

The following slides are a walkthrough of the demonstration done in class.
You will need this walkthrough to do your first homework assignment.

Green Bubbles
are just for
“information.”

Yellow Bubbles
mean “do something.”

Red Bubbles
mean “important!”

Green Bubbles
with small text is for
“minor notes.”



Launch Xcode 5 and click here to create a new project.

As you create projects, they will appear here.



Loading

No Issues



Xcode 5 can be used to develop both iOS and Mac OSX applications.

These buttons are used to select a template which Xcode 5 uses to generate some code to get you started.

Cancel

Choose a template for your new project



iOS
Application
Framework & Library
Other



OS X
Application
Framework & Library
Application Plug-in
System Plug-in
Other



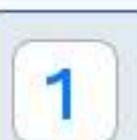
Master-Detail Application



OpenGL Game



Page-Based Application



1 Single View Application



Tabbed Application



Utility Application



Empty Application



SpriteKit Game

1 Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

Previous

Next

Click on the “Single View Application” template. It creates a simple MVC application.

Then click Next.

Our first application is going to be a Card Matching Game

These fields describe your project. We'll be filling them in during the next few slides.

Choose options for your new project:

Product Name Matchismo
Organization Name
Company Identifier
Bundle Identifier com.yourcompany.Matchismo
Class Prefix
Devices Universal

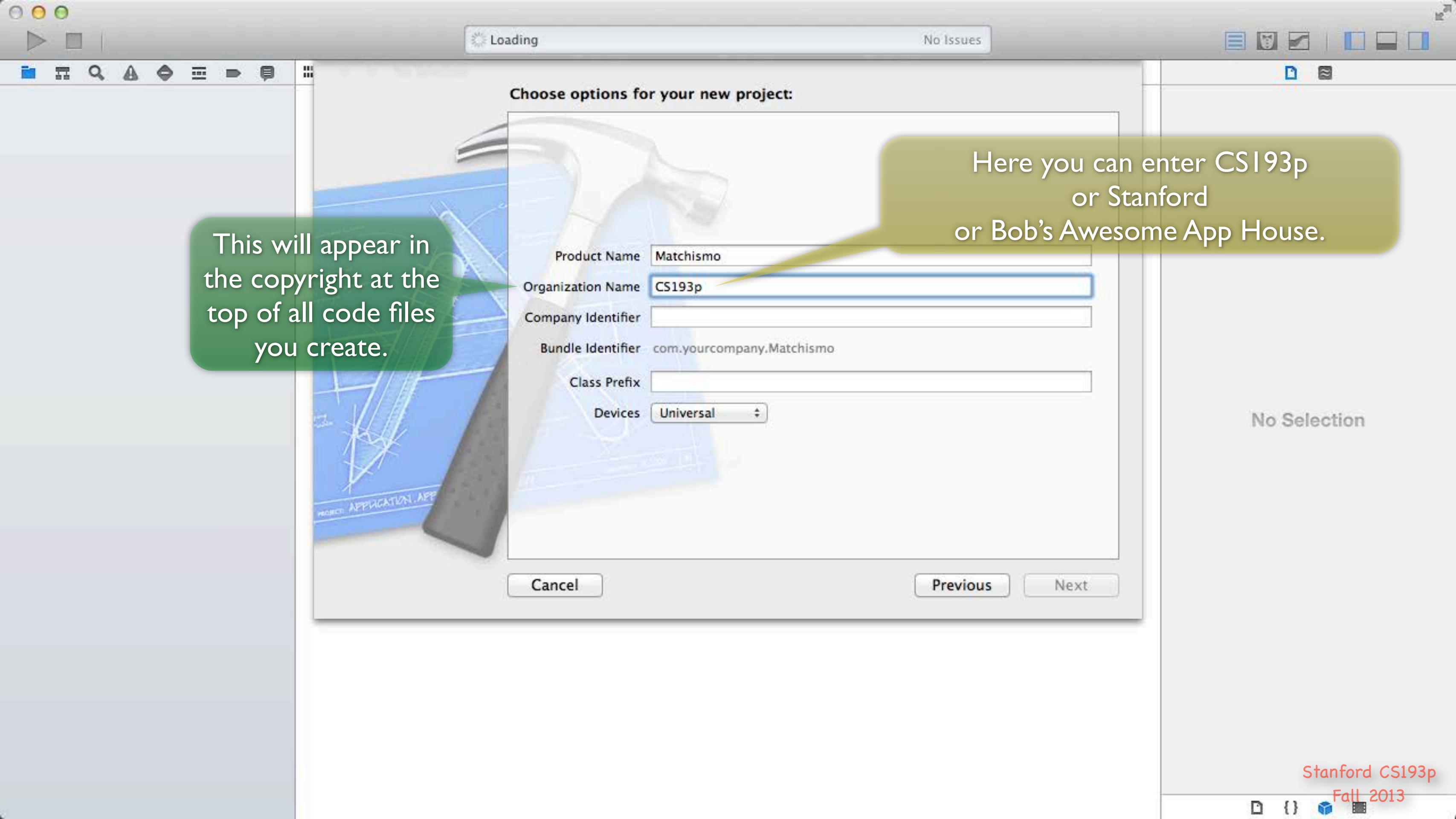
Cancel

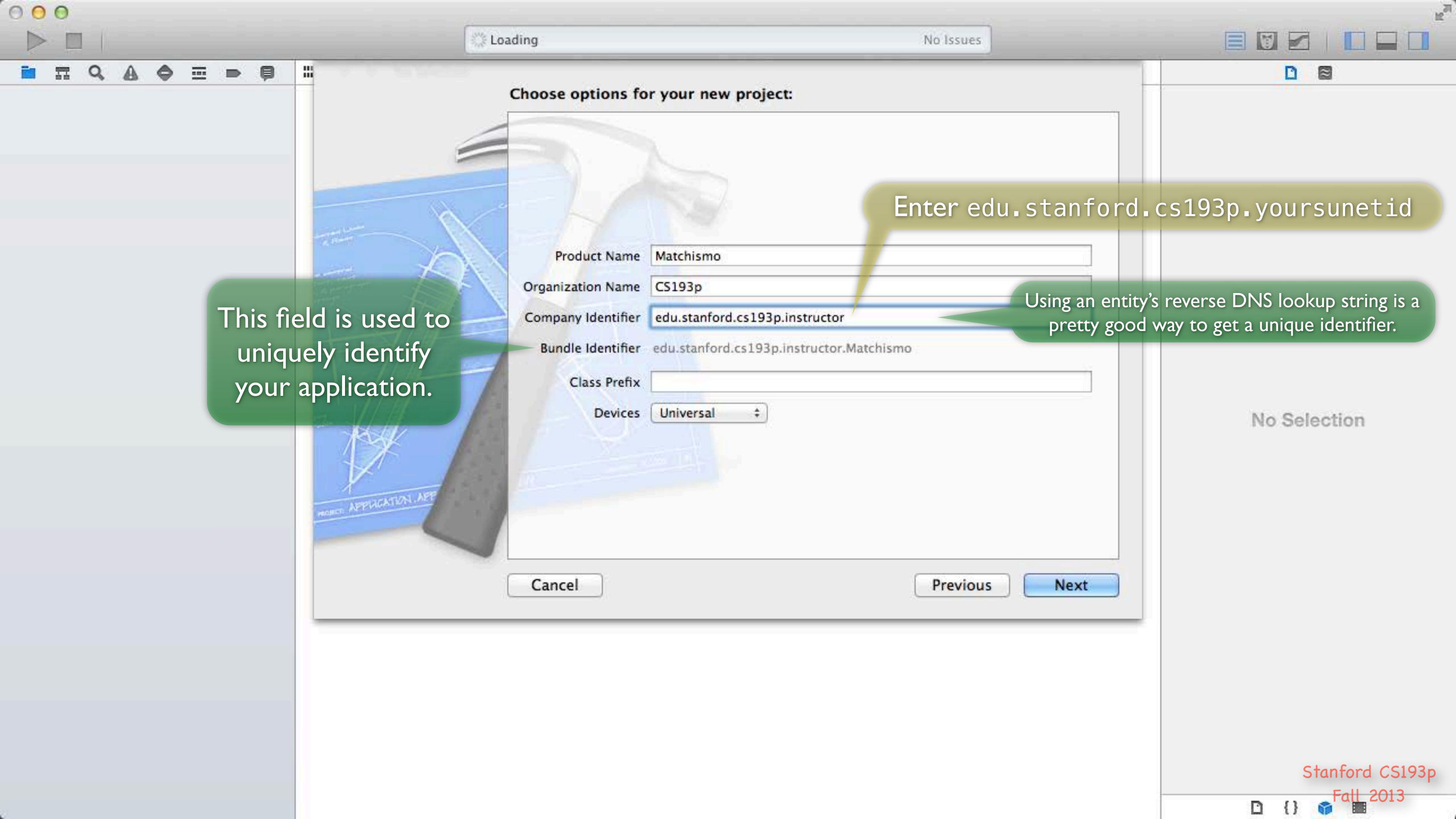
Previous

Next

No Selection

The name of our project is going to be “Matchismo” so type that in here.



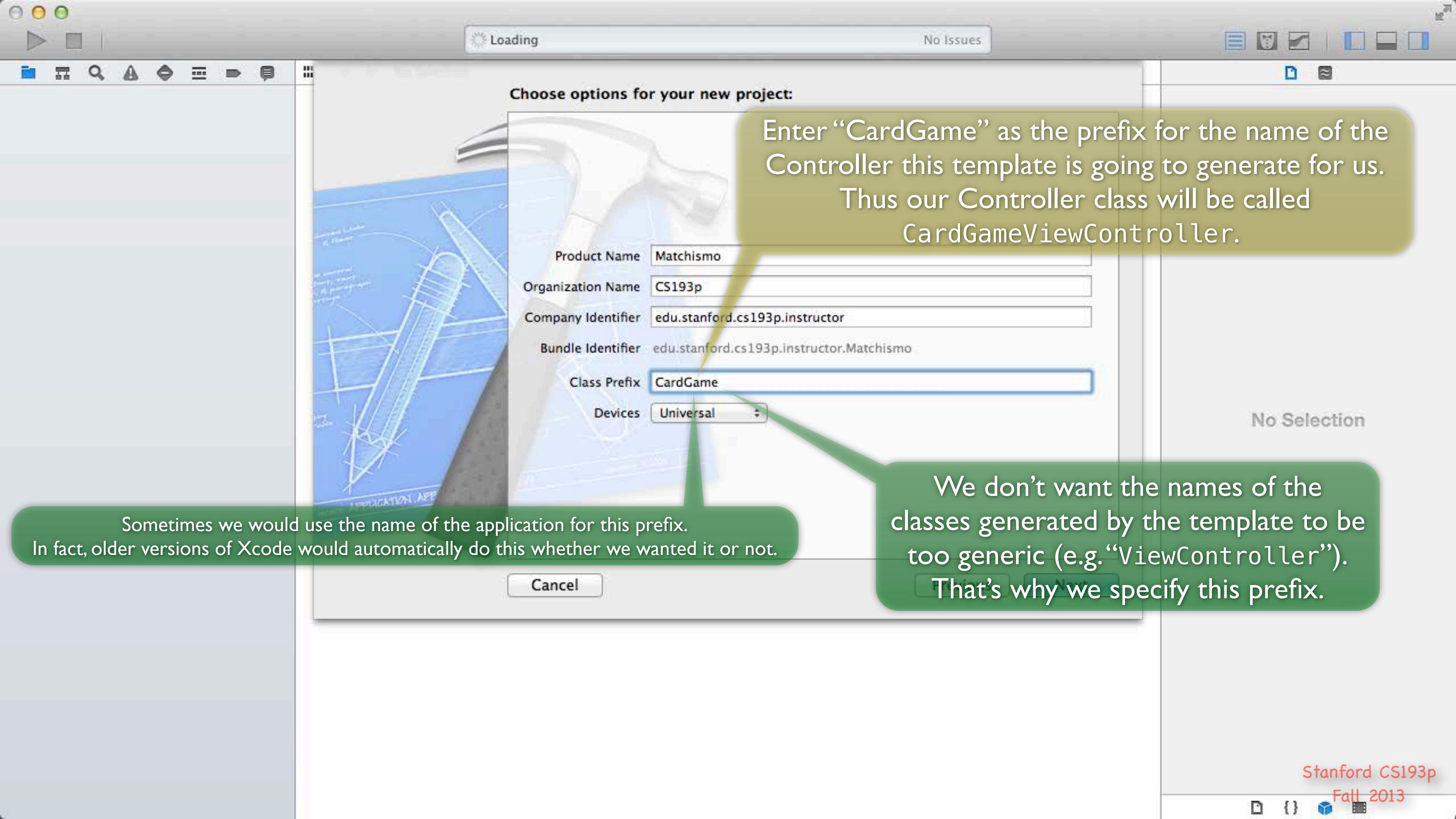


This field is used to uniquely identify your application.

Enter `edu.stanford.cs193p.yoursunetid`

Using an entity's reverse DNS lookup string is a pretty good way to get a unique identifier.

No Selection





Loading

No Issues



Choose options for your new project:

Product Name: Matchismo
Organization Name: CS193p
Company Identifier: edu.stanford.cs193p.instructor
Bundle Identifier: edu.stanford.cs193p.instructor.Matchismo

Class Prefix: iPad

Device: Universal

Cancel Previous Next

No Selection

Set the Device we're developing for to iPhone.

Our first application is going to be for the iPhone (not iPad). At least for starters.

A Universal application runs on both iPhone and iPad. In a Universal application, the iPad and the iPhone each has its own UI design (since they have different UI idioms). Xcode provides tools for designing two different UIs in the same application.

Loading

No Issues

Choose options for your new project:

Product Name Matchismo

Organization Name CS193p

Company Identifier edu.stanford.cs193p.instructor

Bundle Identifier edu.stanford.cs193p.instructor.Matchismo

Class Prefix CardGame

Devices iPhone

Cancel

Previous

Next

No Selection

Then click Next.

Xcode wants to know where to store this project's directory.

Home directory.

“Developer” folder inside the home directory.
There are no projects in it currently.

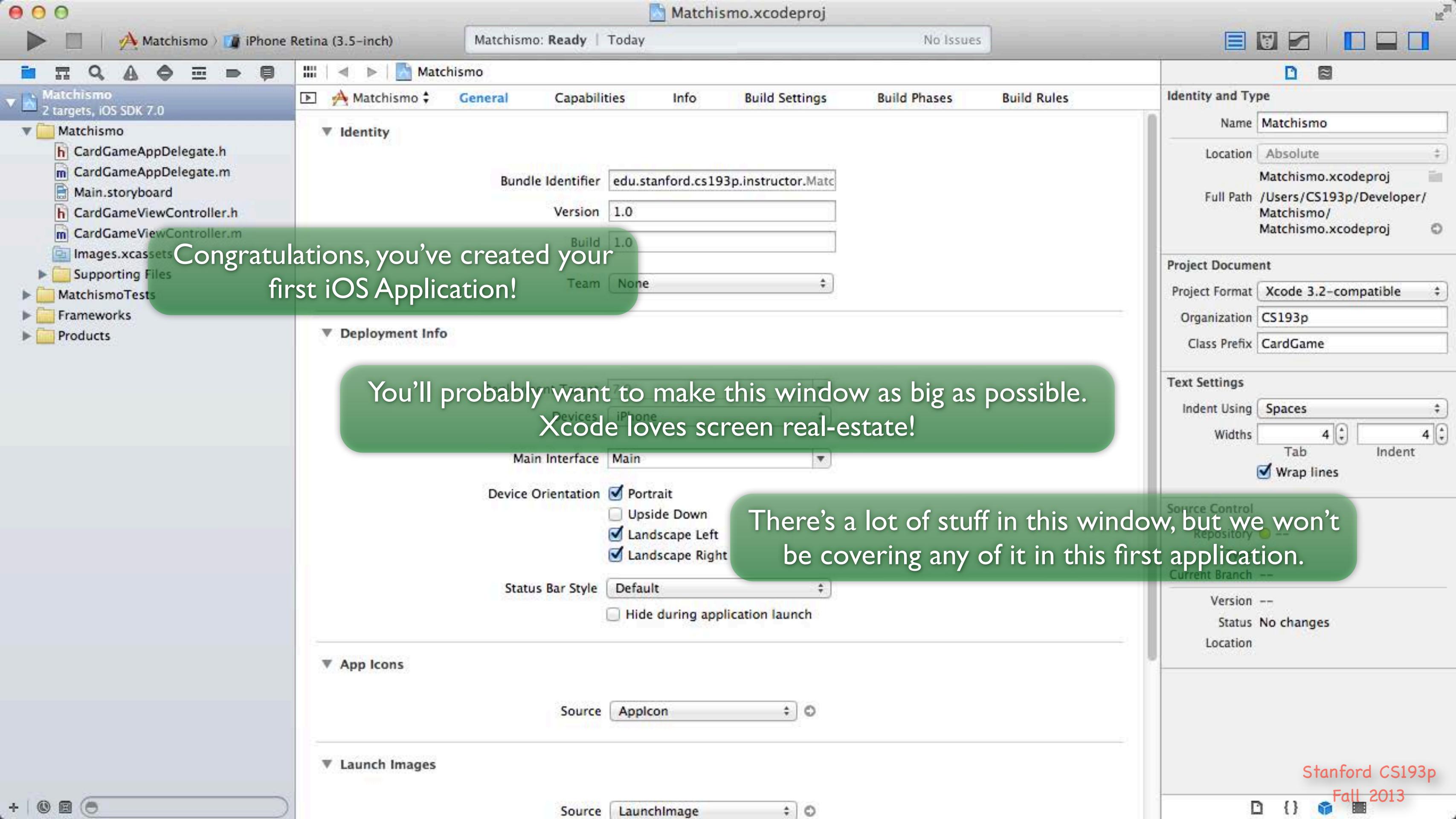
We will hopefully be covering source control in this course.
But not for this first project, so leave this switch turned off.

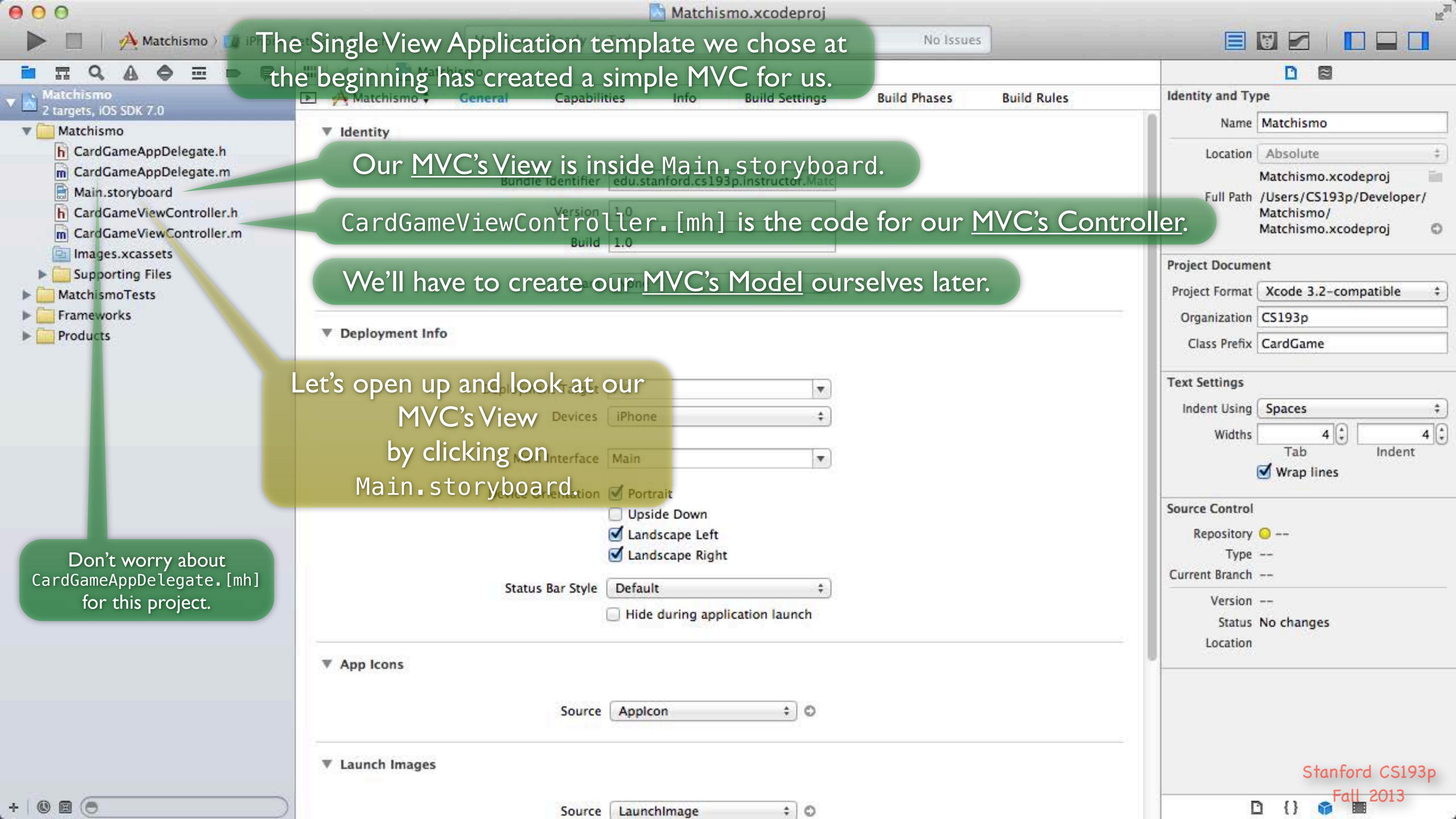
If you don't have a Developer folder in your home directory, you can create it with this New Folder button.

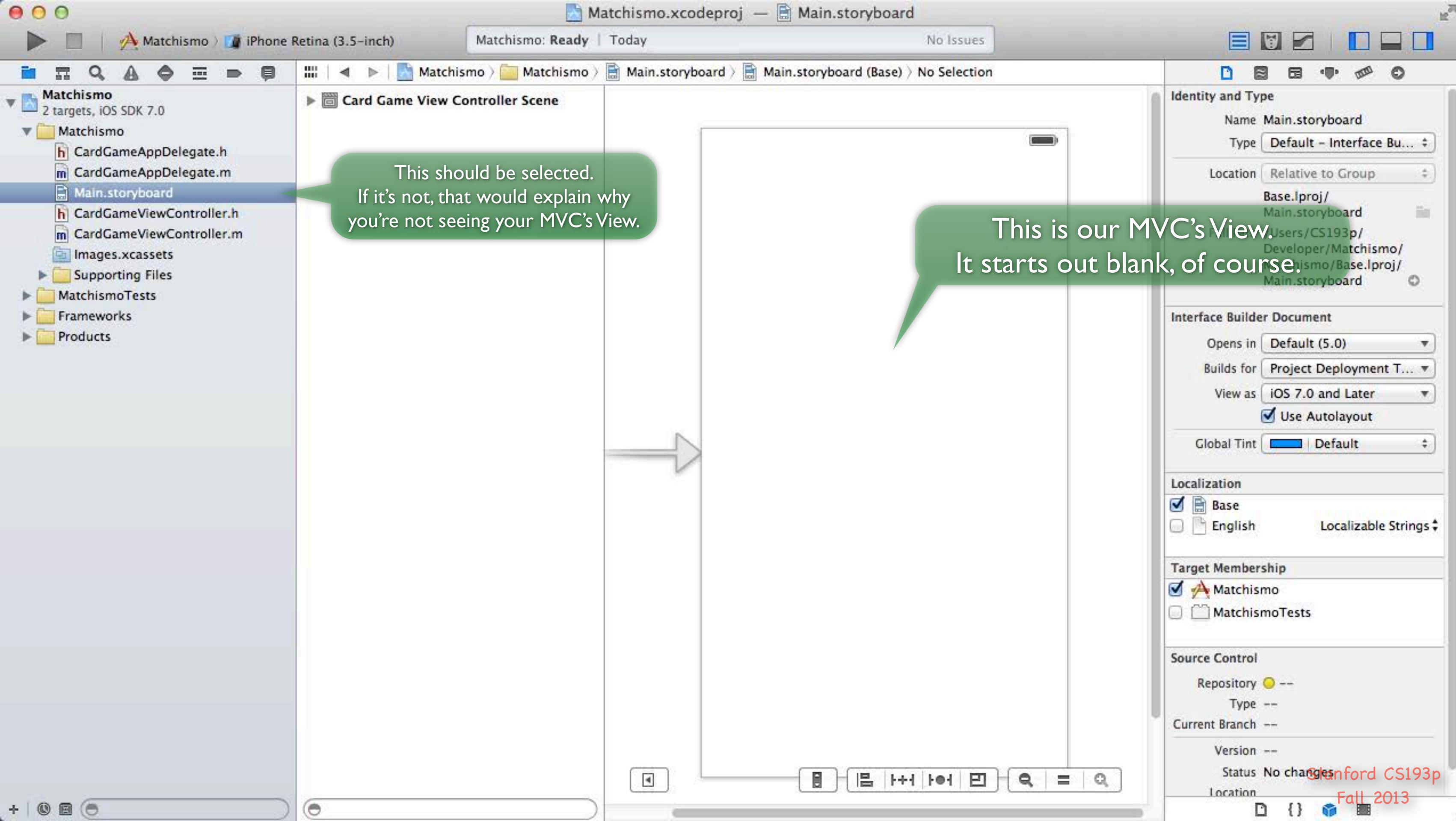
Navigate to a directory called “Developer” in your home directory (create it if needed).

