you implement both setter and getter)

Basic Objective-C

- Types and Memory

  Types: `MyClass *`, `BOOL` (`YES` or `NO`), `int`, `NSUInteger`, etc. (`id` not fully explained yet.)

  All objects live in the heap (i.e. we only have pointers to them).

  Object storage in the heap is managed automatically (guided by `strong` and `weak` declarations).

  Lazy instantiation (using a `@property`'s getter to `alloc`ate and `init`ialize the object that the `@property` points to in an "on demand" fashion). Not everything is lazily instantiated, btw. :)

  If a pointer has the value `nil` (i.e. `0`), it means the pointer does not point to anything.

- Methods

  Declaring and defining instance methods, e.g., `- (int)match:(NSArray *)otherCards`

  Declaring and defining class methods, e.g., `+ (NSArray *)validSuits`

  Invoking instance methods, e.g., `[myArray addObject:anObject]`

  Invoking class methods, e.g., `unsigned int rank = [PlayingCard maxRank]`

  Method's name and its parameters are interspersed, e.g., `[deck addCard:aCard atTop:YES]`

Basic Objective-C

- NSString

  Immutable and usually created by manipulating other strings or @"" notation or class methods.

  e.g. `NSString *myString = @"hello"`

  e.g. `NSString *myString = [otherString stringByAppendingString:yetAnotherString]`

  e.g. `NSString *myString = [NSString stringWithFormat:@"%d%@", myInt, myObj]`

  There is an `NSMutableString` subclass but we almost never use it.

  Instead, we create new strings by asking existing ones to create a modified version of themselves.

Basic Objective-C

- NSArray

  Immutable and usually created by manipulating other arrays (not seen yet) or with @[] notation.

  @[@“a”,@“b”] is the same as [[NSArray alloc] initWithObjects:@“a”,@“b”,nil].

  Access the array using [] notation (like a normal C array), e.g., myArray[index].

  myArray[index] works the same as [myArray objectAtIndex:index].

  The method count (which returns NSUInteger) will tell you how many items in the array.

  (We accidentally used dot notation to call this method in Lecture 2!)

  Be careful not to access array index out of bounds (crashes). Only lastObject method immune.

  Can contain any mix of objects of any class) No syntax to say which it contains.

  In other words, the “type” of object in an NSArray is id (pointer to object of any/unknown class).

  Use NSMutableArray subclass if mutability is needed. Then you get …

  ```
  - (void)addObject:(id)anObject;
  - (void)insertObject:(id)anObject atIndex:(int)index;
  - (void)removeObjectAtIndex:(int)index;
  ```

  Usually created with [[NSMutableArray alloc] init]

Basic Objective-C

- Creating Objects in the Heap

Allocation (`NSObject`'s `alloc`) and initialization (with an `init`... method) always happen together!

e.g. `[[NSMutableArray alloc] init]`

e.g. `[[CardMatchingGame alloc] initWithCardCount:c usingDeck:d]`

Writing initializers for your own classes ...

Two kinds of initializers: designated (one per class) and convenience (zero or more per class).

Only denoted by comments (not enforced by the syntax of the language in any way).

Must call your `super`'s designated initializer (from your designated initializer)

or your own designated initializer (from your own convenience initializers).

This whole concept takes some getting used to.

Luckily, because of lazy instantiation, et. al., we don't need initializers that much in Objective-C.

And calling initializers is easy (it's just `alloc` plus whatever `init`ializer you can find that you like).

Basic Objective-C

- Other

  Fast enumeration: `for (MyClass *myObject in arrayOfMyObjects) { }`.

  `#define`

  `NSLog(@“show this object %@ in the console”, anObject)`

- Quiz

  What does this do?

  ```
  cardA.contents = @[cardB.contents,cardC.contents][[cardB match:@[cardC]] ? 1 : 0]
  ```

  This line has a setter, getters, method invocation, array creation and array accessing all in one.

  And lots of square brackets.

Coming Up

- Next Lecture
  - More detail about Objective-C
  - More Foundation classes (besides strings and arrays)
  - Attributed strings
- Next week …
  - Multiple MVCs in your storyboard
  - View Controller Lifecycle