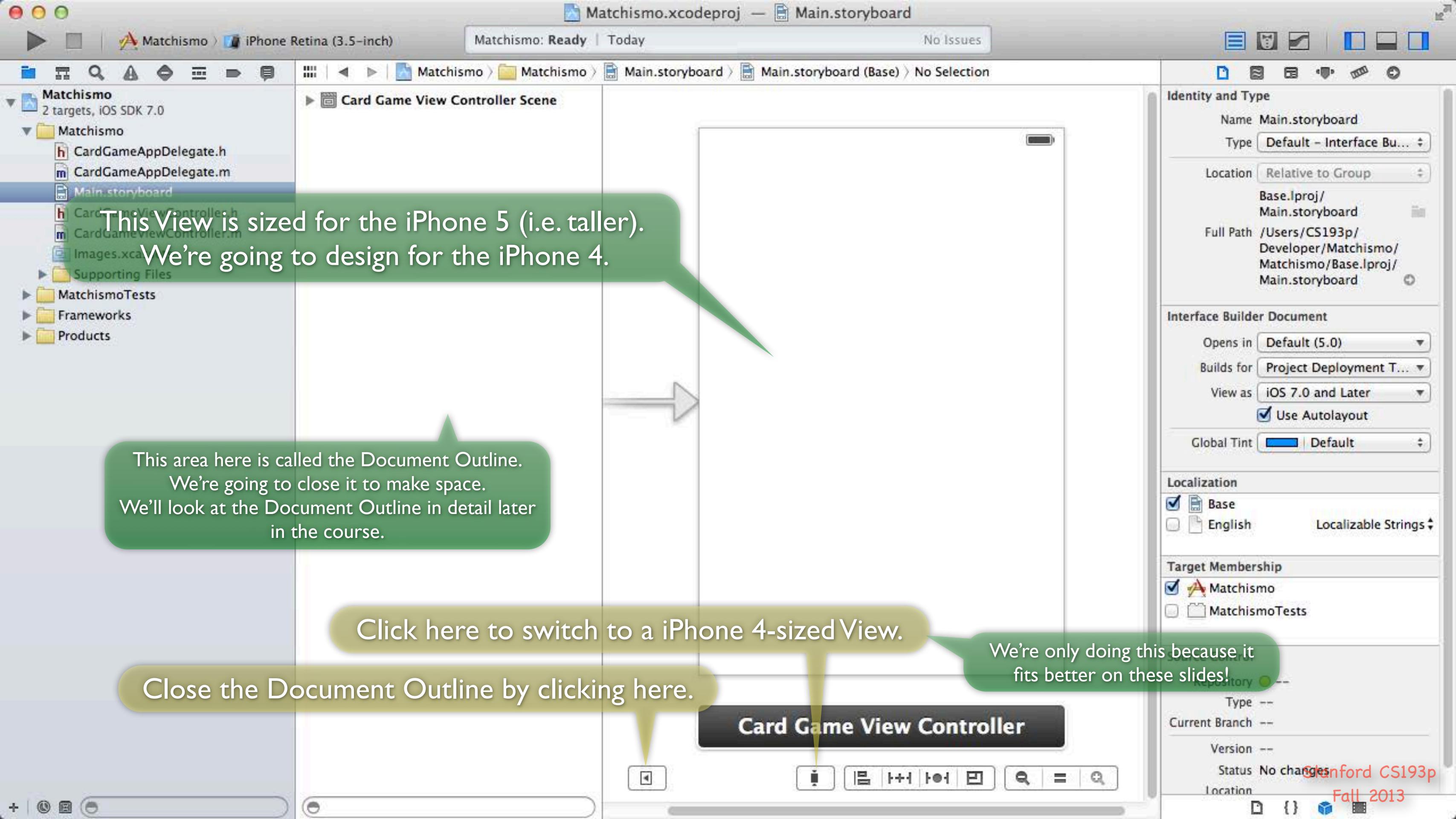
Then click Create to create your project directory inside ~/Developer.

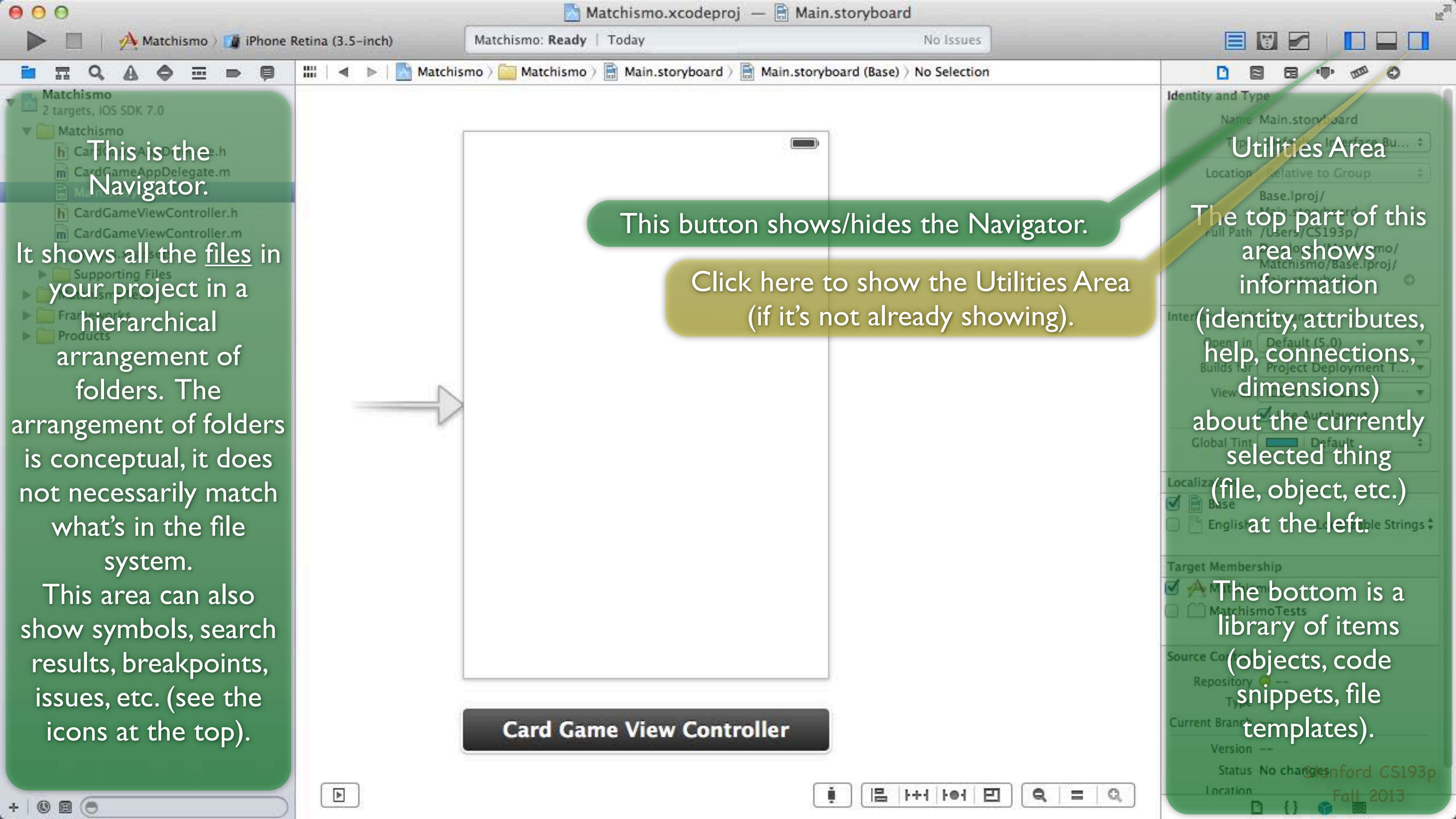












Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard (Base) No Selection

Identity and Type

Name Main.storyboard

Type Default - Interface Bu...

Location Relative to Group

Base.Iproj/
Main.storyboard

Full Path /Users/CS193p/
Developer/Matchismo/
Matchismo/Base.Iproj/
Main.storyboard

Interface Builder Document

Opens in Default (5.0)

Builds for Project Deployment T...

View as iOS 7.0 and Later

Use Autolayout

Drag this bar up (if necessary)
to expose the Library Area.

Click here (if necessary) to
select the Objects Palette.

Card Game View Controller

Stanford CS193p Fall 2013

This slide is just for reference.
Don't worry about all these details for now.

Card Game View Controller

File Inspector
Shows information about the file containing the selected item.

Quick Help
If the selected item at the left has some documentation reference, this shows a “summary” version of it.

Media Library
Images, sounds, etc.

Object Library
Buttons, text fields, controllers, etc.

File Template Library
Templates for storyboards, classes, etc.

Code Snippet Library
Snippets of code for common tasks.

Class Hierarchy

Logs
Every time you build/run, a log of it is saved.
Access old ones here.

Breakpoints
We'll cover the debugger later.

Threads
We'll cover multithreading later too.

Tests
We'll cover Unit Testing later.

Issues
Compiler warnings/errors, etc.

Search
Find/replace in files in your Project.

Matchismo.xcodeproj — Main.storyboard
Matchismo: Ready | Today
No Issues

Matchismo
2 targets, iOS SDK 7.0
Matchismo
CardGameAppDelegate.h
CardGameAppDelegate.m
Main.storyboard
CardGameViewController.h
CardGameViewController.m
Images.xcassets
Supporting Files
MatchismoTests
Frameworks
Products

Matchismo
Main.storyboard
CardGameViewController.h
CardGameViewController.m
Images.xcassets
Supporting Files
MatchismoTests
Frameworks
Products

Identity and Type
Name: Main.storyboard
Type: Default - Interface Bu...
Location: Relative to Group
Base.Iproj/
Main.storyboard
/Users/CS193p/
Developer/Matchismo/
Matchismo/Base.Iproj/
Main.storyboard

Builds for: Project Deployment T...
View as: iOS 7.0 and Later
 Use Autolayout

Stanford CS193p Winter 2013

Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard (Base) No Selection

Identity and Type

Name Main.storyboard

Type Default - Interface Bu...

Location Relative to Group

Base.Iproj/ Main.storyboard

Full Path /Users/CS193p/ Developer/Matchismo/ Matchismo/Base.Iproj/ Main.storyboard

Interface Builder Document

Opens in Default (5.0)

Builds for Project Deployment T...

View as iOS 7.0 and Later

Use Autolayout

directly available in Interface...

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text Field - Displays editable text and sends an action message to a target object when Return...

Slider Slider - Displays a continuous

Stanford CS193p Fall 2013

It's time to start building the user-interface in our MVC View. We're building a card game, so we'll start with our first "card." We'll use a button to represent it.

The Objects Palette contains a bunch of objects you can use to build your View.

Scroll down to find Button.

If you are not seeing Button in the list, try clicking on your View.

Card Game View Controller

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard (Base) No Selection

Identity and Type

- Name Main.storyboard
- Type Default – Interface Bu...
- Location Relative to Group
- Base.Iproj/ Main.storyboard
- Full Path /Users/CS193p/ Developer/Matchismo/ Matchismo/Base.Iproj/ Main.storyboard

Interface Builder Document

- Opens in Default (5.0)
- Builds for Project Deployment T...
- View as iOS 7.0 and Later
- Use Autolayout

directly available in Interface...

Label Label – A variably sized amount of static text.

Button Button – Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Text Text Field – Displays editable text and sends an action message to a target object when Return...

Slider Slider – Displays a continuous

Stanford CS193p Fall 2013

Card Game View Controller

Drag a Button from the Object Library to your View.

The screenshot shows the Xcode interface with the project 'Matchismo' selected. The storyboard file 'Main.storyboard' is open. A callout bubble with the text 'Drag a Button from the Object Library to your View.' points to the 'Button' item in the object library on the right. A large yellow arrow points from the bottom right towards the storyboard canvas.

Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo > iPhone Retina (3.5-inch)

Main.storyboard (Base) > No Selection

Identity and Type

- Name Main.storyboard
- Type Default – Interface Bu...
- Location Relative to Group
- Base.lproj/
- Main.storyboard
- Full Path /Users/CS193p/Developer/Matchismo/Matchismo/Base.lproj/Main.storyboard

Interface Builder Document

- Opens in Default (5.0)
- Builds for Project Deployment T...
- View as iOS 7.0 and Later
- Use Autolayout

directly available in Interface...

Label Label – A variably sized amount of static text.

Button Button – Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Text Text Field – Displays editable text and sends an action message to a target object when Return...

Slider Slider – Displays a continuous

Stanford CS193p Fall 2013

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Card Game View Controller

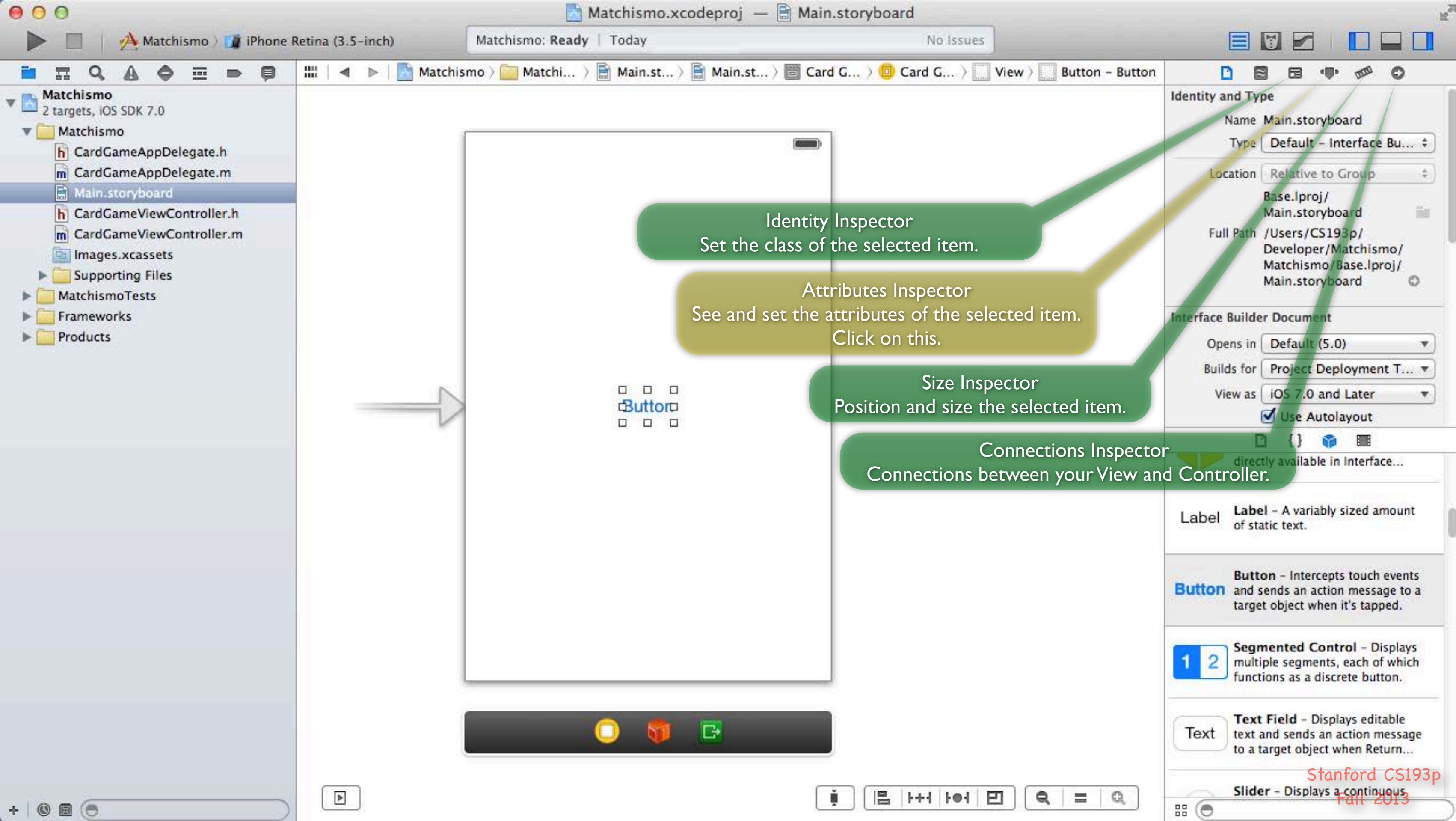
View

Drop the Button into the very center of your View.

Notice the dashed blue lines which Xcode displays as you drag which help you line things up nicely.

Buttons are instances of the class `UIButton` in the iOS SDK.

The screenshot shows the Xcode interface with the Main.storyboard file open. On the left is the project navigator with files like CardGameAppDelegate.h, CardGameViewController.h, and Images.xcassets. The main canvas shows a view controller with a placeholder for a button. A green callout points to a dashed blue alignment guide, explaining its use for alignment. A yellow callout points to a button placeholder, instructing to drop it into the center. A purple callout points to the UIButton class name in the documentation. The right side shows the Identity and Type inspector with details about the storyboard, and the Utilities inspector with various interface elements like labels and buttons.



Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo > iPhone Retina (3.5-inch)

Main.storyboard

Card Game View Controller

Card Game View Controller

View

Button - Button

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

There are all kinds of attributes about the button you can set. We'll do this in a moment.

Drag this back down to make more room for the Attributes.

Button

Type System

State Config Default

Title Plain

Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background Default Background Image

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Drawing Shows Touch On Highlight

directly available in Interface...

Label Label - A variably sized amount of static text.

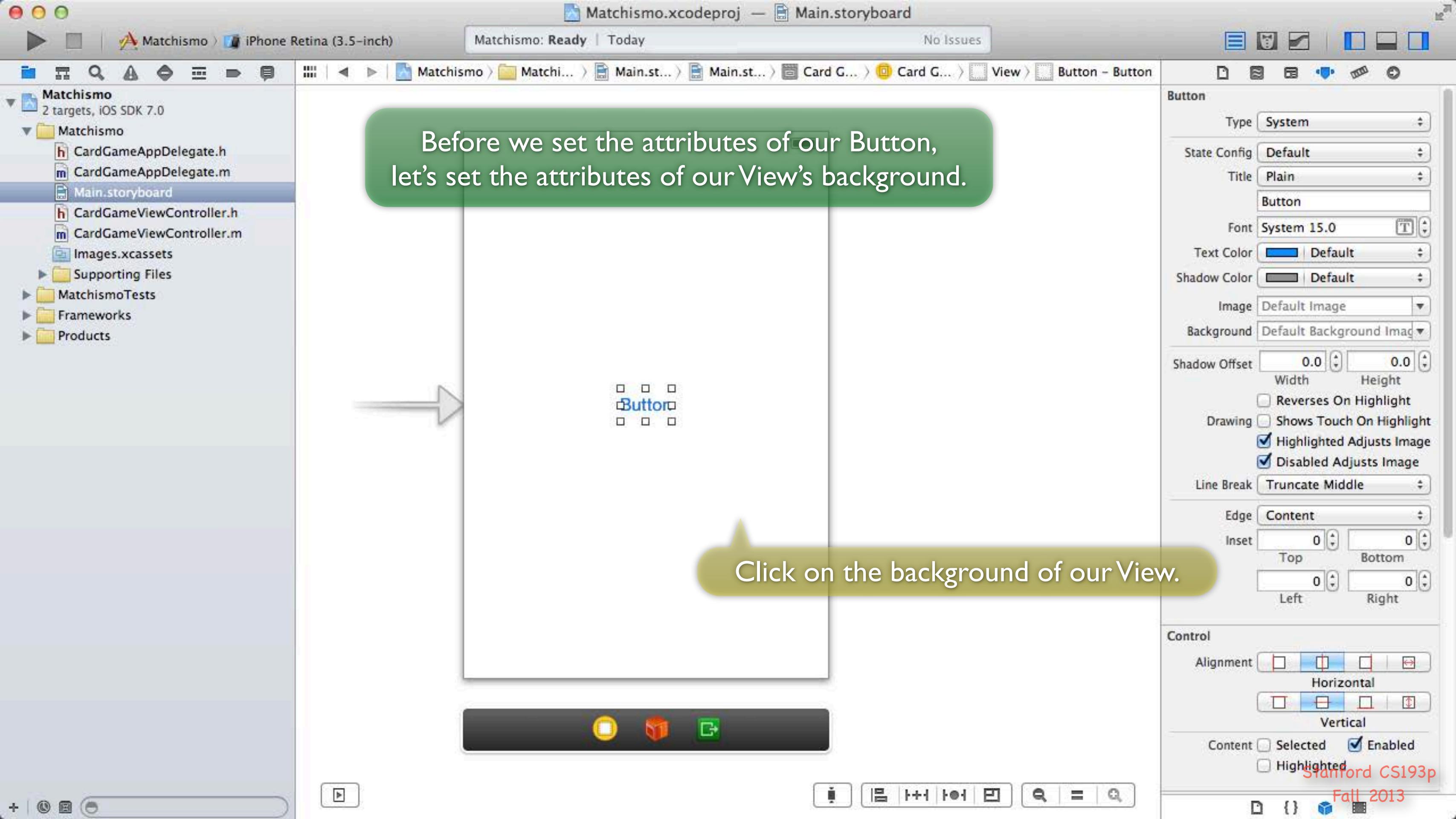
Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text - Displays editable text and sends an action message to a target object when Return...

Slider - Displays a continuous

Stanford CS193p Fall 2013



Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard

Card Game View Controller

View

Mode Scale To Fill

Tag 0

User Interaction Enabled

Multiple Touch

Alpha 1

Background White Color

Tint Default

Opaque

Hidden

Clears Graphics Context

Clip Subviews

Autoresizes Subviews

Stretching X 0 Y 0

Width 1 Height 1

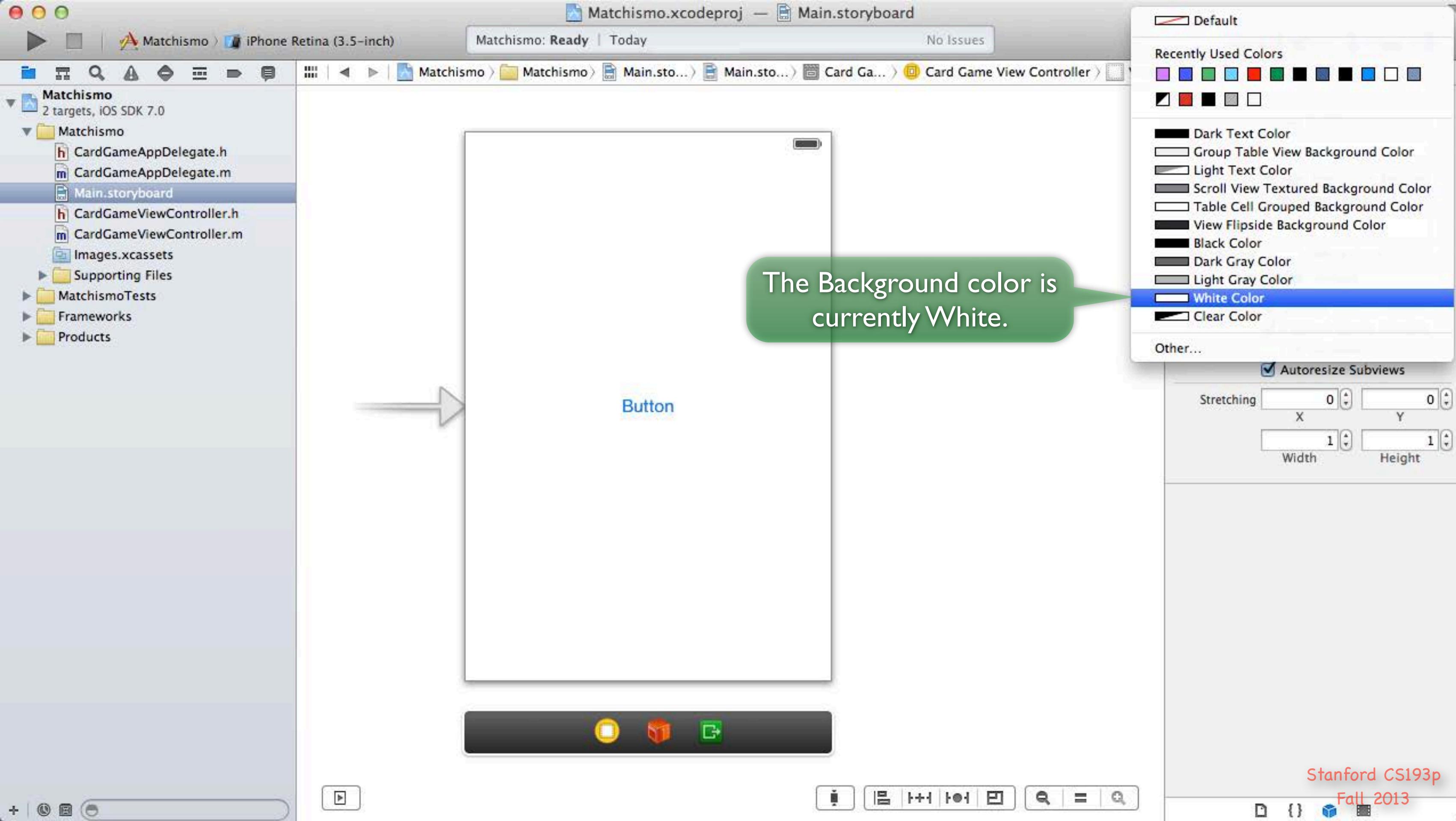
This is what buttons look like usually in iOS 7.
Think of them sort of like links in a web page.

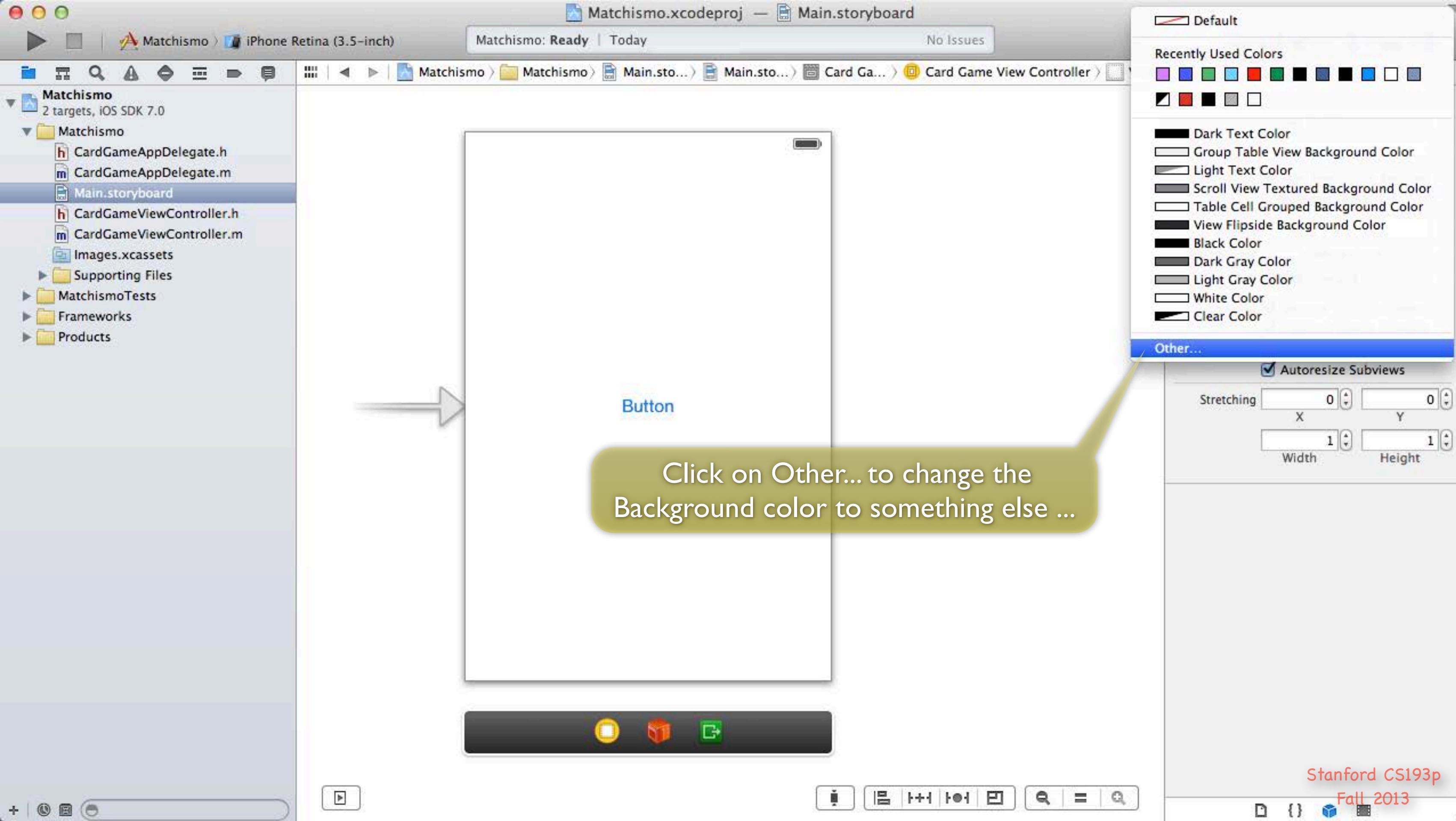
We're going to change the look of the button dramatically
(to look like a card) using images, but that's actually somewhat unusual.

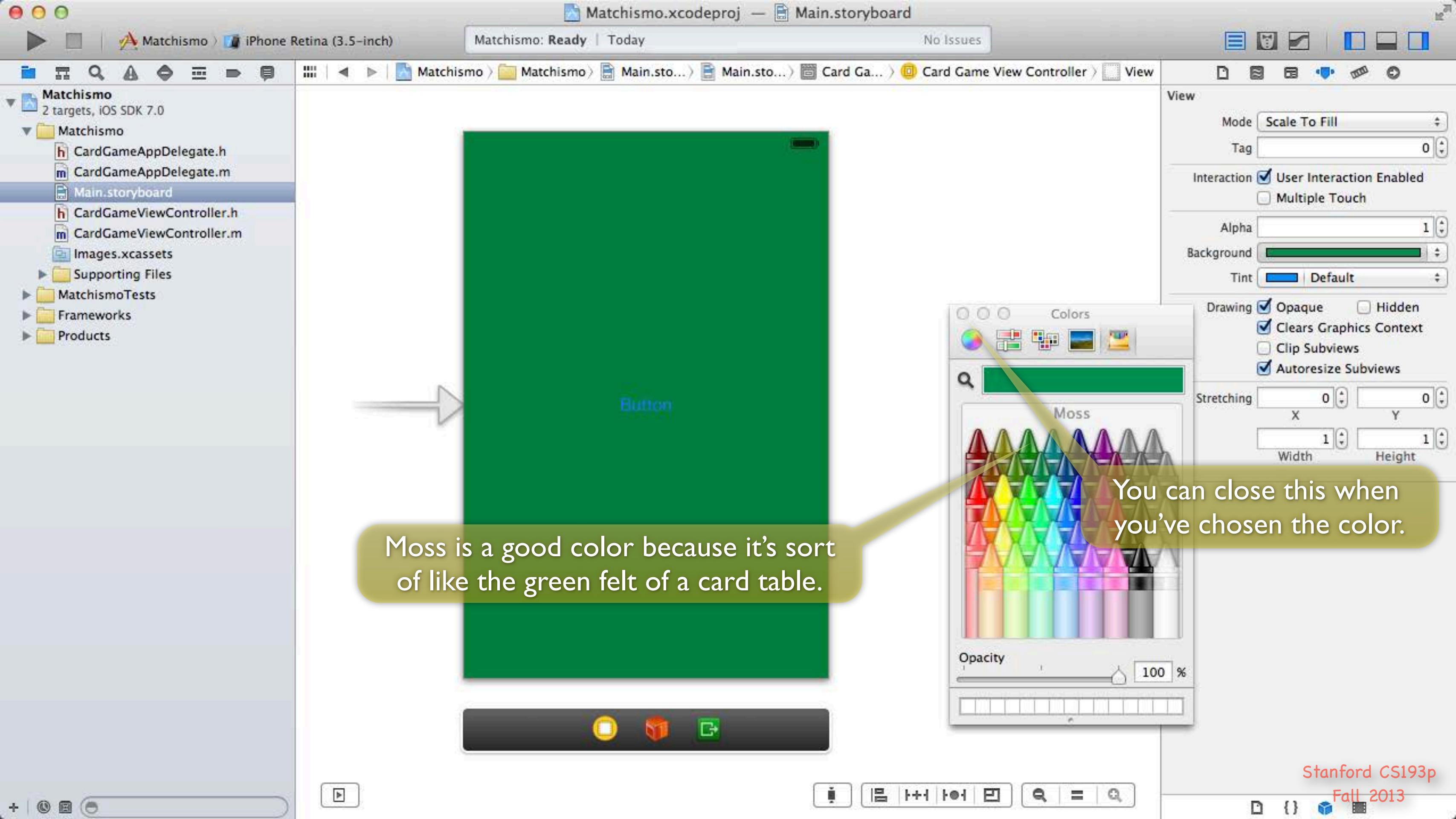
Hopefully the Attributes Inspector now
shows the attributes of the area which is
at the root of our MVC's View.

Let's change the Background
color of the root of our
MVC's View by clicking here ...

Stanford CS193p
Fall 2013







Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo

iPhone Retina (3.5-inch)

Matchismo: Ready | Today No Issues

Matchismo

Main.storyboard

Card Game

View

Button - Button

Button

Type System

State Config Default

Title Plain

Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background Default Background Image

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Drawing Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

Left Right

Control

Alignment Horizontal

Vertical

Content Selected Enabled

Highlighted

Stanford CS193p

Fall 2013

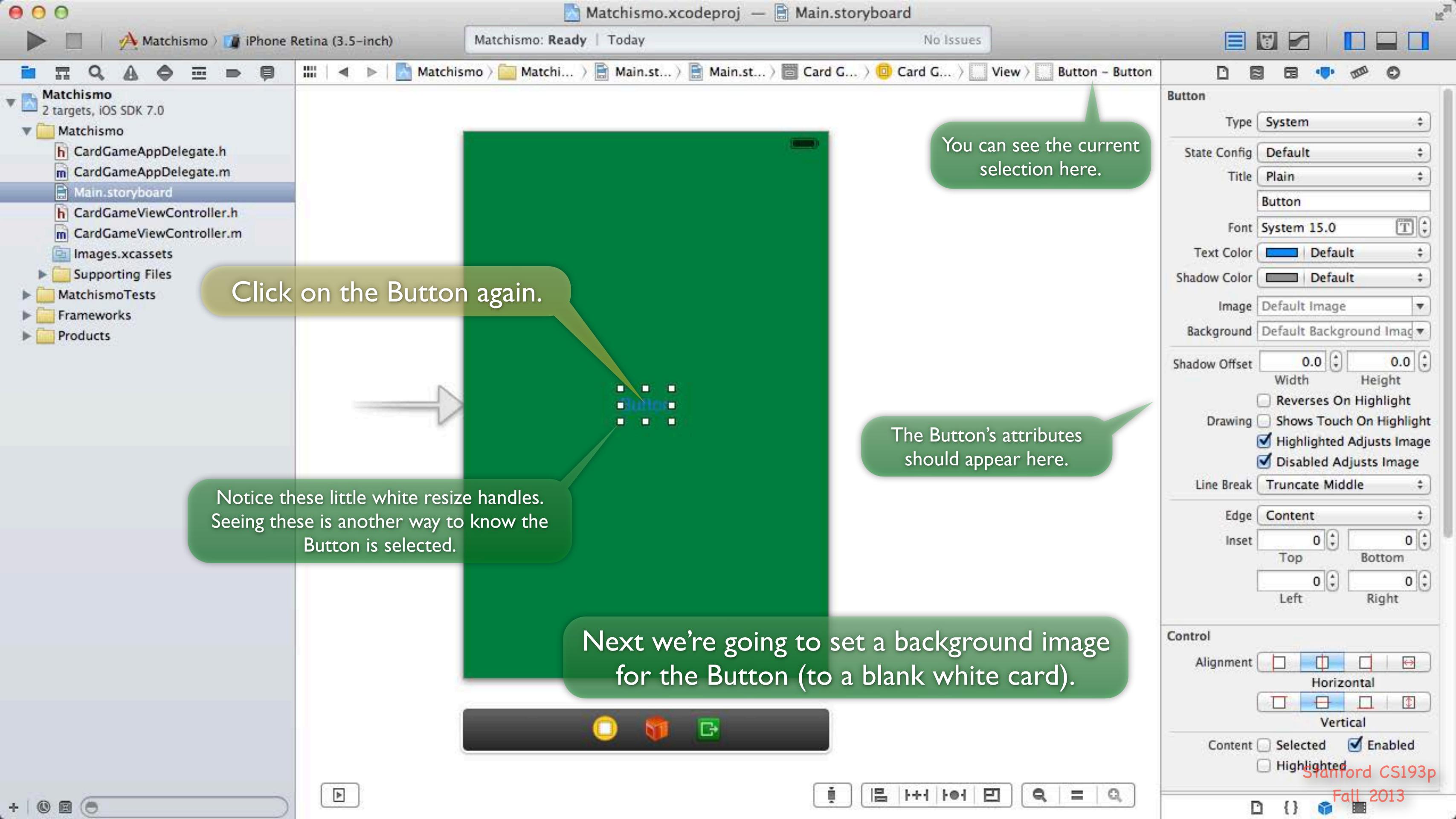
Click on the Button again.

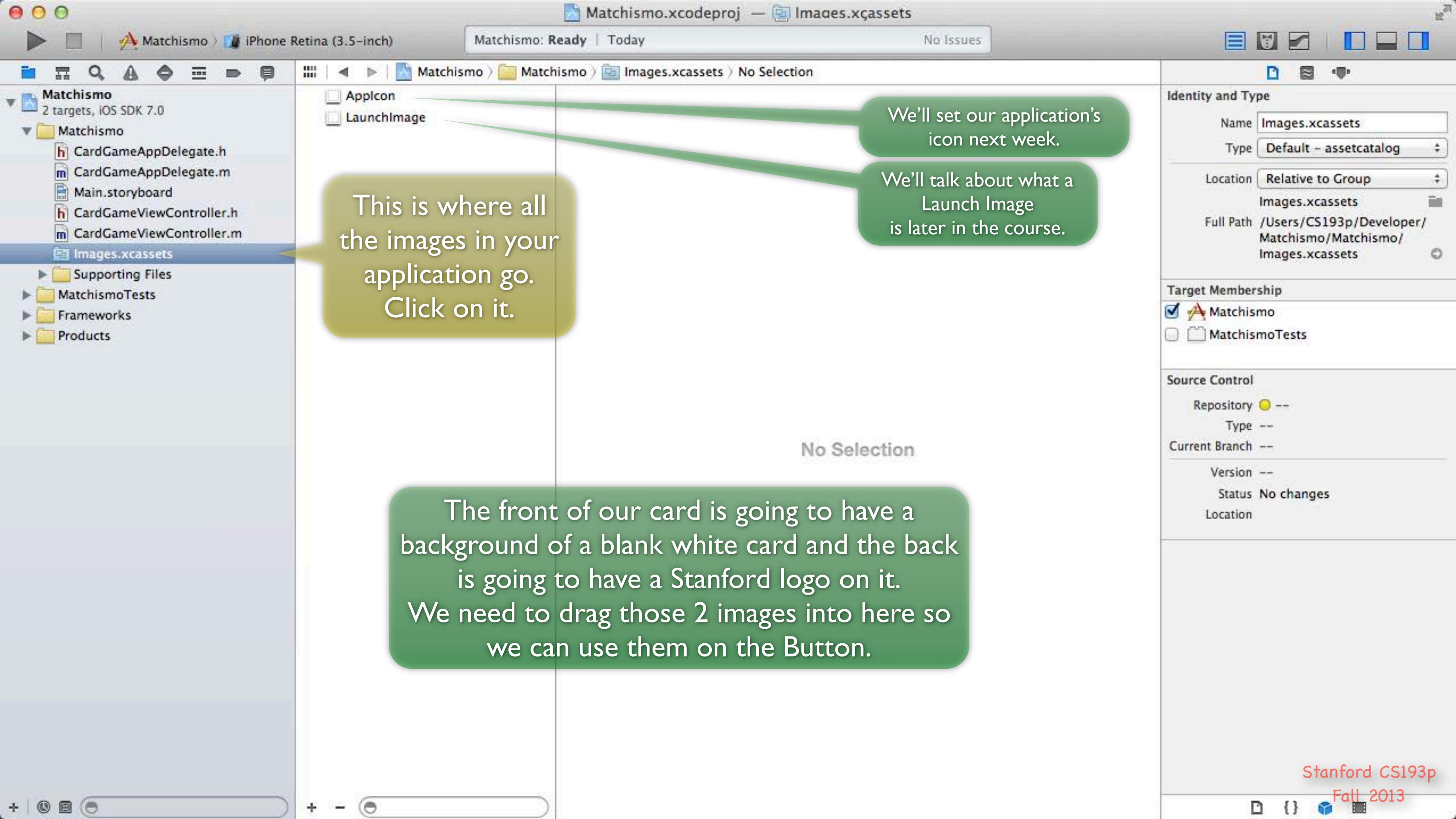
You can see the current selection here.

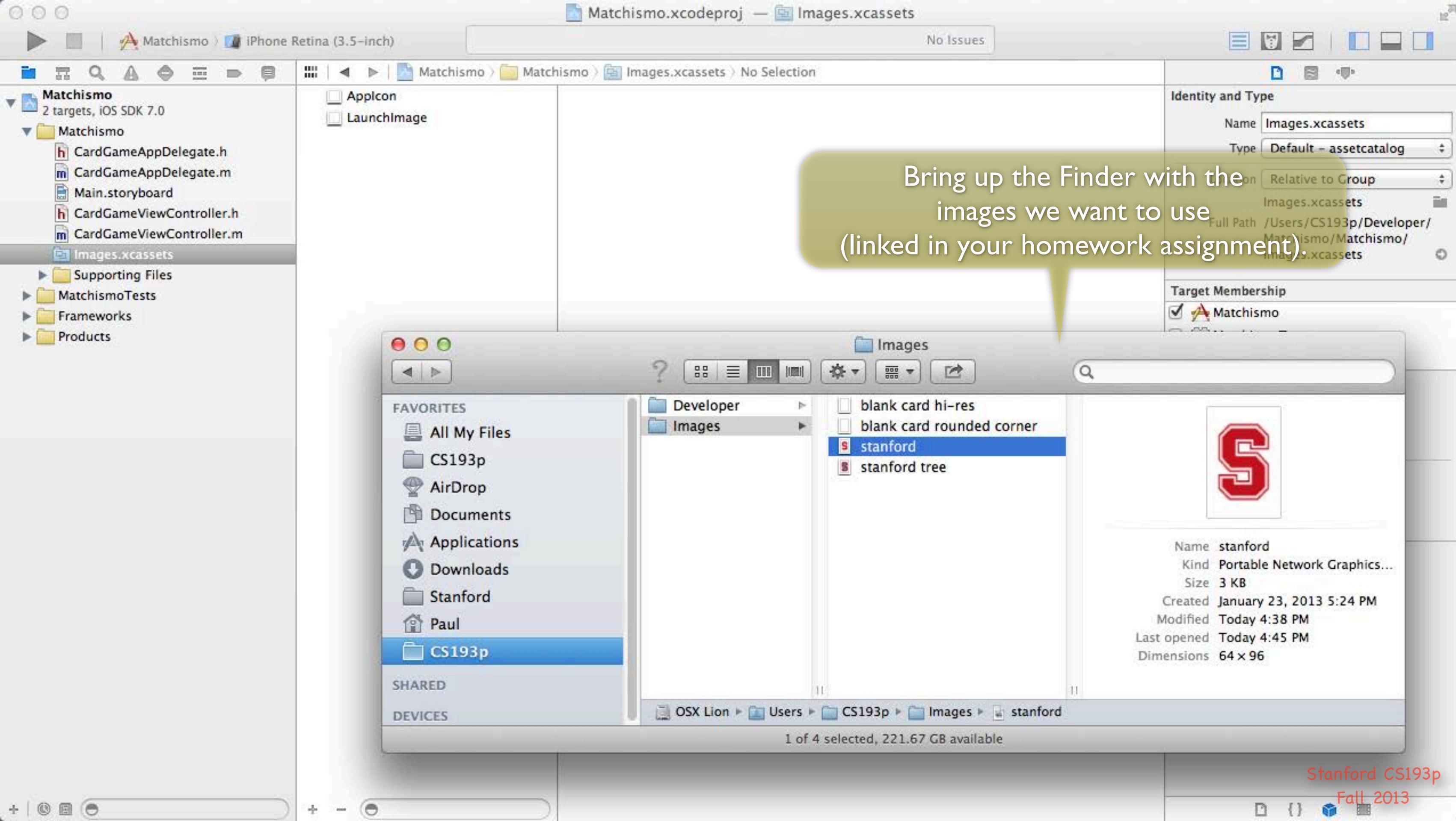
Notice these little white resize handles. Seeing these is another way to know the Button is selected.

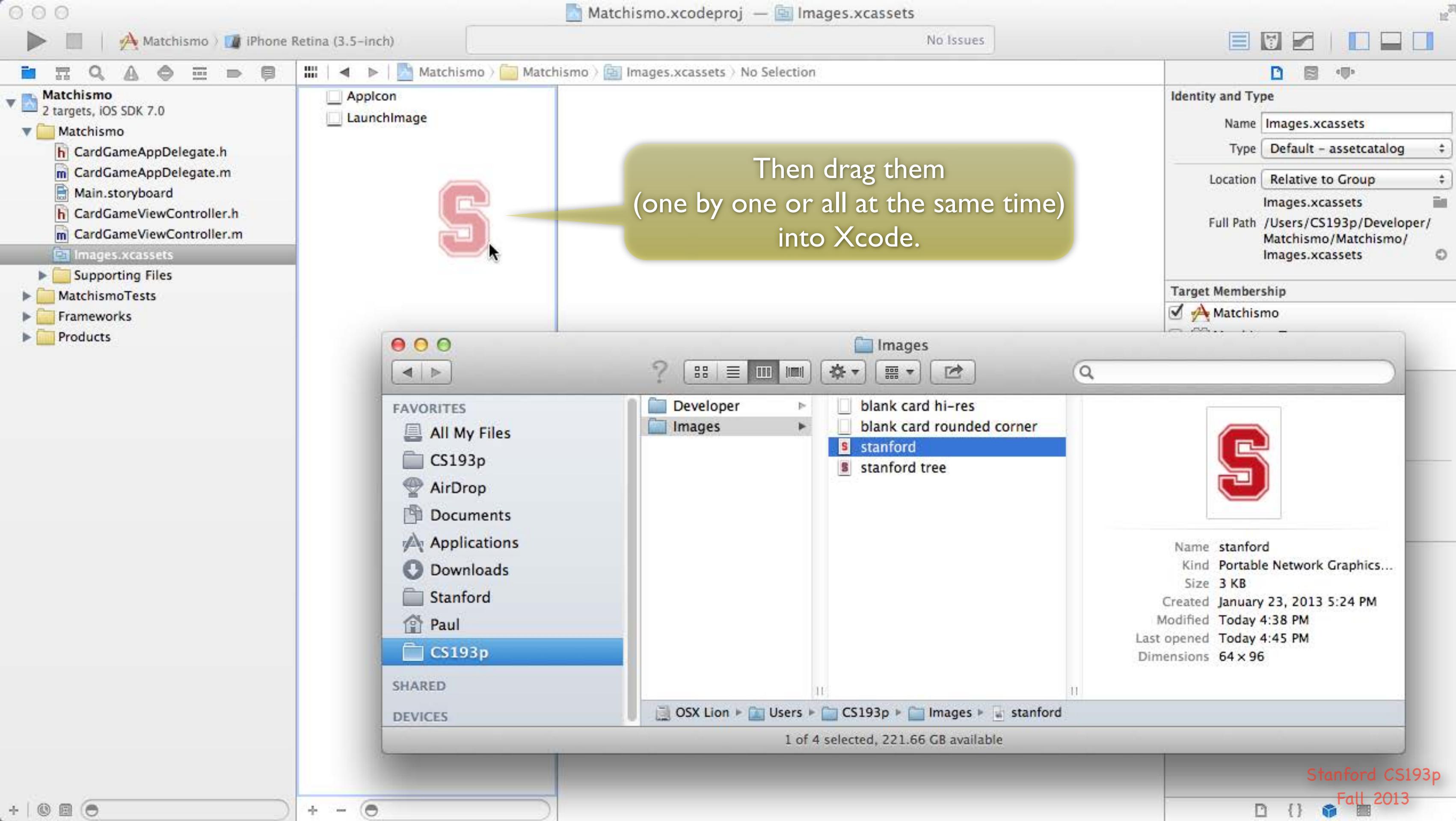
The Button's attributes should appear here.

Next we're going to set a background image for the Button (to a blank white card).









Matchismo.xcodeproj — Images.xcassets

No Issues

Matchismo > iPhone Retina (3.5-inch)

Matchismo > Matchismo > Images.xcassets > stanford

Identity and Type

- Name Images.xcassets
- Type Default - assetcatalog
- Location Relative to Group
- Images.xcassets
- Full Path /Users/CS193p/Developer/Matchismo/Matchismo/Images.xcassets

Target Membership

- Matchismo

stanford

blank card rounded corner

1x 2x

Universal

Images

FAVORITES

- All My Files
- CS193p
- AirDrop
- Documents
- Applications
- Downloads
- Stanford
- Paul
- CS193p

SHARED

DEVICES

Developer Images

blank card hi-res

blank card rounded corner

stanford

stanford tree

Name blank card rounded corner

Kind Portable Network Graphics...

Size 2 KB

Created January 23, 2013 5:24 PM

Modified Yesterday 11:34 AM

Last opened Yesterday 11:34 AM

Dimensions 64 x 96

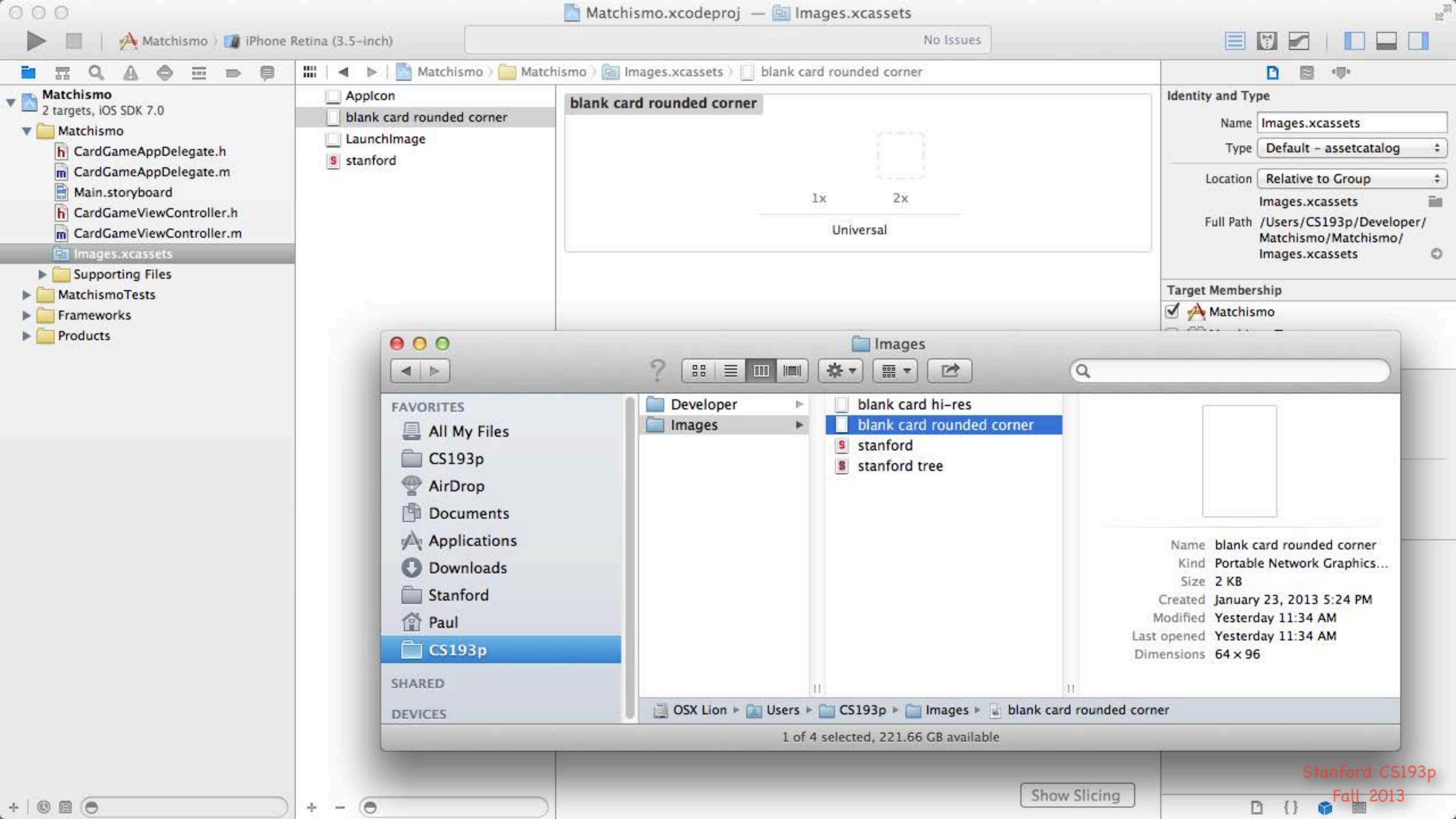
OSX Lion > Users > CS193p > Images > blank card rounded corner

1 of 4 selected, 221.66 GB available

Show Slicing

Stanford CS193p

Fall 2013



Matchismo.xcodeproj — Images.xcassets

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

blank card rounded corner

AppIcon blank card rounded corner LaunchImage stanford

blank card rounded corner

1x 2x Universal

You may not want the name of the image
(as referred to in your code)
to be the same as the name of the file.
You can simply double-click on it
and change the name.

Identity and Type

- Name Images.xcassets
- Type Default - assetcatalog
- Location Relative to Group
- Full Path /Users/CS193p/Developer/Matchismo/Matchismo/Images.xcassets

Target Membership

- Matchismo
- MatchismoTests

Source Control

- Repository --
- Type --
- Current Branch --
- Version --
- Status No changes
- Location

Show Slicing

Stanford CS193p Fall 2013

The screenshot shows the Xcode interface with the project 'Matchismo' selected. In the center, the 'Images.xcassets' editor is open, displaying a list of assets: AppIcon, blank card rounded corner, LaunchImage, and stanford. A callout bubble originates from the text 'You may not want the name of the image' and points to the 'blank card rounded corner' item in the list. The 'blank card rounded corner' asset is highlighted with a blue selection bar. The right-hand panel shows the 'Identity and Type' settings for the selected asset, including its name, type, location, and target membership. The 'Source Control' section is also visible. At the bottom right, there is a red watermark reading 'Stanford CS193p Fall 2013'.

Matchismo.xcodeproj — Images.xcassets

Matchismo: Ready | Today No Issues

Matchismo > iPhone Retina (3.5-inch)

Matchismo: cardfront

Images.xcassets

cardfront

AppIcon

cardfront

LaunchImage

stanford

1x 2x

Universal

We'll use the name "cardfront" to refer to the button's background when the card is "face up".

Show Slicing

Identity and Type

Name Images.xcassets

Type Default - assetcatalog

Location Relative to Group

Images.xcassets

Full Path /Users/CS193p/Developer/Matchismo/Matchismo/Images.xcassets

Target Membership

Matchismo

MatchismoTests

Source Control

Repository --

Type --

Current Branch --

Version --

Status No changes

Location

Stanford CS193p

Fall 2013

Matchismo.xcodeproj — Images.xcassets

Matchismo: Ready | Today No Issues

Matchismo > iPhone Retina (3.5-inch)

Matchismo

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo > Matchismo > Images.xcassets > stanford

stanford

AppIcon

cardfront

LaunchImage

stanford

S

1x 2x

Universal

We'll change the name of the image on the back of the card from "stanford" to ...

Identity and Type

Name Images.xcassets

Type Default - assetcatalog

Location Relative to Group

Images.xcassets

Full Path /Users/CS193p/Developer/Matchismo/Matchismo/Images.xcassets

Target Membership

Matchismo

MatchismoTests

Source Control

Repository --

Type --

Current Branch --

Version --

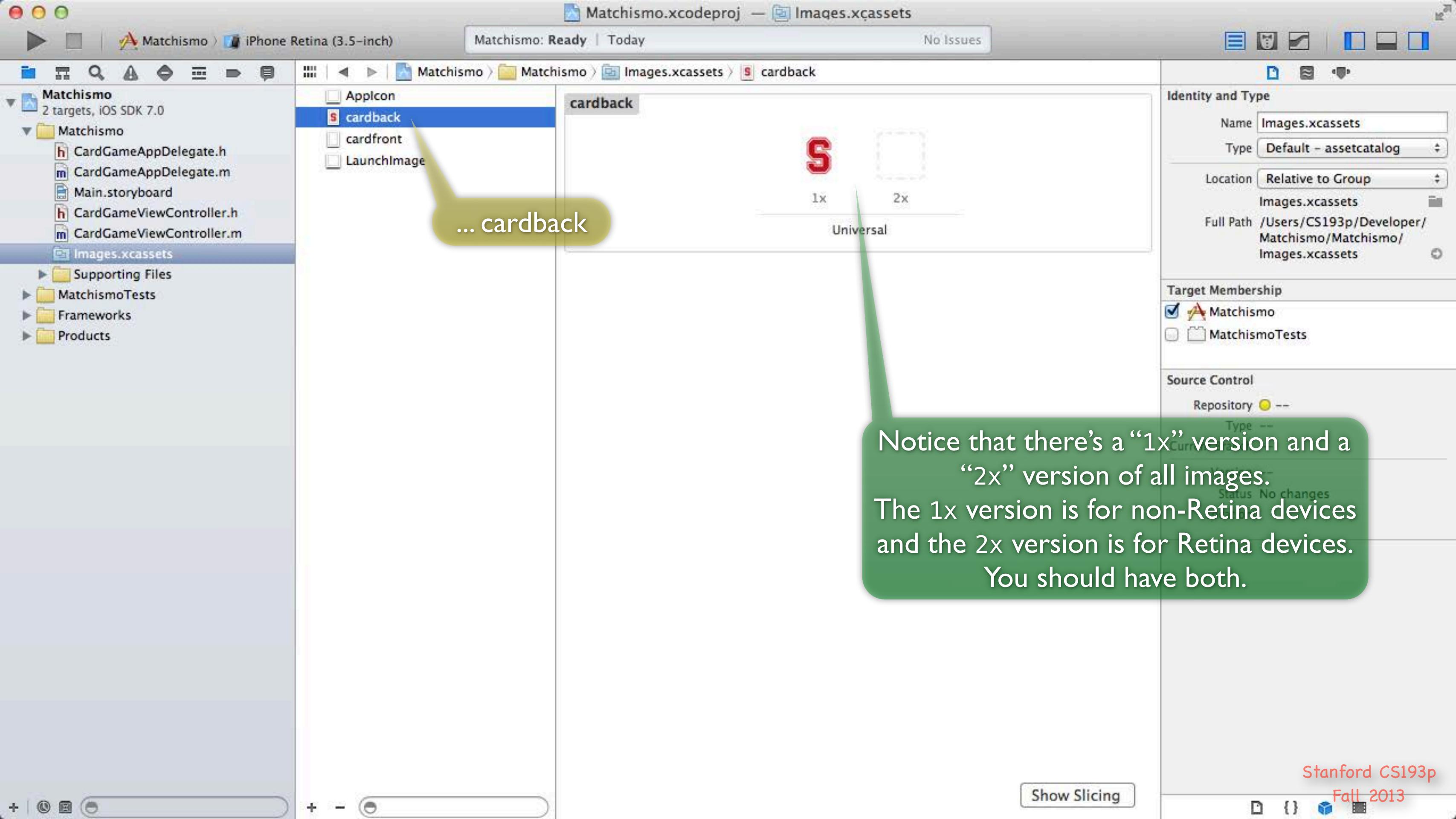
Status No changes

Location

Show Slicing

Stanford CS193p

Fall 2013



Notice that there's a “1x” version and a “2x” version of all images. The 1x version is for non-Retina devices and the 2x version is for Retina devices. You should have both.

Matchismo.xcodeproj — Images.xcassets

No Issues

Matchismo iPhone Retina (3.5-inch)

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

Items can also be deleted from this list by right clicking on them and choosing Remove.

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo

Matchismo

Images.xcassets

cardback

AppIcon

cardback

cardfront

LaunchImage

cardback

1x

2x

Universal

Identity and Type

Type Default assetcatalog

Location Relative to Group

Full Path /Users/CS193p/Developer/Matchismo/Matchismo/Images.xcassets

Target Membership Matchismo

Drag the higher resolution version into the 2x slot.

Sometimes the Retina version might be able to show more detail.

Showing the tree here is sort of an extreme example just to demonstrate this.

Hi-res Stanford Logo.

These buttons also can be used to add/remove images.

blank card hi-res

blank card rounded corner

stanford

stanford tree

Name stanford tree

Kind Portable Network Graphics...

Size 12 KB

Created Today 2:42 PM

Modified Today 2:49 PM

Last opened Today 4:45 PM

Dimensions 128 x 192

Applications

Downloads

Stanford

Paul

CS193p

Shared

Devices

OSX Lion > Users > CS193p > Images > stanford tree

1 of 4 selected, 221.65 GB available

Show Slicing

Stanford CS193p

Fall 2013

Matchismo.xcodeproj — Images.xcassets

No Issues

Matchismo iPhone Retina (3.5-inch)

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo

Images.xcassets

Matchismo

Images.xcassets

cardback

AppIcon

cardback

cardfront

LaunchImage

cardback

1x 2x

Universal

Identity and Type

Name Images.xcassets

Type Default - assetcatalog

Location Relative to Group

Images.xcassets

Full Path /Users/CS193p/Developer/Matchismo/Matchismo/Images.xcassets

Target Membership

Matchismo

Now you can see both versions.

Images

FAVORITES

All My Files

CS193p

AirDrop

Documents

Applications

Downloads

Stanford

Paul

CS193p

SHARED

DEVICES

Developer

Images

blank card hi-res

blank card rounded corner

stanford

stanford tree

stanford tree

Name stanford tree

Kind Portable Network Graphics...

Size 12 KB

Created Today 2:42 PM

Modified Today 2:49 PM

Last opened Today 4:45 PM

Dimensions 128 x 192

OSX Lion > Users > CS193p > Images > stanford tree

1 of 4 selected, 221.65 GB available

Show Slicing

Stanford CS193p

Fall 2013

Matchismo.xcodeproj — Images.xcassets

No Issues

Matchismo iPhone Retina (3.5-inch)

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo > Matchismo > Images.xcassets > cardfront

cardfront

AppIcon

cardback

cardfront

LaunchImage

blank card hi-res

1x

2x

Universal

Identity and Type

Name Images.xcassets

Type Default - assetcatalog

Location Relative to Group

Images.xcassets

Let's do the same for the hi-res version of the background of the front of the card.

blank card hi-res

Developer

Images

blank card rounded corner

stanford

stanford tree

FAVORITES

All My Files

CS193p

AirDrop

Documents

Applications

Downloads

Stanford

Paul

CS193p

SHARED

DEVICES

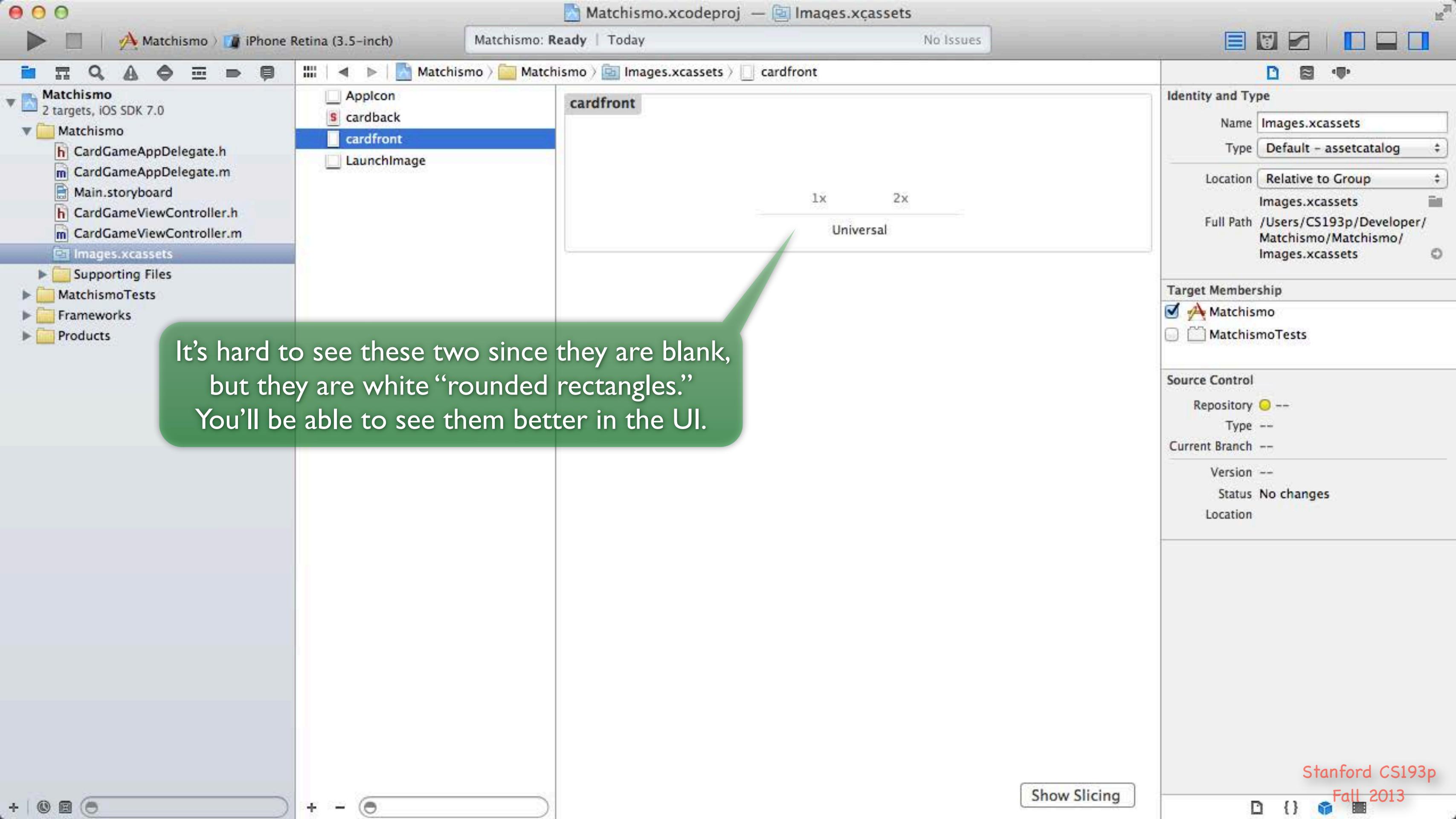
OSX Lion > Users > CS193p > Images > blank card hi-res

1 of 4 selected, 221.65 GB available

Show Slicing

Stanford CS193p

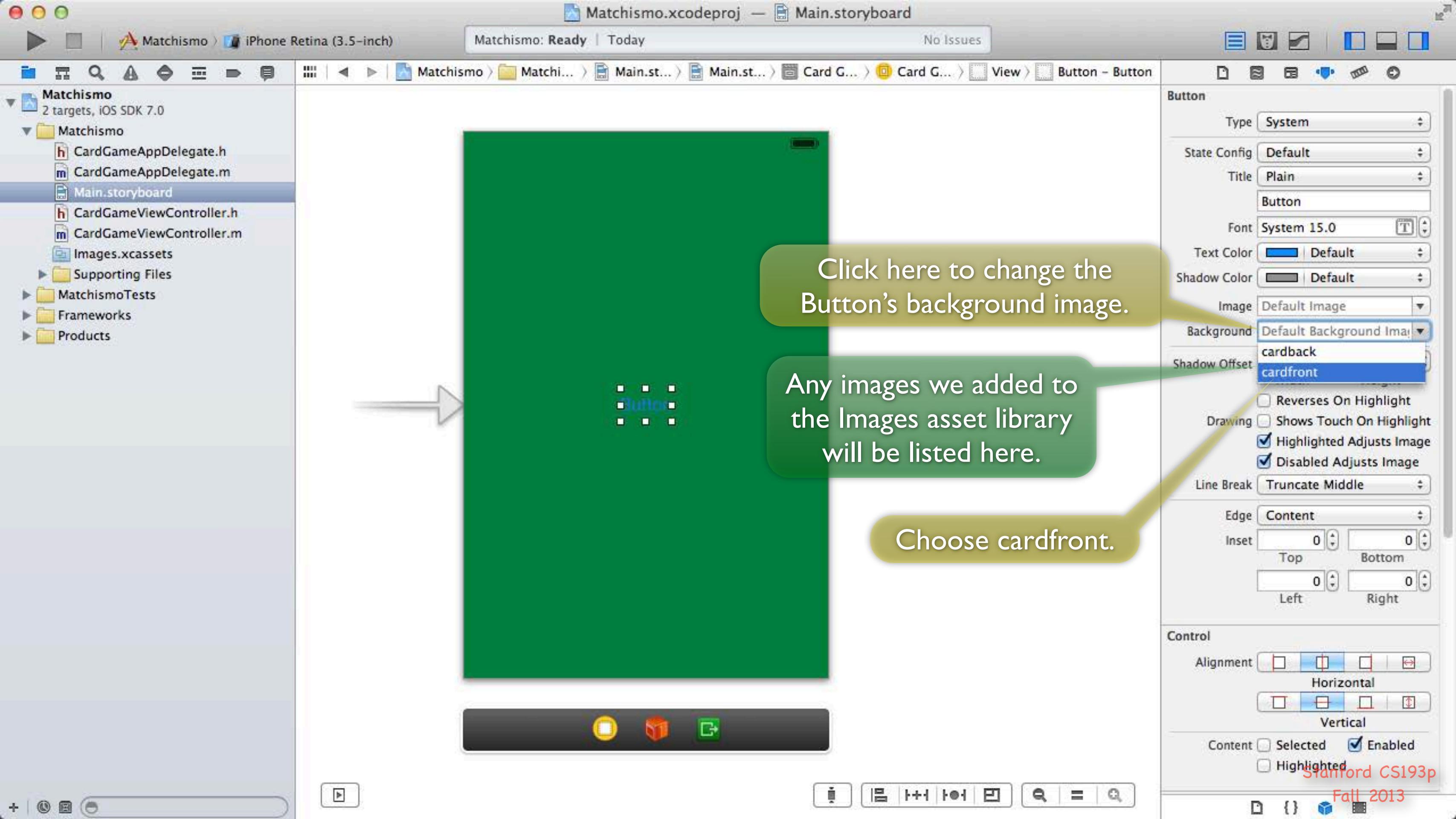
Fall 2013

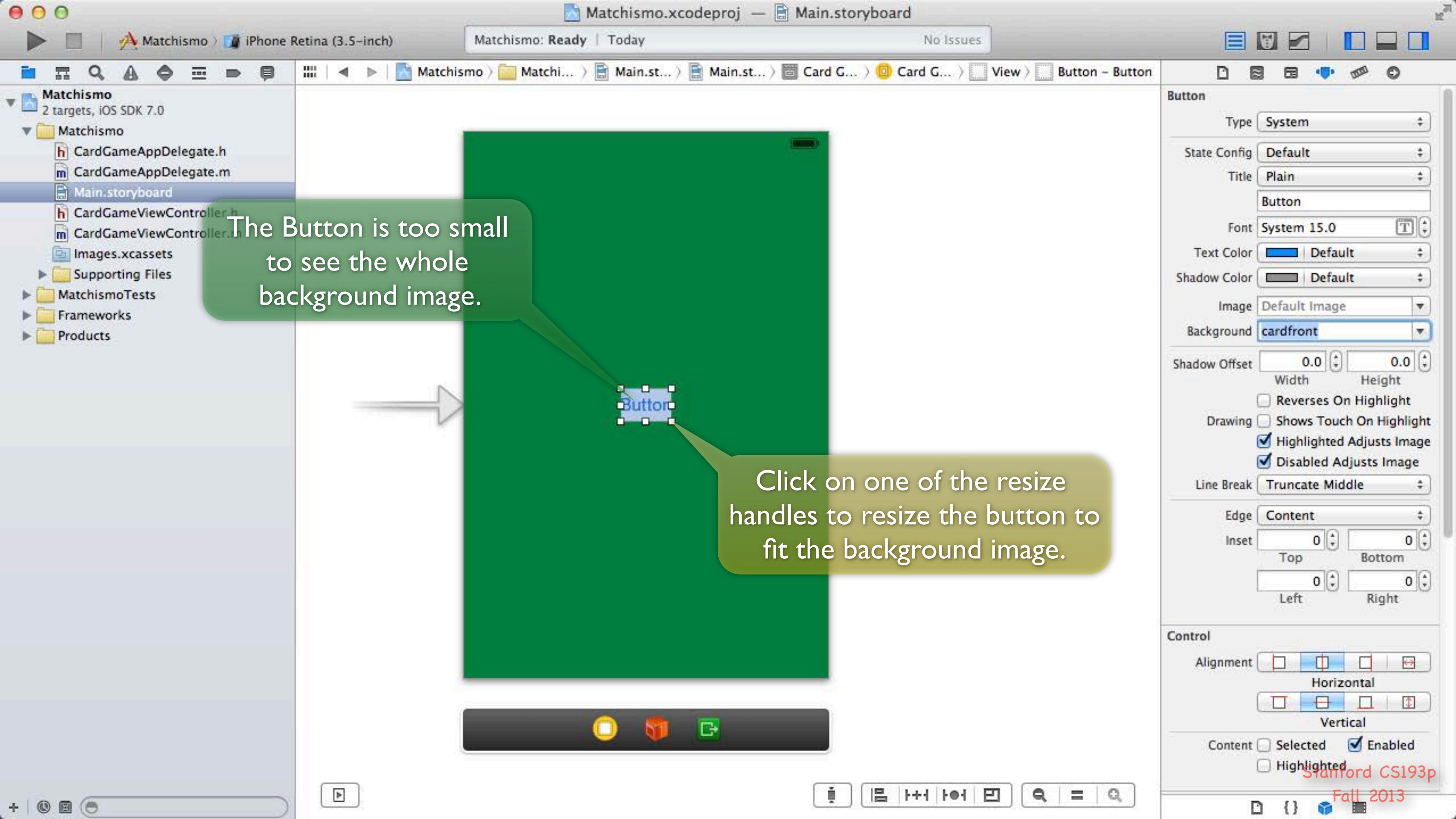


It's hard to see these two since they are blank,
but they are white "rounded rectangles."
You'll be able to see them better in the UI.

Stanford CS193p

Fall 2013





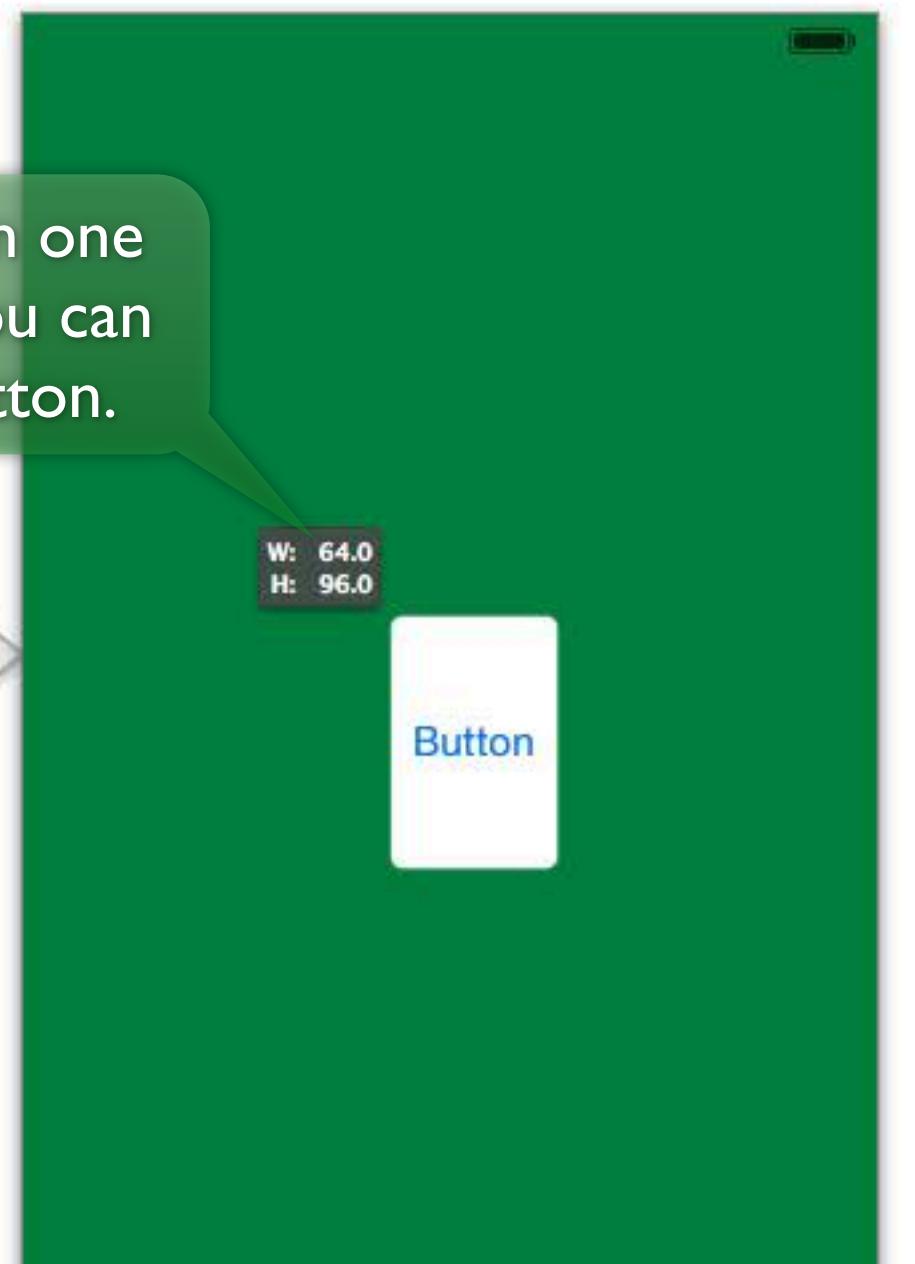


Matchismo
2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

If you press and hold on one of the resize handles, you can see the size of the button.



Button

Type System

State Config Default

Title Plain

Button

Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Drawing Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

0 0

Left Right

Control

Alignment

Horizontal

Vertical

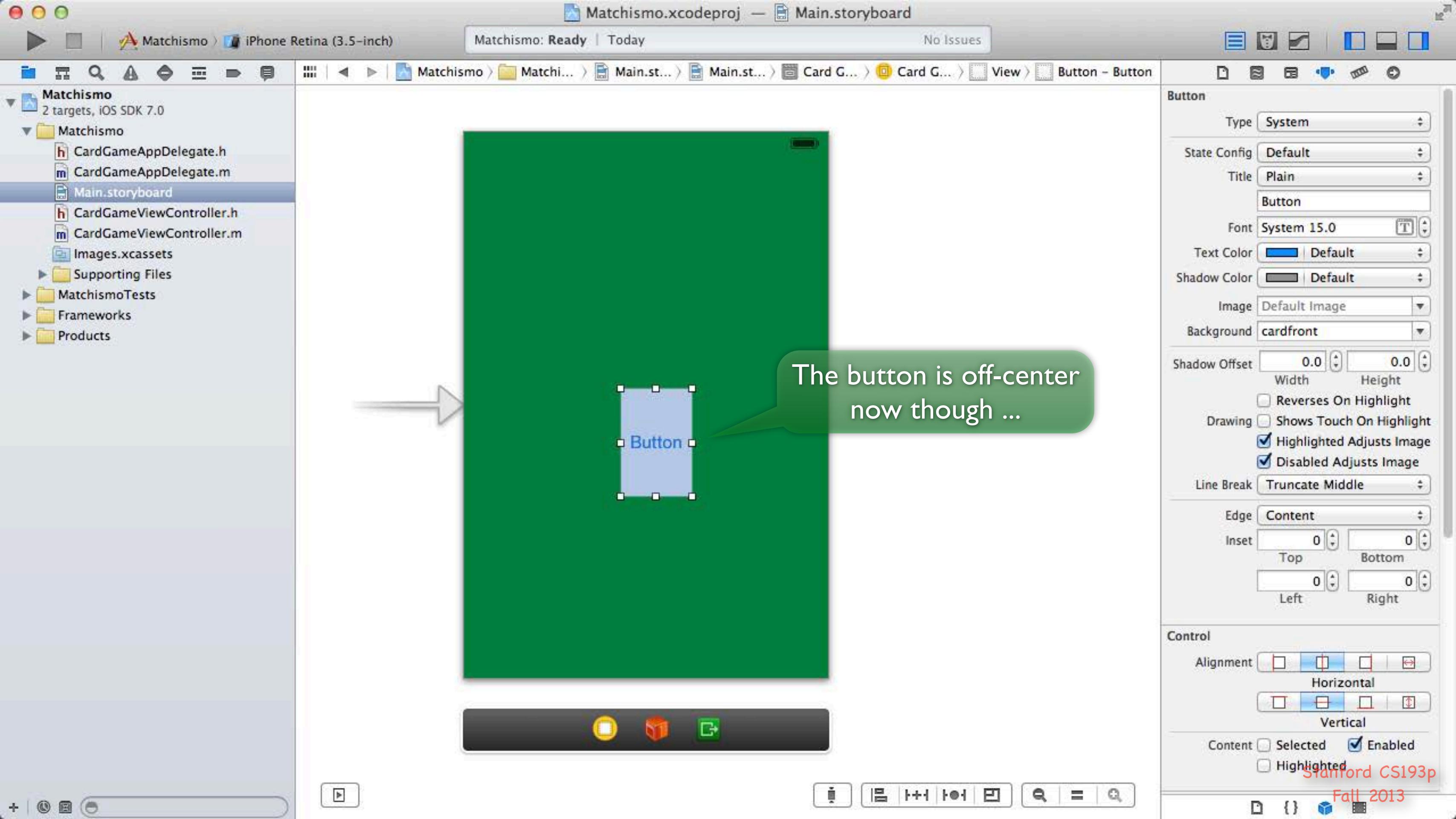
Content Selected Enabled

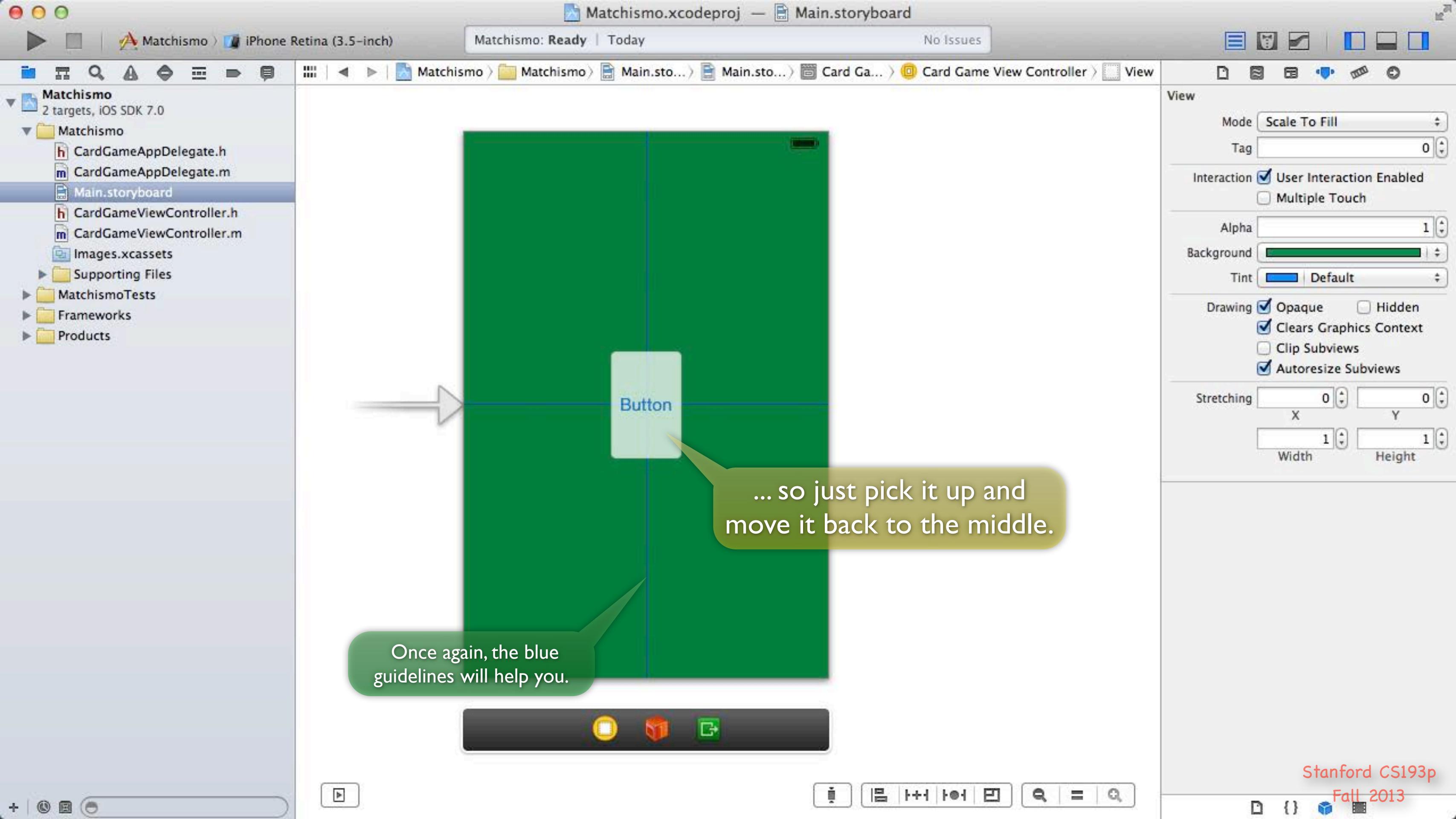
Highlighted

Stanford CS193p

Fall 2013







Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Main.storyboard

CardGameAppDelegate.h

CardGameAppDelegate.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Button - Button

Type System

State Config Default

Title Plain

Button

Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Drawing Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

Left Right

Control

Alignment Horizontal

Vertical

Content Selected Enabled

Highlighted

Double-click on the button to change the text on the card.

You can also edit the button's text in the Attributes Inspector here.

Stanford CS193p Fall 2013

The image shows the Xcode interface with the Main.storyboard file open. A button labeled "Button" is selected in the storyboard. Two callout bubbles provide instructions: one pointing to the button says "Double-click on the button to change the text on the card.", and another pointing to the Attributes Inspector says "You can also edit the button's text in the Attributes Inspector here.". The Attributes Inspector displays numerous properties for the selected button, including Type (System), Title (Plain), Font (System 15.0), and Background (cardfront). The storyboard preview shows a green card with the button. The bottom of the screen features the Xcode toolbar with icons for Save, Undo, Redo, and others.

Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo > iPhone Retina (3.5-inch)

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo: Ready | Today No Issues

Main.storyboard

CardGameViewController.h

CardGameViewController.m

View

Button - Button

Button

Type System

State Config Default

Title Plain

Button

Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Drawing Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

Left Right

Control

Alignment Horizontal

Vertical

Content Selected Enabled

Highlighted

Stanford CS193p

Fall 2013

Let's put the A♣ on the card.
The A is easy ...

A

Matchismo

Main.storyboard

CardGameViewController.h

CardGameViewController.m

View

Button - Button

Button

Type System

State Config Default

Title Plain

Button

Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Drawing Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

Left Right

Control

Alignment Horizontal

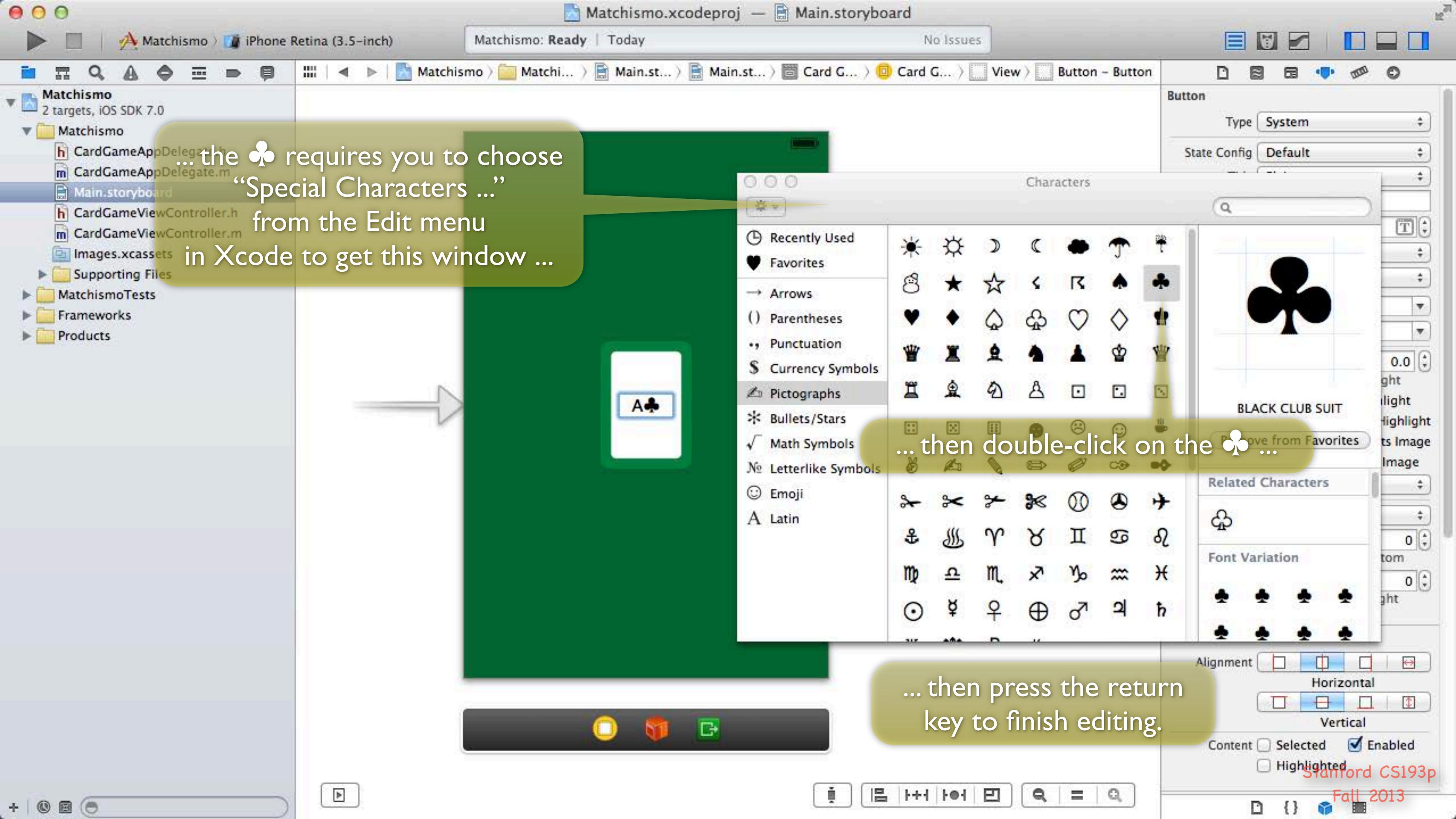
Vertical

Content Selected Enabled

Highlighted

Stanford CS193p

Fall 2013



Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard

A button can show different attributes in different states.
We're going to only set Default attributes.
The Default attributes are what will show in any button state
that does not have specific attributes set.

This is not quite what we want.
We want a little bigger font and for
the A to be black, not blue.

Button

Type System

State Conf ✓ Default

Title

- Highlighted
- Selected
- Disabled

Font System 15.0

Text Color Default

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset Top 0 Bottom 0

Left 0 Right 0

Control

Alignment Horizontal

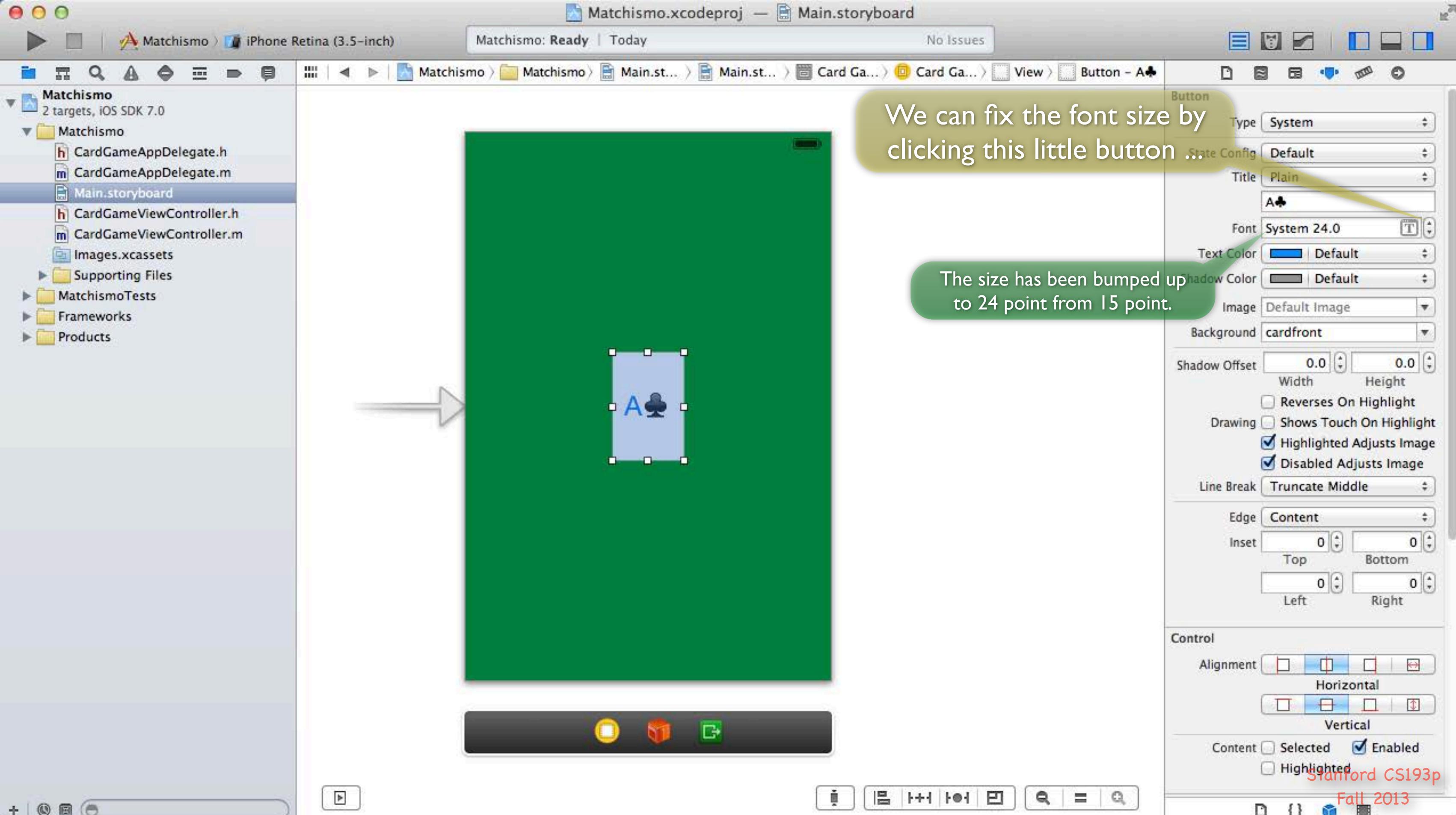
Vertical

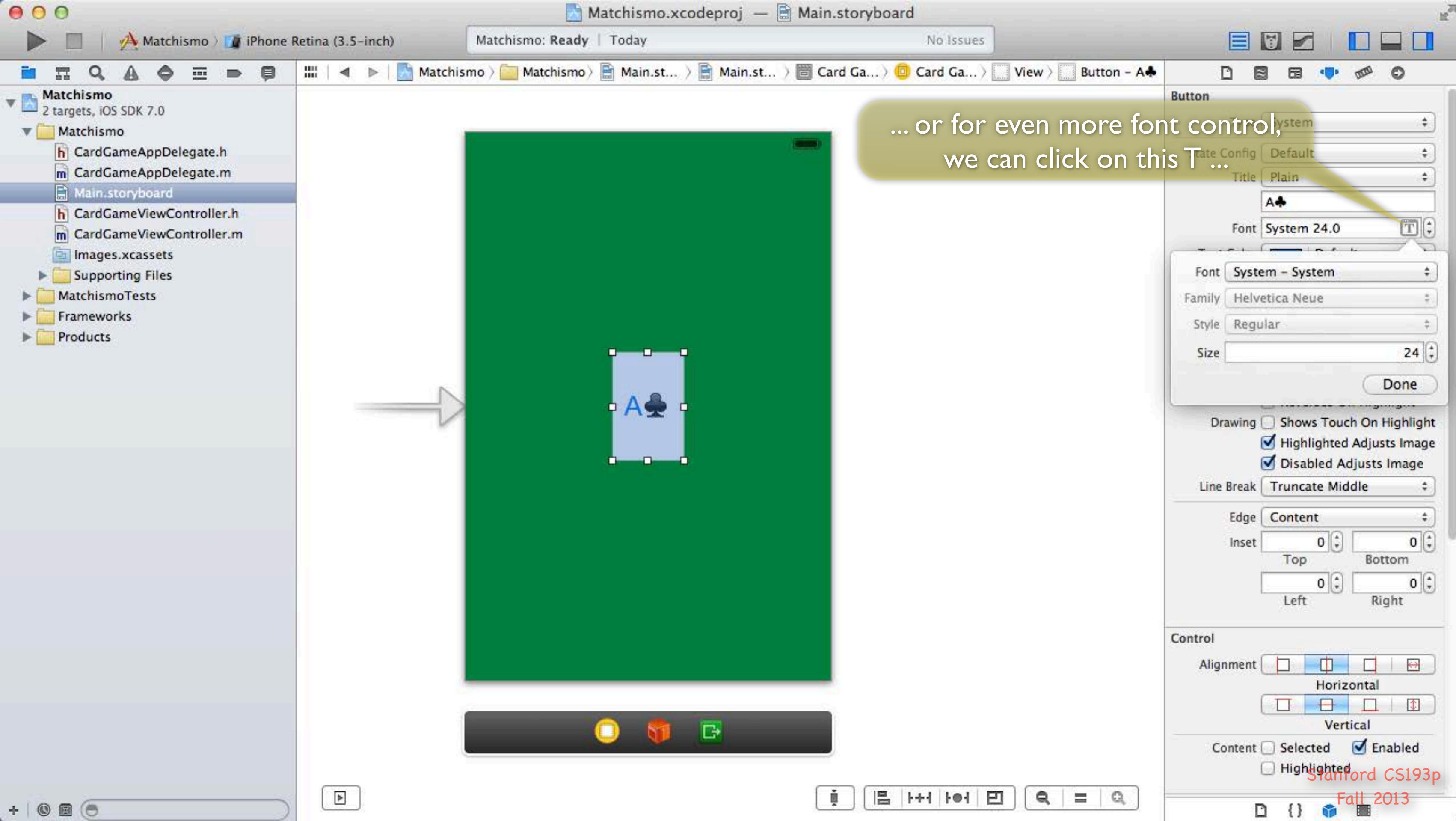
Content Selected Enabled

Highlighted

Stanford CS193p

Fall 2013





Matchismo.xcodeproj — Main.storyboard

Matchismo: Ready | Today No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

View

Button - A♣

Button

Type System

State Config Default

Title Plain

A♣

Custom

System

Font System

Family System

Style System

Size System

Text Styles

Body

Caption 1

Caption 2

Footnote

Headline

Subhead

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top 0 Bottom 0

Left 0 Right 0

Control

Alignment Horizontal

Vertical

Content Selected Enabled

Highlighted

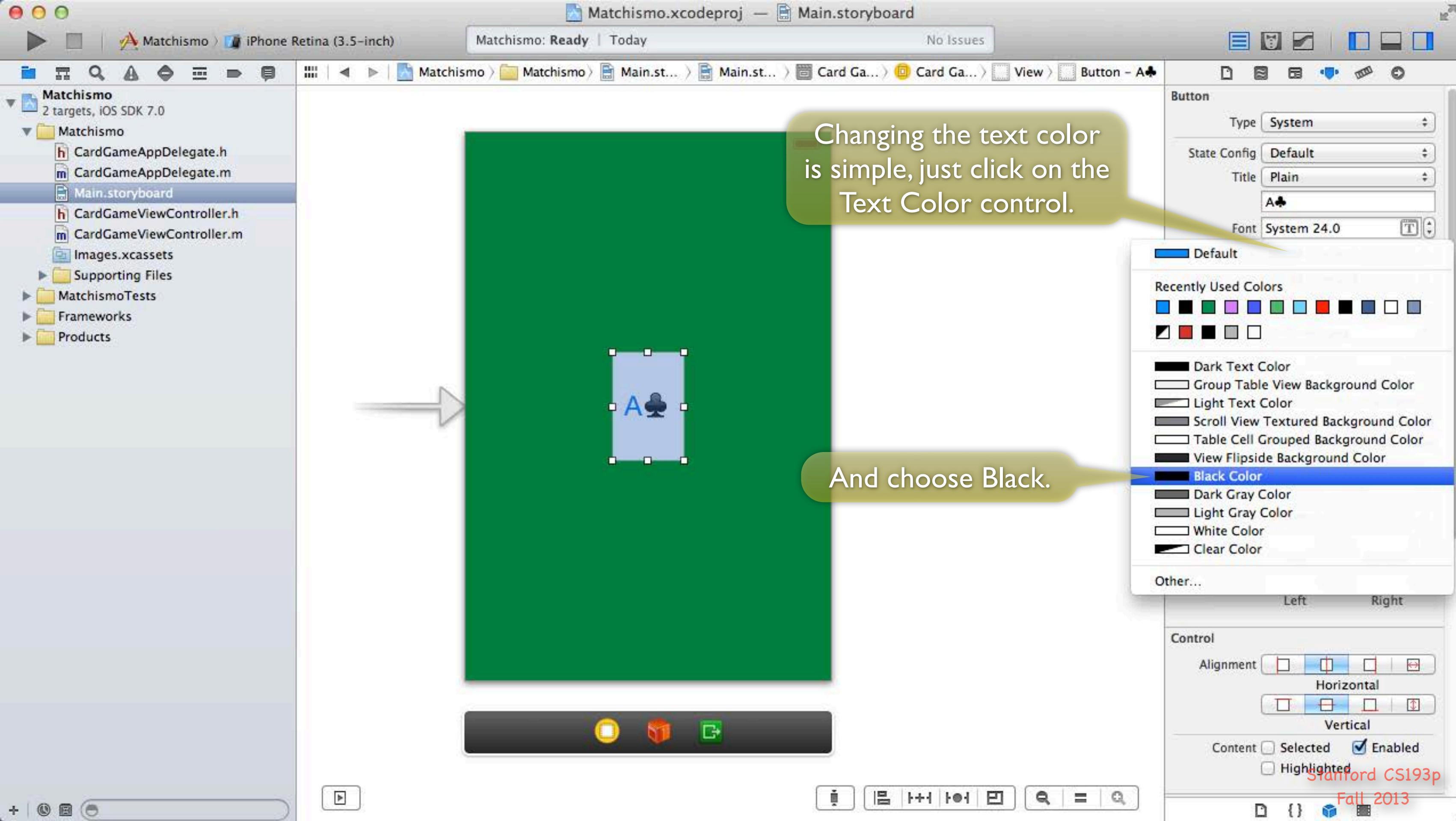
Stanford CS193p

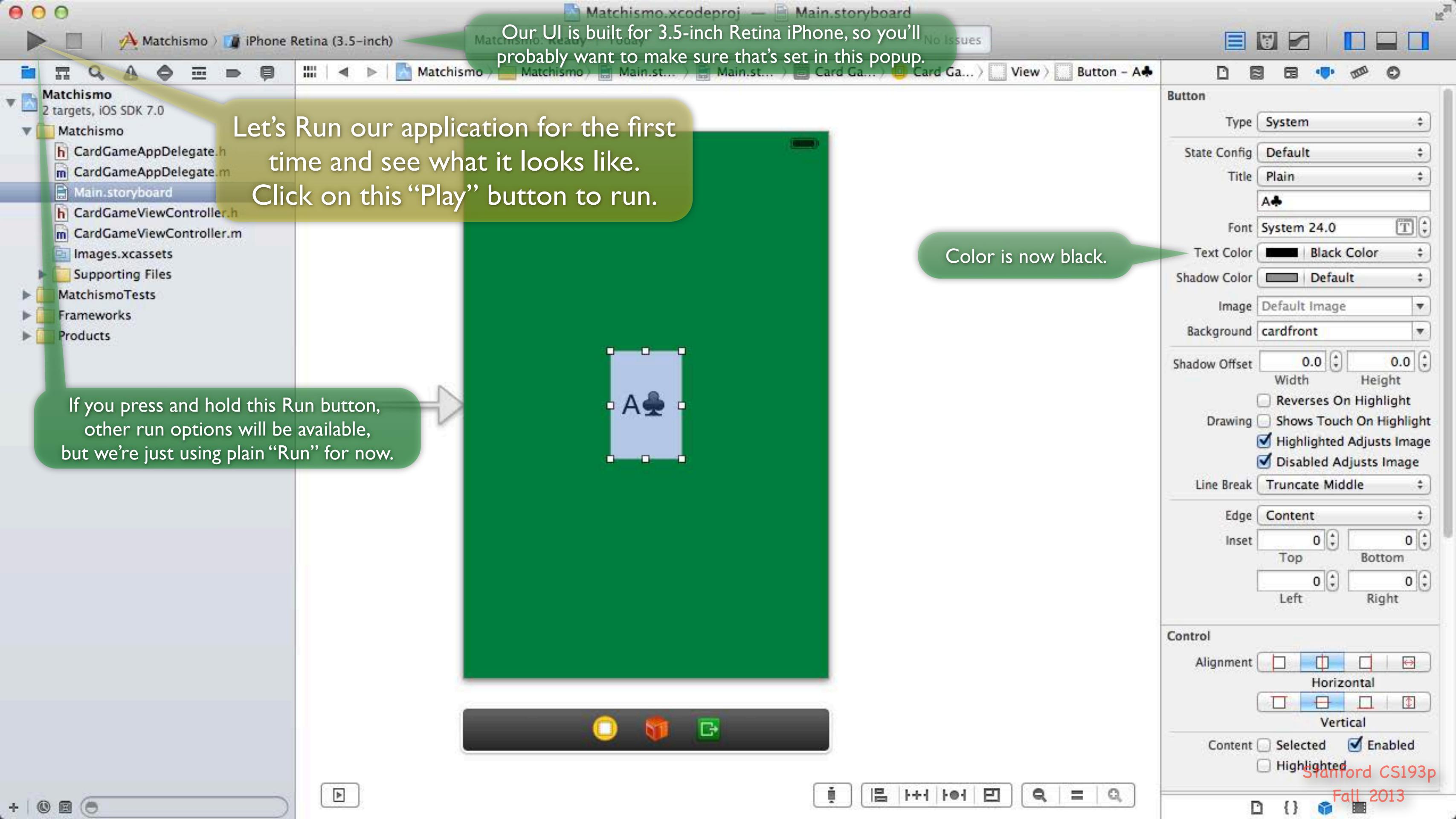
Fall 2013

... and then change the font, it's family or style, and size.

We'll leave this set to the System font for now.

Typography is crucially important to iOS 7 user-interface design. We'll talk about that (especially these Text Styles) later in the course.





Matchismo.xcodeproj — Main.storyboard

Building Matchismo: Matchismo | Linking No Issues

Matchismo iPhone Retina (3.5-inch)

Main.storyboard

Card Game View Controller

As the application is built, you'll see status here.

Button

Type: System

State Config: Default

Title: Plain

Image: A♣

Font: System 24.0

Text Color: Black Color

Shadow Color: Default

Background: cardfront

Shadow Offset: 0.0 0.0

Width: 0.0 Height: 0.0

Reverses On Highlight:

Drawing: Shows Touch On Highlight:

Highlighted Adjusts Image:

Disabled Adjusts Image:

Line Break: Truncate Middle

Edge: Content

Inset: 0 0

Top: 0 Bottom: 0

Left: 0 Right: 0

Control

Alignment: Horizontal

Vertical

Content: Selected: Enabled:

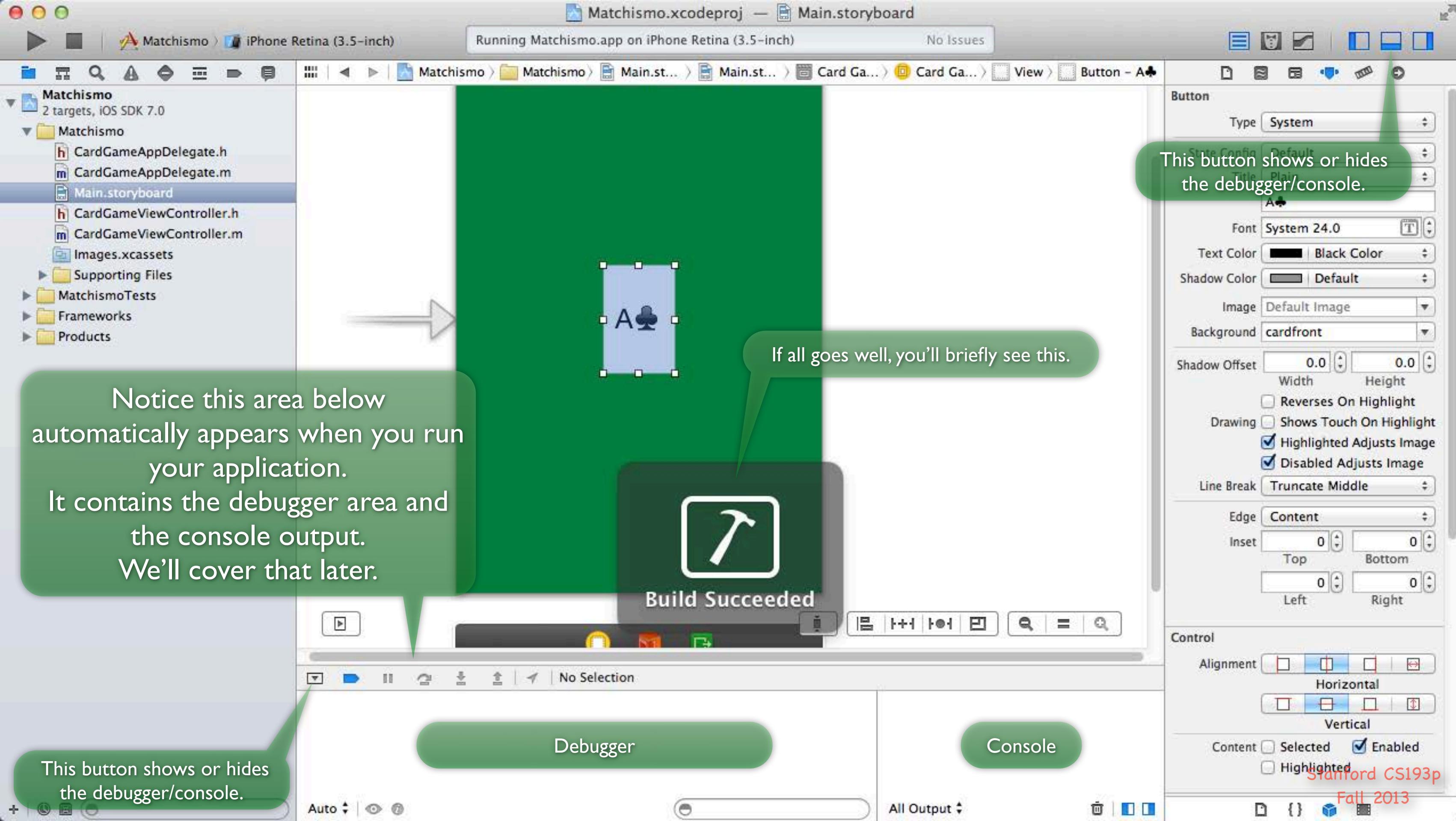
Highlighted:

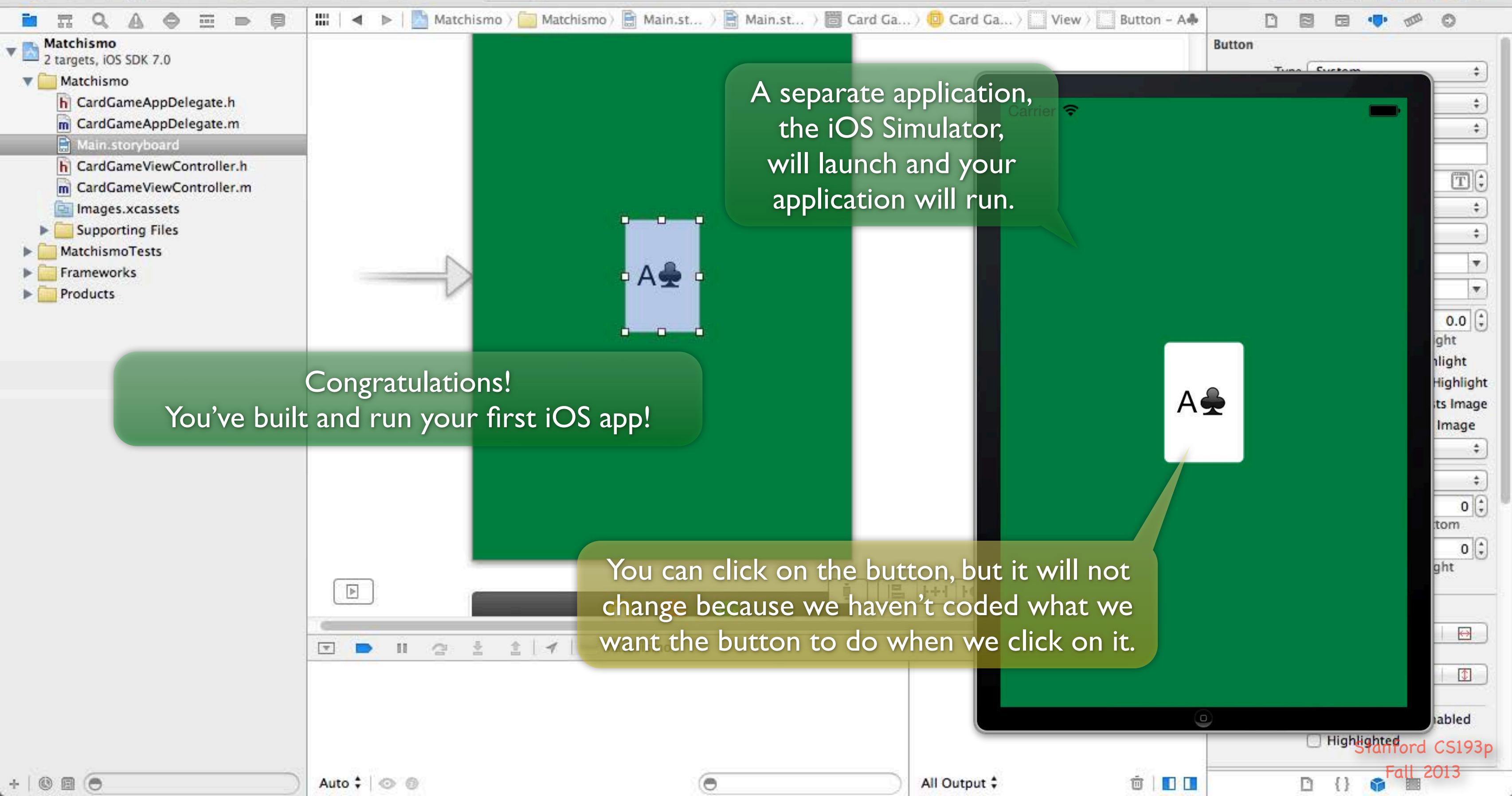
Stanford CS193p Fall 2013

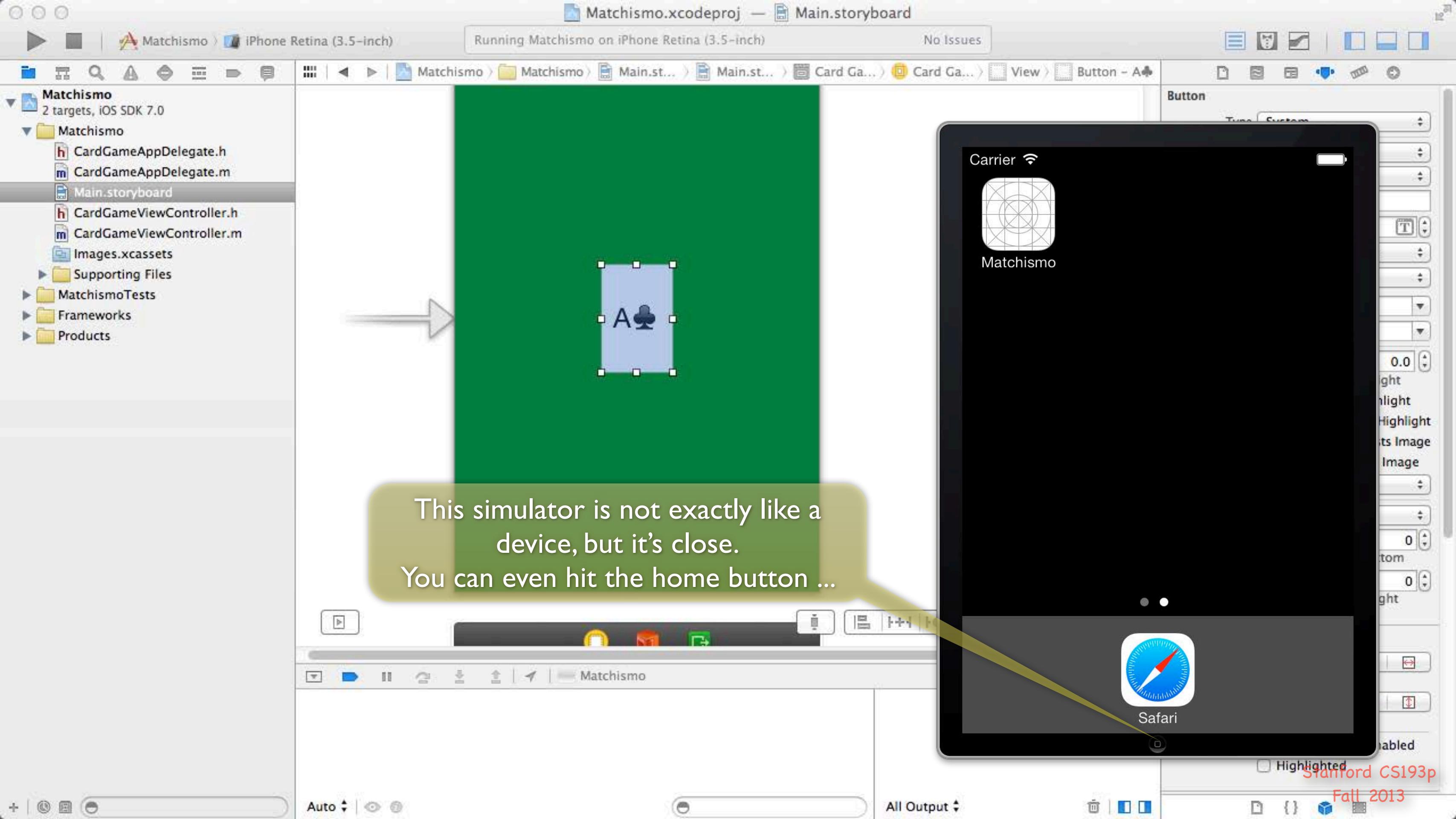
File Editor View Library Inspector

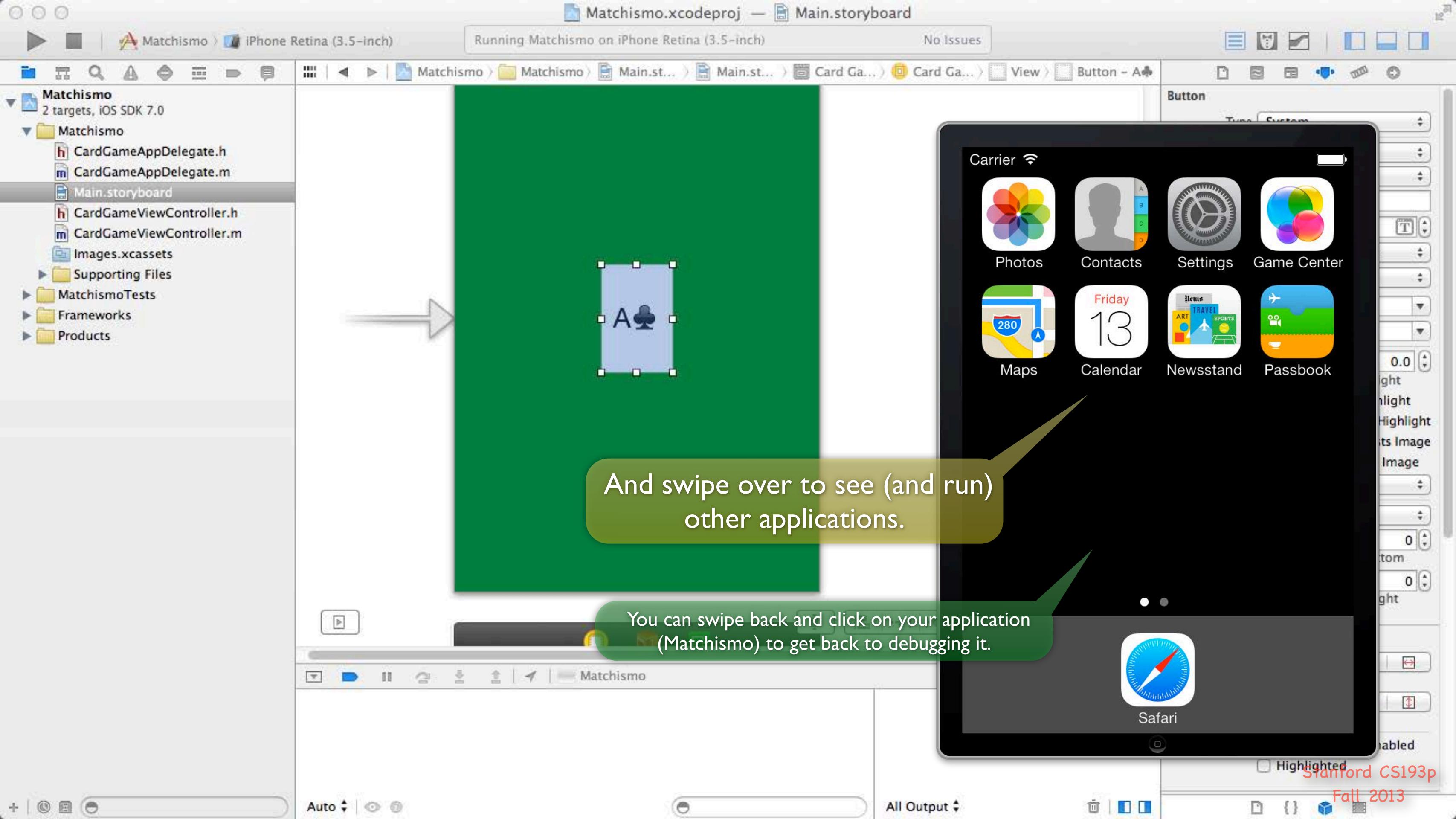
Run Stop

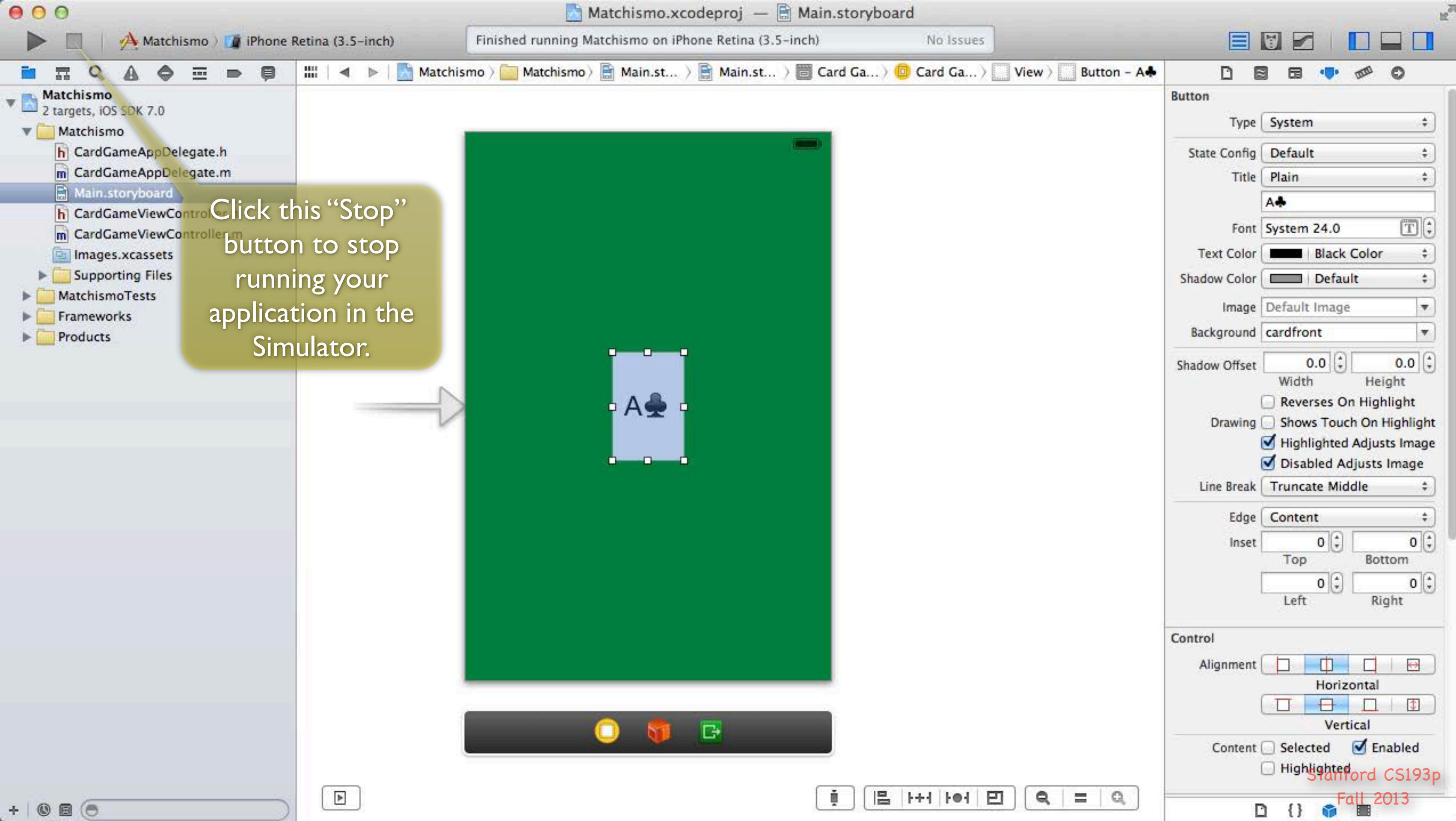
Search

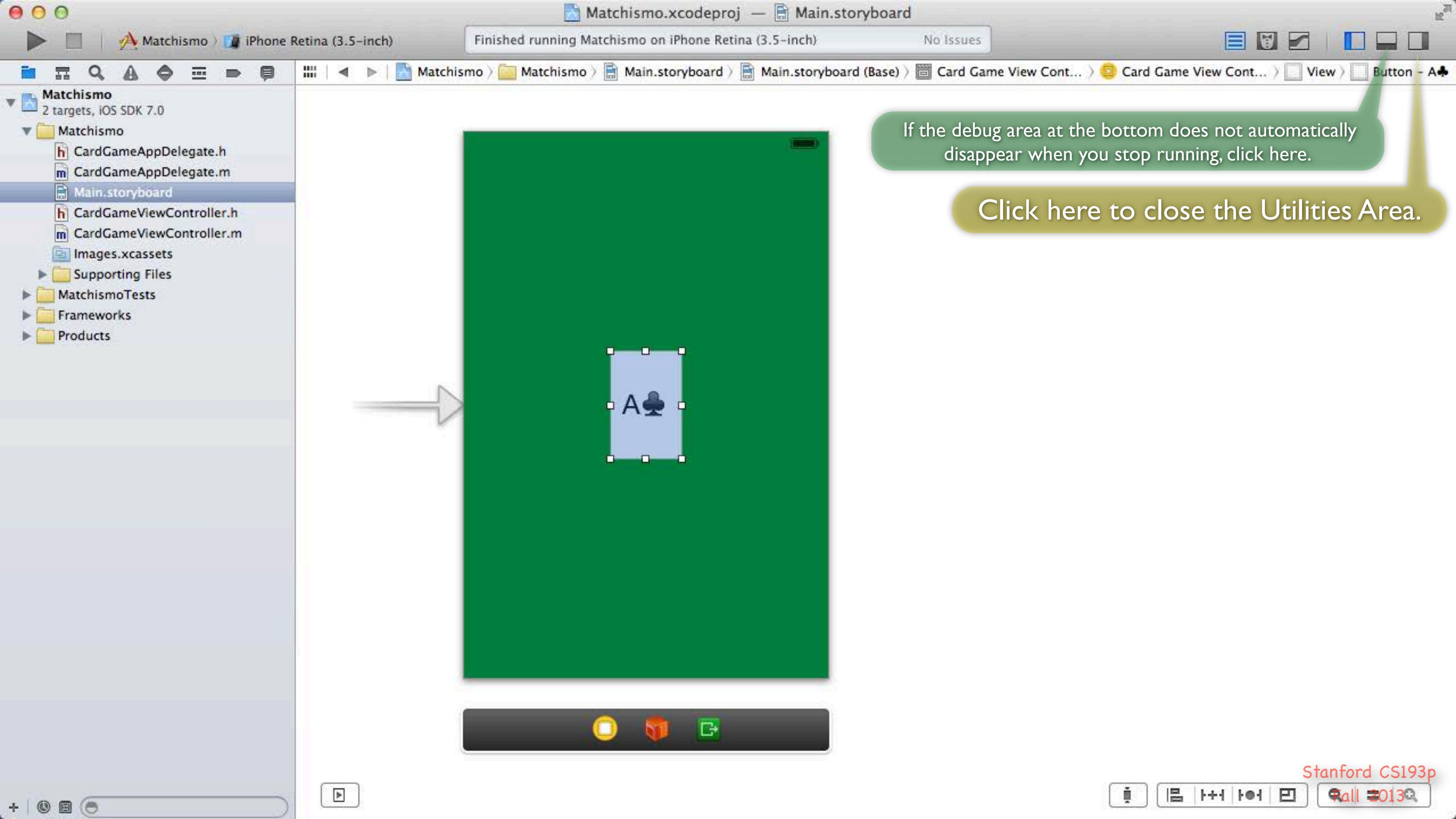


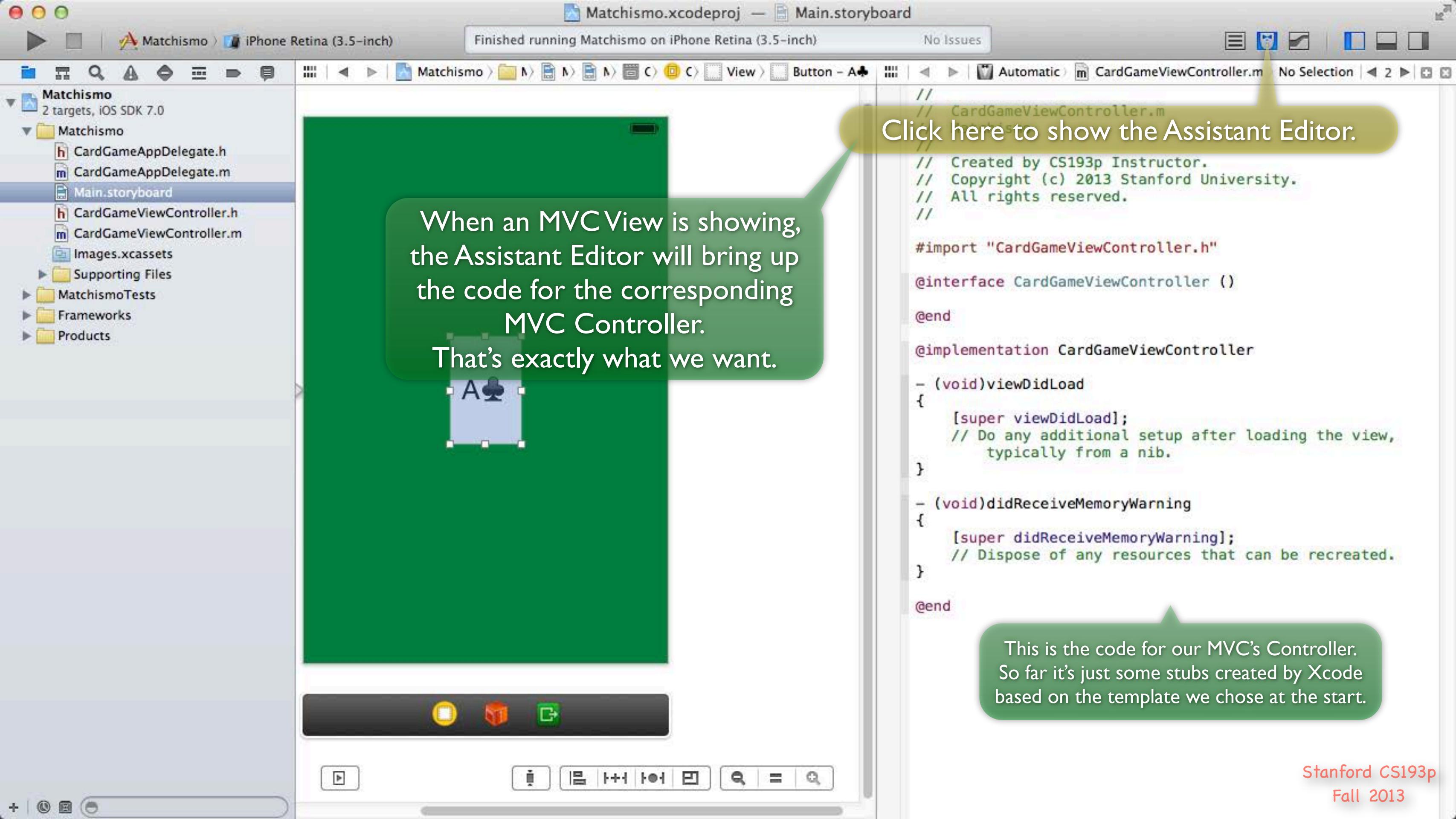












When an MVC View is showing,
the Assistant Editor will bring up
the code for the corresponding
MVC Controller.
That's exactly what we want.

Click here to show the Assistant Editor.

This is the code for our MVC's Controller.
So far it's just some stubs created by Xcode
based on the template we chose at the start.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
  
@end  
  
@implementation CardGameViewController  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.}  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Dispose of any resources that can be recreated.}  
  
@end
```

You can adjust the space between the two panes of the editor by just dragging the space between them.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Main.storyboard CardGameViewController.m Selection

CardGameViewController.h

```
//  
// CardGameViewController.h  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
  
@end  
  
@implementation CardGameViewController  
  
- (void)viewDidLoad  
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
}  
  
- (void)didReceiveMemoryWarning  
{  
    [super didReceiveMemoryWarning];  
    // Dispose of any resources that can be recreated.  
}  
  
@end
```

If you want to edit the header file of the Controller (to make something public for example), you can switch it here.

A Ace of Spades card is displayed on the storyboard canvas.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files

MatchismoTests

Frameworks

Products

iPhone Retina (3.5-inch)

Automatic CardGameViewController.h No Selection

```
//  
// CardGameViewController.h  
// Matchismo  
//  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import <UIKit/UIKit.h>  
  
@interface CardGameViewController : UIViewController  
  
@end
```

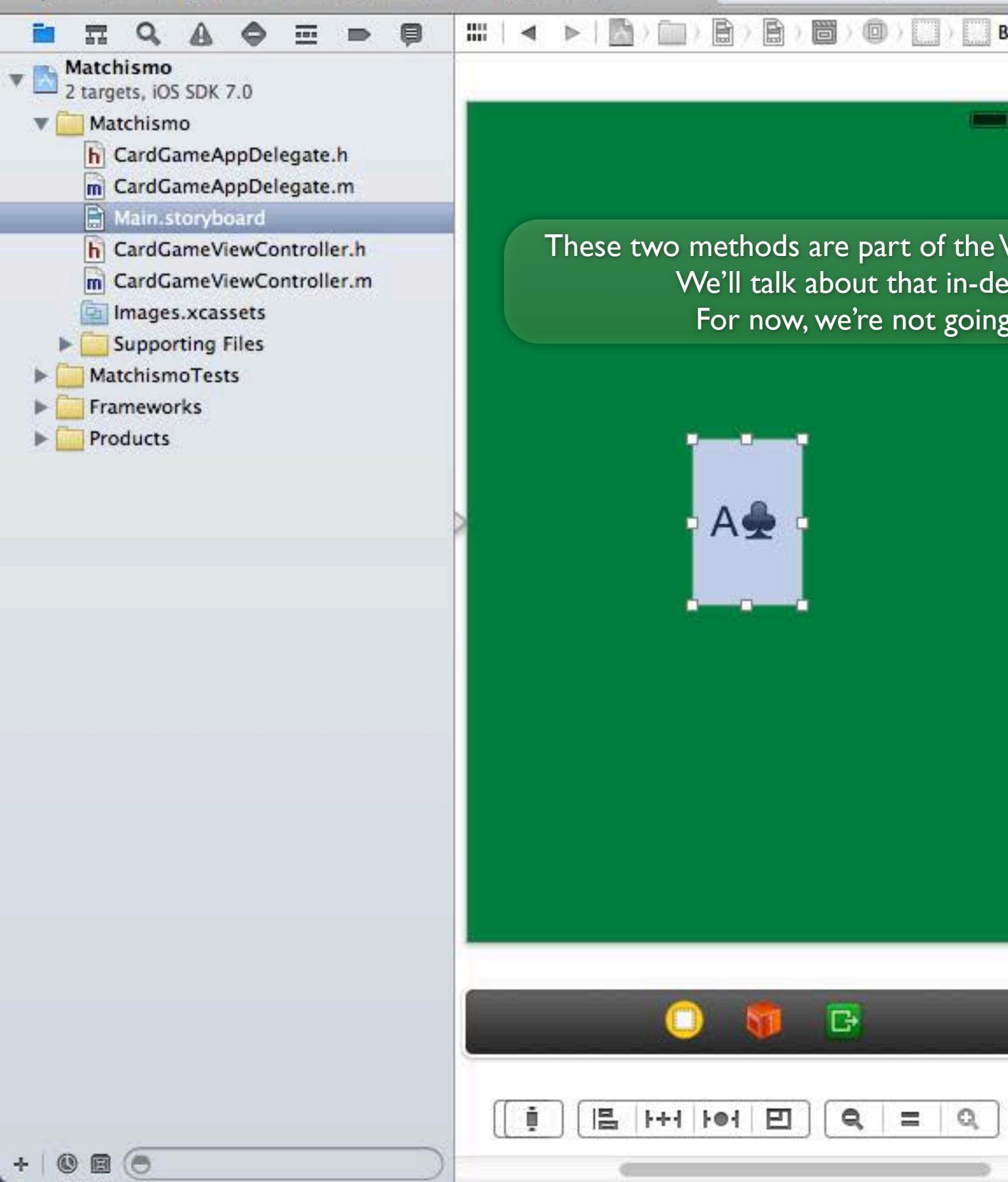
All Controllers inherit from UIViewController.

Switch back to the Controller's implementation.

A speech bubble points to the line "@interface CardGameViewController : UIViewController".

The storyboard preview shows a single card labeled "A ♠".

Bottom navigation bar icons: file, undo, redo, search, etc.



These two methods are part of the View Controller Lifecycle.

We'll talk about that in-depth next week.

For now, we're not going to use them.

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

```
#import "CardGameViewController.h"
```

```
@interface CardGameViewController : UIViewController
```

```
@end
```

```
@implementation CardGameViewController
```

```
- (void)viewDidLoad
```

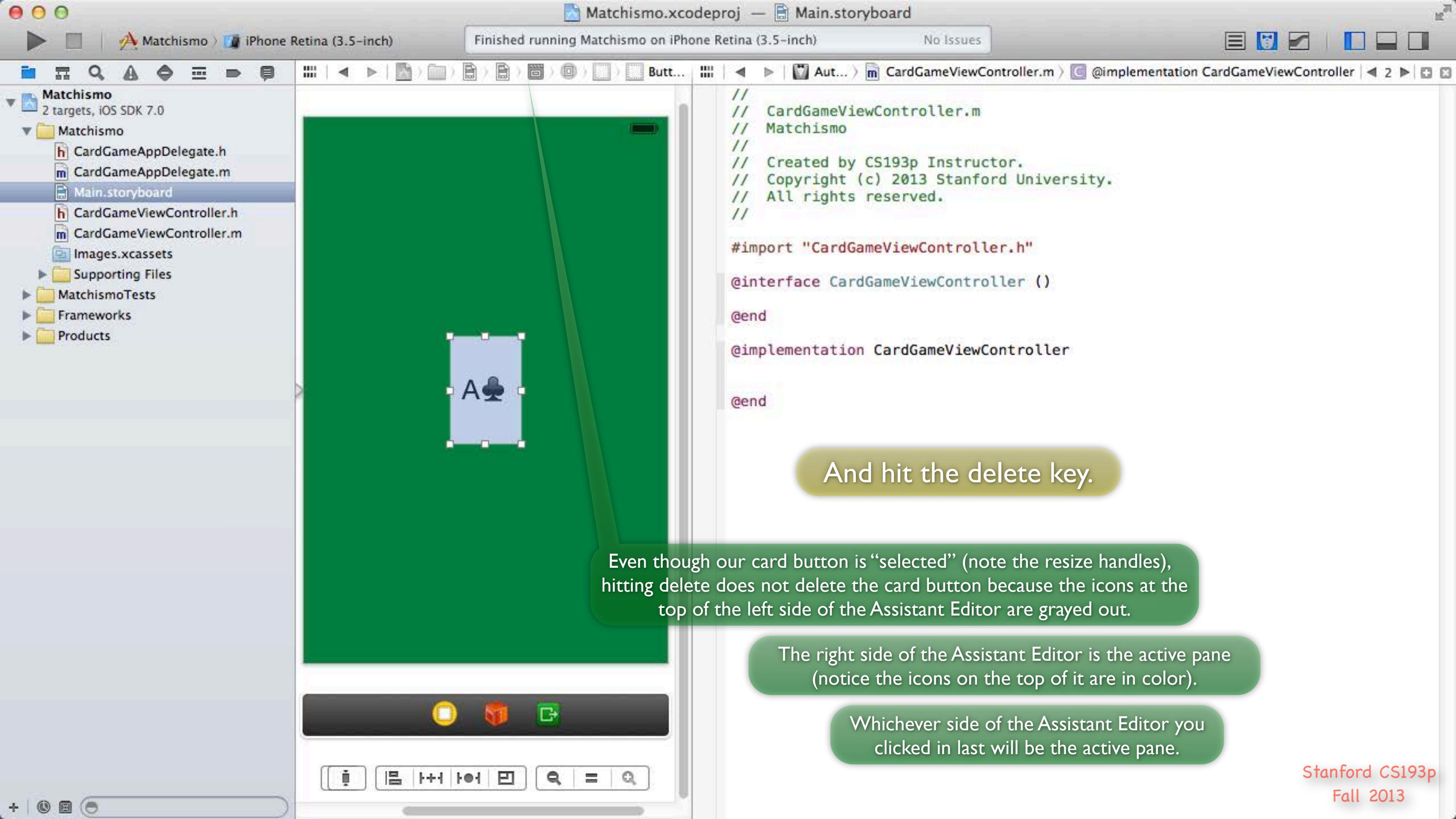
```
{  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
}
```

```
- (void)didReceiveMemoryWarning
```

```
{  
    [super didReceiveMemoryWarning];  
    // Dispose of any resources that can be recreated.  
}
```

```
@end
```

Select all of the code inside the **@implementation** block ...

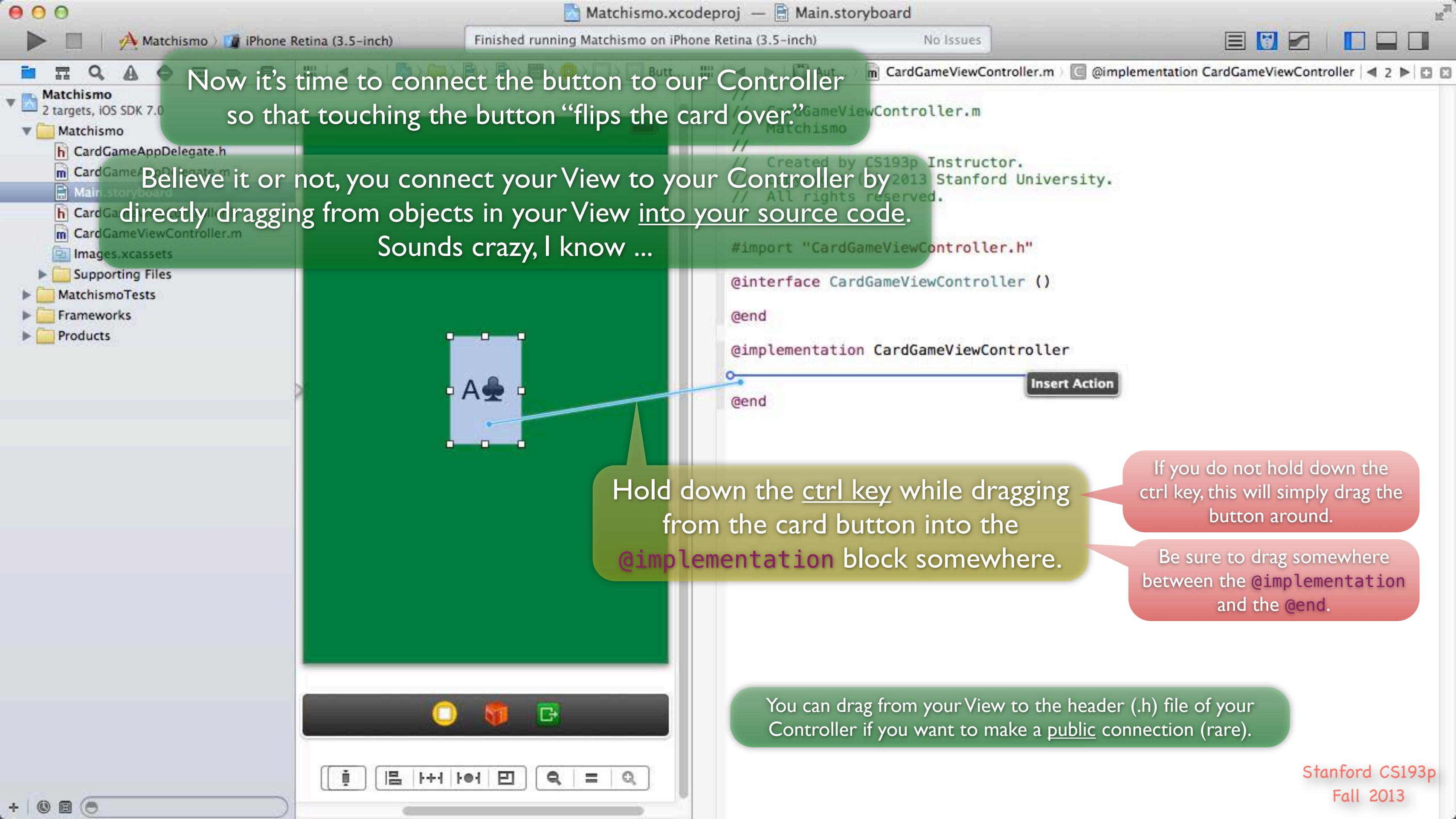


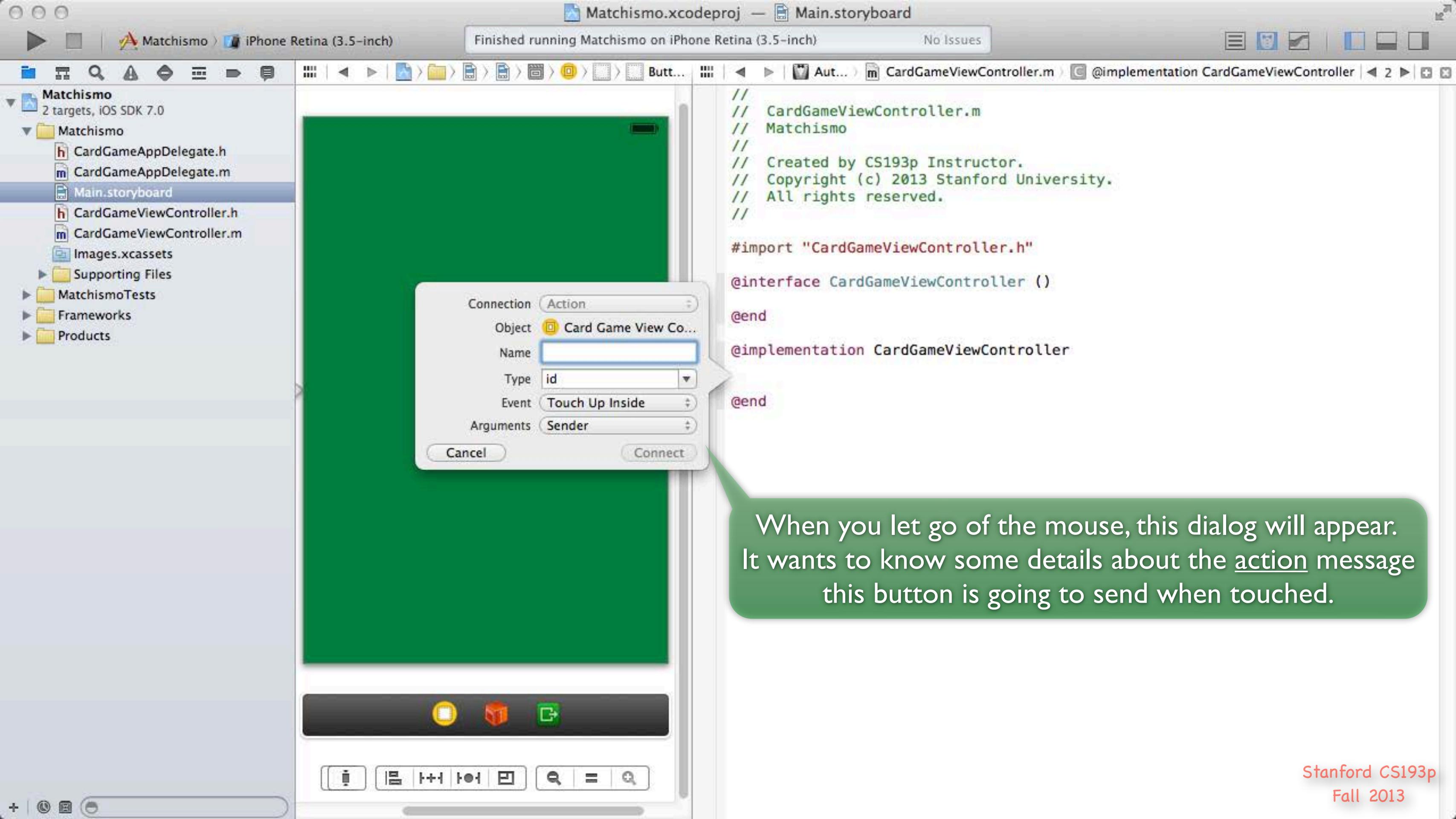
And hit the delete key.

Even though our card button is “selected” (note the resize handles), hitting delete does not delete the card button because the icons at the top of the left side of the Assistant Editor are grayed out.

The right side of the Assistant Editor is the active pane (notice the icons on the top of it are in color).

Whichever side of the Assistant Editor you clicked in last will be the active pane.







Main.storyboard

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
@end
```

Card Game View Co... touchCardButton id Touch Up Inside Sender

Let's name this action method touchCardButton.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

Main.storyboard

Action

Object: Card Game View Co...

Name: touchCardBody

Type: id

Event: UIButton

Arguments: Sender

Cancel Connect

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
@end
```

We know that the sender of this action is a **UIButton**. If this action could be sent from objects of another class (e.g. a **UISlider**), we could leave the type “**id**” (which means an object of any (unknown) class).

Specifying that we know the class of the sender makes it easier for the compiler to check that the code in our action method is not faulty.

We'll talk more about this type **id** next week.

Stanford CS193p Fall 2013



Matchismo
2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files

MatchismoTests

Frameworks

Products

Main.storyboard

Action

Object: Card Game View Co...

Name: touchCardButton

Type: UIButton

Event: Touch Up Inside

Arguments: Sender

Cancel Connect

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
@end  
  
@implementation CardGameViewController  
@end
```

Make sure it says UIButton here.

Stanford CS193p
Fall 2013

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

```
#import "CardGameViewController.h"
```

```
@interface CardGameViewController ()
```

@end

```
@implementation CardGameViewController
```

@end

The screenshot shows the Xcode Interface Builder Attributes Inspector. The top bar has tabs for "Connection" and "Action". The "Action" tab is selected. Under "Object", there is a yellow square icon followed by "Card Game View Co...". Below it, "Name" is set to "touchCardButton" and "Type" is set to "UIButton". A dropdown arrow indicates more options. The "Event" dropdown is open, showing "None" as the current selection. The "Argument" dropdown is also open, showing "Sender" as the current selection. At the bottom left are "Cancel" and "OK" buttons.

Action methods are allowed to have either no arguments,
one argument (the object sending the message),
or even two arguments

(the sender and the touch event provoking the action). In this case, though, what we want is just the sending button so that we can change it when it is touched.

The screenshot shows the Xcode interface with the project "Matchismo" selected. The storyboard file "Main.storyboard" is open, showing a green view controller. A context menu is displayed over the view, with the "Touch Up Inside" option highlighted.

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
@end
```

Normally buttons send their action when a touch that started by landing on them goes “up” while still inside the button’s boundaries. That’s what we want.

Stanford CS193p
Fall 2013

Matchismo
2 targets, iOS SDK 7.0

Matchismo

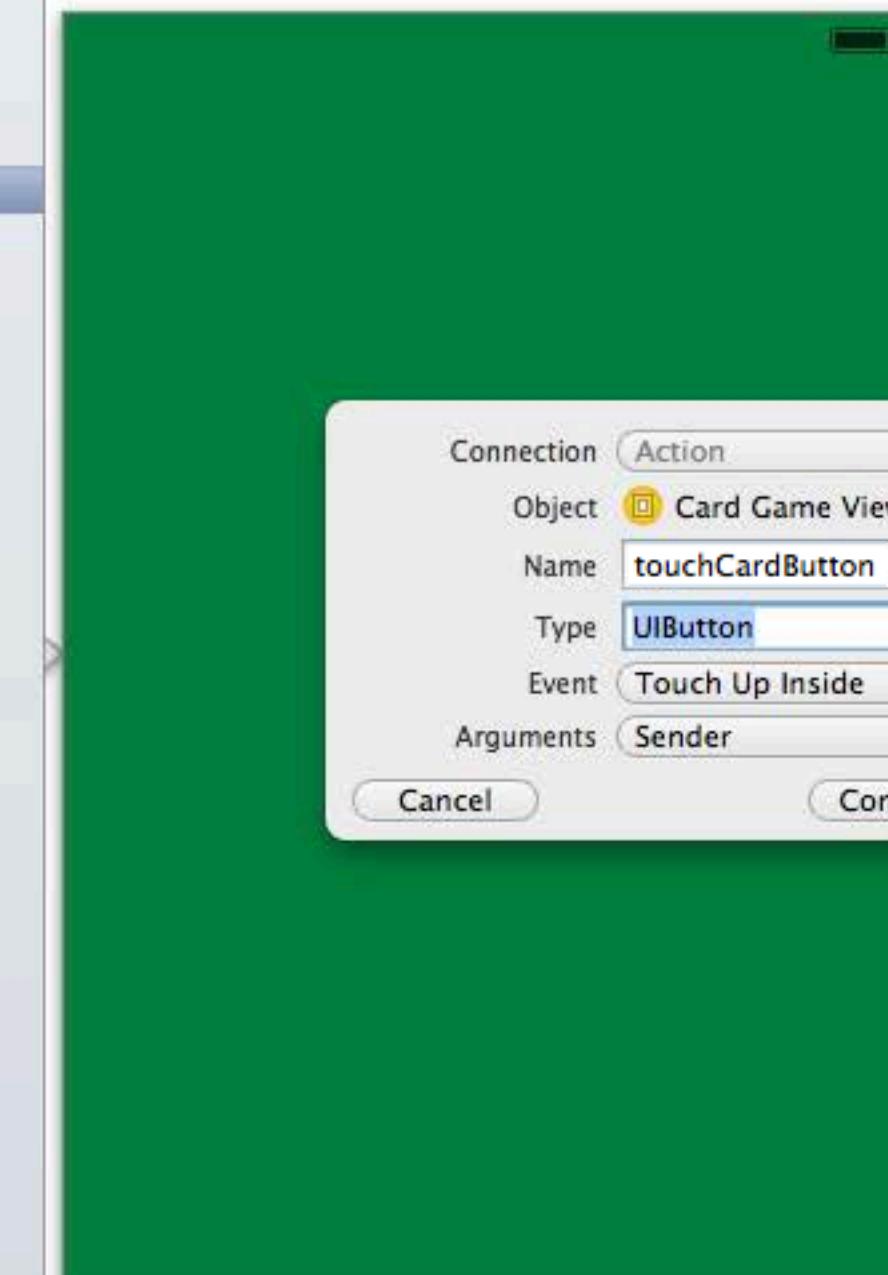
- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products



```
//  
//  CardGameViewController.m  
//  Matchismo  
//  
//  Created by CS193p Instructor.  
//  Copyright (c) 2013 Stanford University.  
//  All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
@end
```

Once you've set up the action as shown, click Connect to create the action method.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

Main.storyboard

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

#import "CardGameViewController.h"

@interface CardGameViewController : UIViewController

@end

@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
    NSLog(@"Button tapped!");
}

@end
```

The only argument to this method is the object that is sending the message to us ("us" is the Controller). In the previous slide, we made it clear that it is a button, so that's why the type of this argument is `UIButton` (instead of `id`).

The name of this method is actually `touchCardButton:`, not `touchCardButton`.

This method's return type is actually `void`, but Xcode uses the `typedef IBAction` instead just so that Xcode can keep track that this is not just a random method that returns `void`, but rather, it's an action method.

Apart from that, `IBAction` is exactly the same thing as `void`.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

Main.storyboard

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
}  
  
@end
```

Highlighted!

Mouse over (do not click) this little icon. The object in the View that sends this message will highlight.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Our implementation of this method is quite simple.
We're just going to change the text on the button to be blank
and change the background image to be our card back
(the Stanford logo), thus “flipping the card over to its back.”

```
// CardGameViewController.m
// Copyright (c) 2013 Stanford University.

#import "CardGameViewController.h"

@interface CardGameViewController : UIViewController
@end

@implementation CardGameViewController
- (IBAction)touchCardButton:(UIButton *)sender
{
    UIImage *cardImage =
}
@end
```

Let's start by declaring a *local variable* called `cardImage` to hold the cardback image (the Stanford logo we imported). The local variable is a pointer to an instance of the class `UIImage` which represents a JPEG, PNG, TIFF or other image.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

Uh-oh, a warning!

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];  
}  
@end
```

Remember that @"" notation just creates an NSString object.

UIImage has a *class method* called imageNamed: which creates an instance of UIImage given the name of an image in the image assets library. We just specify cardback (what we called the Stanford logo in the assets library).

Stanford CS193p
Fall 2013

Xcode is constantly parsing your code in the background.
Thus, warnings and errors will appear in this gutter without
your having to explicitly build your project.

```
#import "CardGameViewController.h"

@interface CardGameViewController : UIViewController

@end

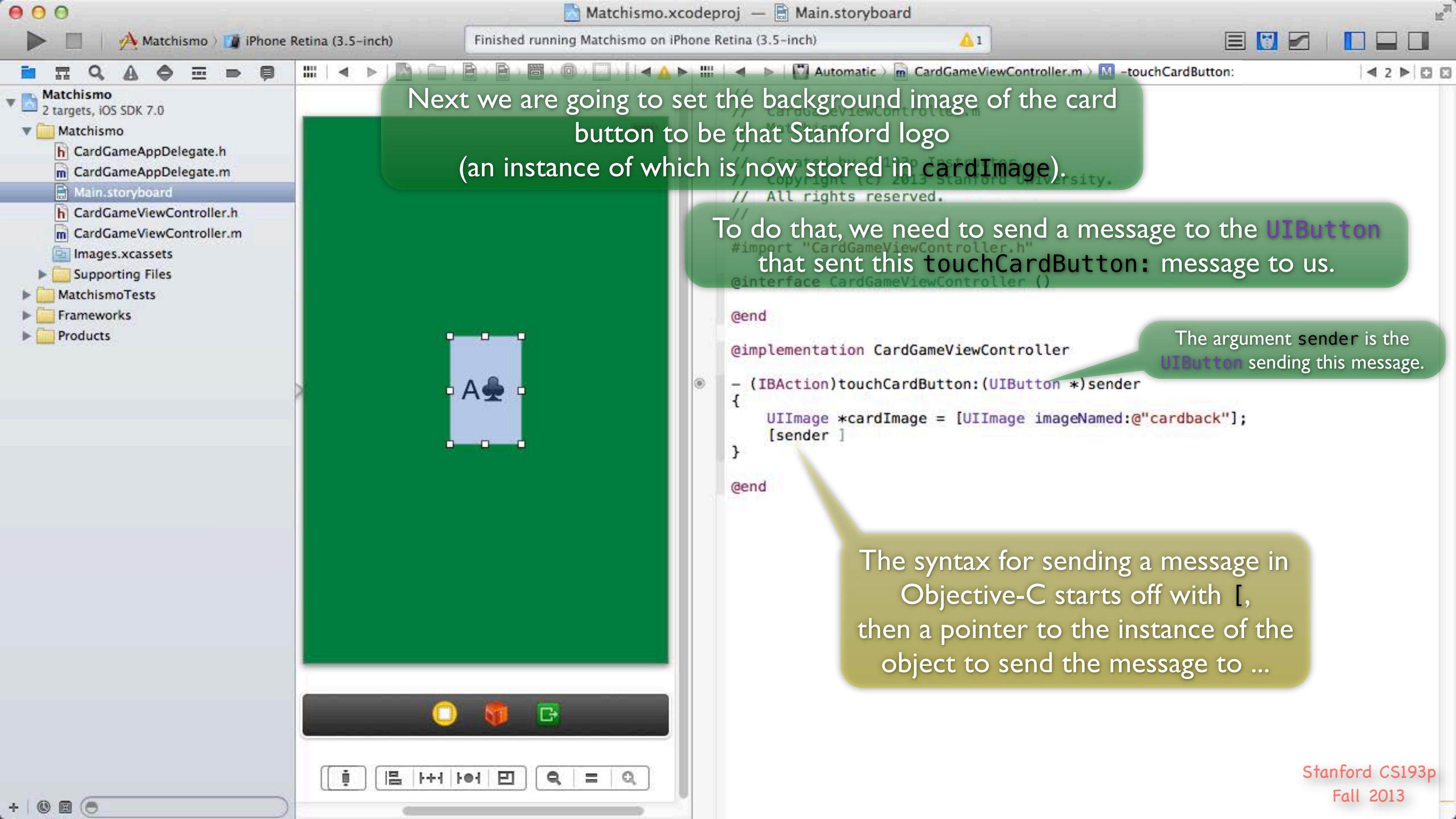
@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];
}

@end
```

To get more details about a warning or error, click on the triangle.

This warning is correct (we are not using the local variable cardImage yet), but nothing to worry about since we'll be using it in a moment.



Next we are going to set the background image of the card button to be that Stanford logo (an instance of which is now stored in `cardImage`).

To do that, we need to send a message to the `UIButton` that sent this `touchCardButton:` message to us.

The argument `sender` is the `UIButton` sending this message.

The syntax for sending a message in Objective-C starts off with `[`, then a pointer to the instance of the object to send the message to ...

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files

MatchismoTests

Frameworks

Products

A card labeled "A" with a spade symbol is displayed on the storyboard.

... then the name of the message with the arguments interspersed ...

```
//  
// CardGameViewController.m  
// Matchismo  
// Copyright (c) 2013 Stanford University.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];  
    [sender setBackgroundImage:(UIImage *) forState:(UIControlState)]  
}  
M void setBackgroundColor:(UIColor *)  
@end M void setBackgroundImage:(UIImage *) forState:(UIControlState)  
M void setBounds:(CGRect)  
  
Sets the background image to use for the specified button state. More...
```

We only have to type the first few letters of the message and Xcode will immediately suggest matching method names.

Type setB and then choose the method setBackgroundImage:forState: from the list that appears.

You can select from the list using the arrow keys and then TAB or you can double-click on the method you want.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A card with the text "A ♠" is displayed on the storyboard.

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];  
    [sender setBackgroundImage:(UIImage *) forState:(UIControlState)]  
}  
@end
```

After choosing the method you want, it should fill that method in and highlight the first argument. This argument is obviously the image to set as the background of the `UIButton`.

For us, that's the `cardImage` local variable.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) ! 1

Automatic > CardGameViewController.m -touchCardButton:

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A Ace of Spades card is displayed on the storyboard.

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

Again, you need only type a couple of characters before Xcode will suggest what you want.

Then double-click on the list or hit TAB once the one you want is selected.

```
#import "CardGameViewController.h"

@interface CardGameViewController : UIViewController

@end

@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];
    [sender setBackgroundImage:cardImage forState:(UIControlState)CALLBACK_API_C_STDCALL(_type, _name)]
    CALLBACK_API_C_STDCALL(_type, _name)
    CALLBACK_API_STDCALL(_type, _name)
    void * calloc(size_t, size_t)
    UIImage * cardImage
    CAST_USER_ADDR_T(a_ptr)
    CATransform3D CATransform3DConcat(CATransform3D a, CATransform3D b)
    bool CATransform3DEqualToTransform(CATransform3D a, CATransform3D b)
    CGAffineTransform CATransform2DConcatAffineTransform(CATransform2D a, CGAffineTransform b)
}
```

Notice that there are LOTS of things that start with the two letters “ca”, but that Xcode is pretty smart about guessing which one you intend based on context.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Matchismo 2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A Ace of Spades card image

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

The `setBackgroundImage:forState:` method asks for the state because you can set different background images for selected, highlighted or disabled states of the `UIButton`.

```
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];  
    [sender setBackgroundImage:cardImage forState:UIControlStateNormal];  
}  
  
@end
```

We're just going to set the background for the default (Normal) state so it will be that way in all button states.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
//  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];  
    [sender setBackgroundImage:cardImage forState:UIControlStateNormal]  
}  
  
@end
```

enum UIControlEvents UIControlEventTouchUpInside
enum UIControlEvents UIControlEventValueChanged
enum UIControlState UIControlStateApplication
enum UIControlState UIControlStateDisabled
enum UIControlState UIControlStateHighlighted
enum UIControlState UIControlStateNormal
enum UIControlState UIControlStateReserved
enum UIControlState UIControlStateSelected

Start typing UIC and you'll get close to what we want (UIControlStateNormal) ...

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

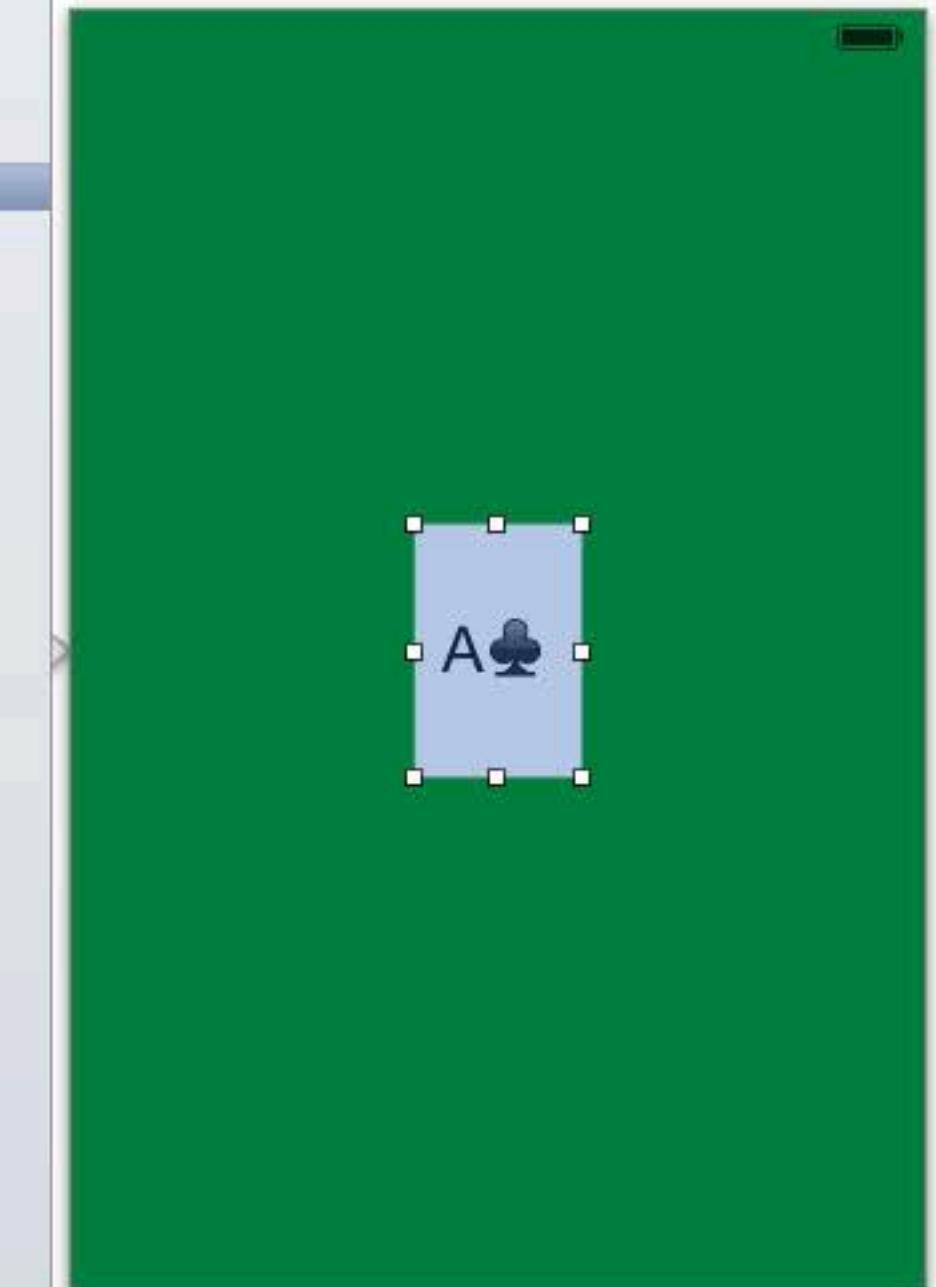
Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products



A card with the value 'A' and suit 'clubs' is displayed on the simulator screen.

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];  
    [sender setBackgroundImage:cardImage forState:UIControlStateNormal]  
}  
  
@end
```

enum UIControlEvents UIControlEventTouchUpInside

enum UIControlEvents UIControlEventValueChanged

enum UIControlState UIControlStateApplication

enum UIControlState UIControlStateDisabled

enum UIControlState UIControlStateHighlighted

enum UIControlState UIControlStateNormal

enum UIControlState UIControlStateReserved

enum UIControlState UIControlStateSelected

Now choose UIControlStateNormal.

Stanford CS193p

Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController()  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = [UIImage imageNamed:@"cardback"];  
    [sender setBackgroundImage:cardImage forState:UIControlStateNormal];  
}  
  
@end
```

Easy!

But we don't actually need this local variable.
We can just embed that message-send in-line.

So select this message send ...

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = ;  
    [sender setBackgroundImage:cardImage forState:UIControlStateNormal];  
}  
  
@end
```

... cut it ...

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = ;  
    [sender setBackgroundImage:cardImage forState:UIControlStateNormal];  
}  
@end
```

... select the variable here ...

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = ;  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"] forState:  
    UIControlStateNormal];  
}  
@end
```

... and paste.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Automatic > CardGameViewController.m -touchCardButton:

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    UIImage *cardImage = ;  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"] forState:  
        UIControlStateNormal];  
}  
@end
```

Now we can select this line ...

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

... and hit delete to get rid of it.

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController()  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"] forState:  
    UIControlStateNormal];  
}  
  
@end
```

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"] forState:  
    UIControlStateNormal];  
}  
@end
```

This line of code is so long
that it is wrapping now.
And not really in a great spot.

Put your cursor here ...

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
        forState:UIControlStateNormal];  
}  
@end
```

... and press the return key.

Notice that Xcode lined up the two colons in the message name!

You should always line up the colons when you manually wrap the arguments of a method (in other words, don't undo what Xcode will do for you).

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files

MatchismoTests

Frameworks

Products

iPhone Retina (3.5-inch)

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

#import "CardGameViewController.h"
@interface CardGameViewController : UIViewController
@end

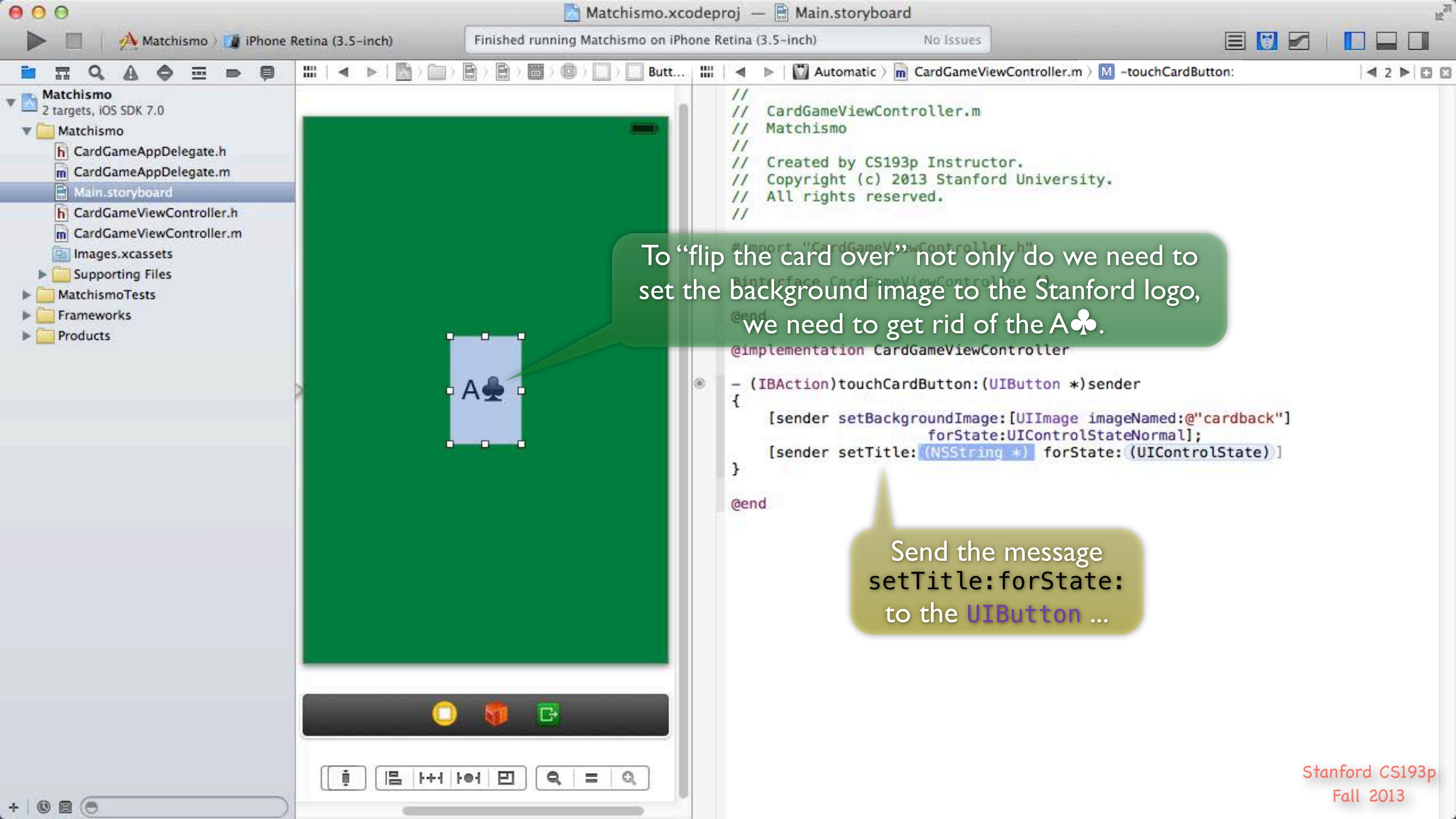
@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
 [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
 forState:UIControlStateNormal];
 [sender setTitle: (NSString *) forState: (UIControlState)]
}

@end

To “flip the card over” not only do we need to set the background image to the Stanford logo, we need to get rid of the A♣.

Send the message `setTitle:forState:` to the `UIButton` ...



Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
        forState:UIControlStateNormal];  
    [sender setTitle:@"" forState:UIControlStateNormal];  
}  
  
@end
```

... with an empty string as the title.

Forgetting the @ before the "" is a very common coding error.
"" without the @ is a const char *.
const char *, while legal, is almost never used in Objective-C.

Matchismo

2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A club card (Ace of Spades) is displayed on the iPhone Retina (3.5-inch) simulator screen.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
@implementation CardGameViewController  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
     forState:UIControlStateNormal];  
    [sender setTitle:@"" forState:UIControlStateNormal];  
}  
@end
```

Again, we specify that we are setting this title for the default (Normal) button state.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Matchismo

Carrier

Let's run again!

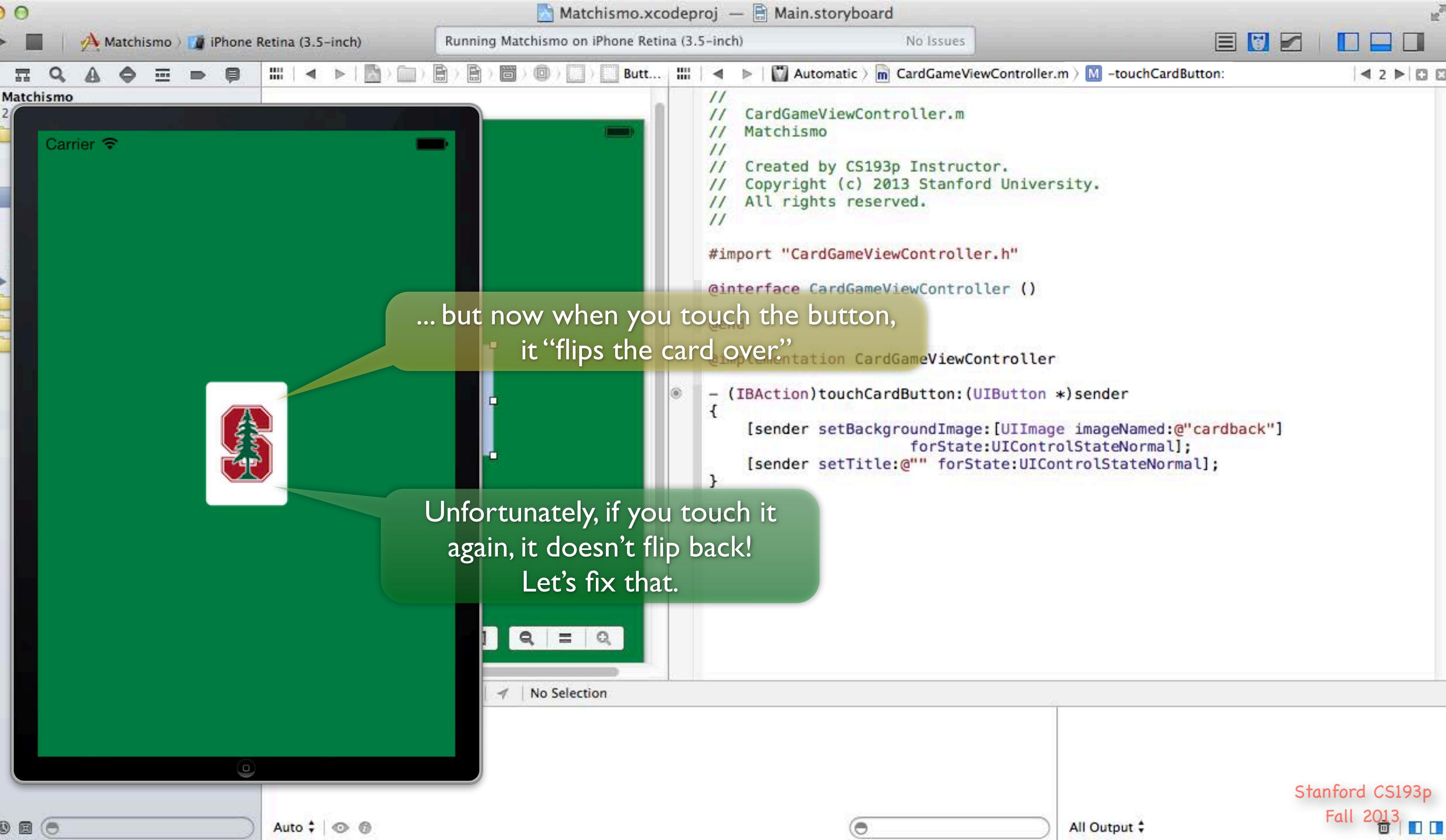
Starts out the same as before ...

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
        forState:UIControlStateNormal];  
    [sender setTitle:@"" forState:UIControlStateNormal];  
}  
  
@end
```

No Selection

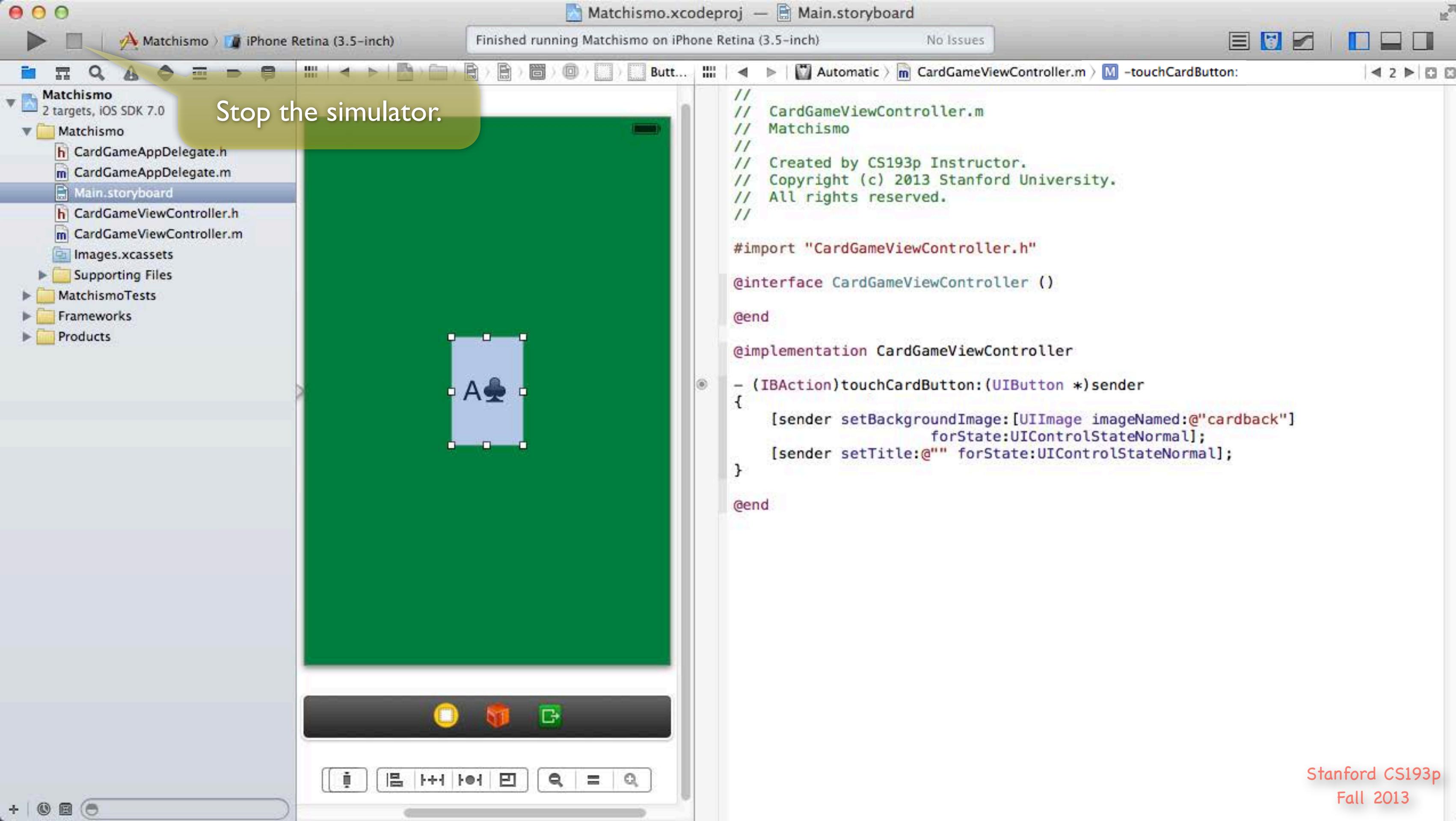
Auto All Output

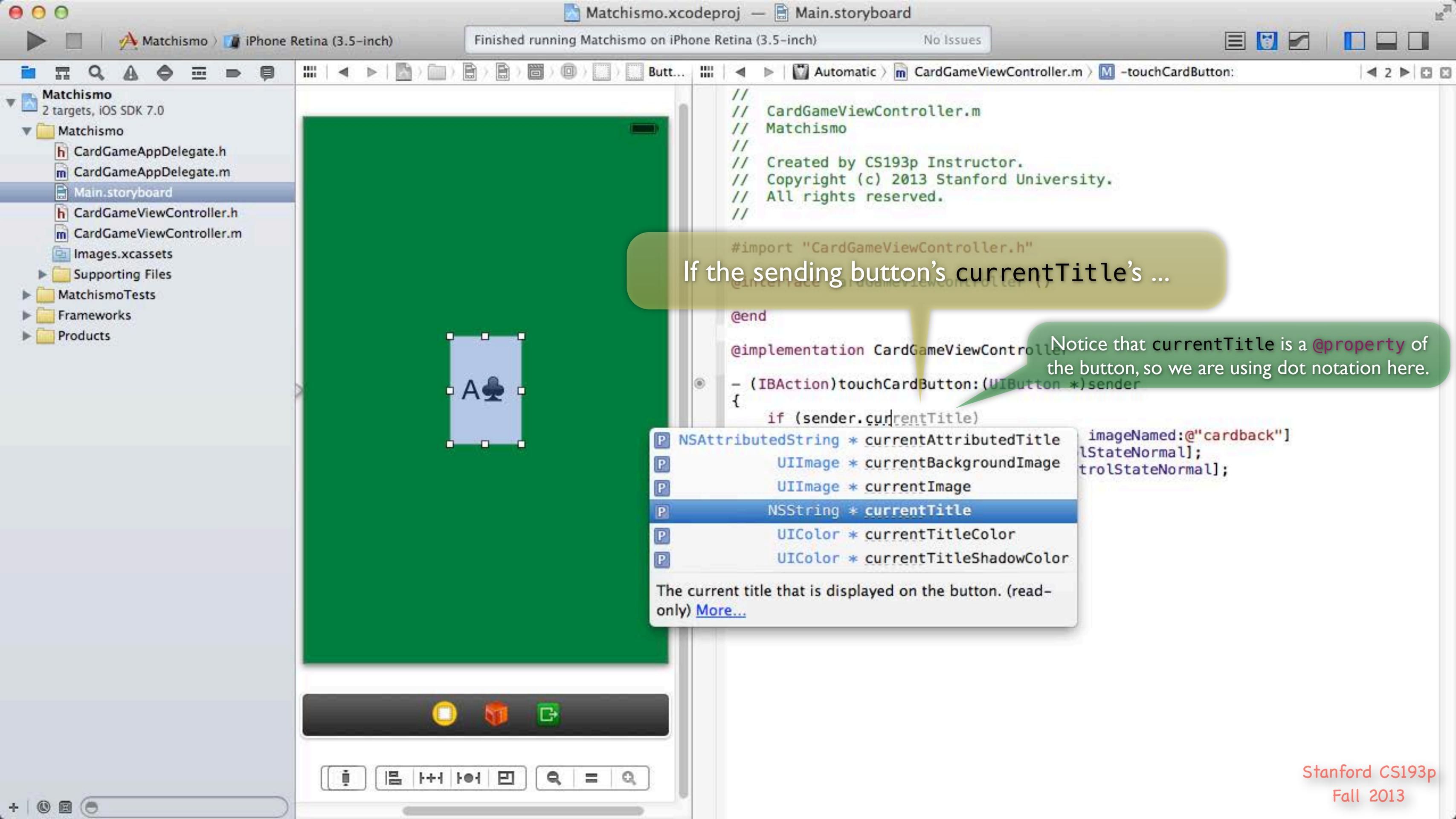
Stanford CS193p Fall 2013



Stanford CS193p

Fall 2013





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

```
#import "CardGameViewController.h"  
If the sending button's current
```

@end

```
@implementation CardGameViewController
```

```
    if (sender.currentTitle)
```

```
P NSAttributedString * currentAttributedTitle  
P UIImage * currentBackgroundImage  
P UIImage * currentImage  
P NSString * currentTitle  
P UIColor * currentTitleColor  
P UIColor * currentTitleShadowColor
```

The current title that is displayed on the button. (read-only) [More...](#)

Notice that `currentTitle` is a `@property` of the button, so we are using dot notation here.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) 1

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
{  
    ...  
}  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if (sender.currentTitle.length)  
        [sender setEnabled:NO];  
    else  
        [sender setEnabled:YES];  
}  
  
@end
```

NSUInteger length

NSUInteger lengthOfBytesUsingEncoding:(NSStringEncoding)

Returns the number of Unicode characters in the receiver. [More...](#)

Checking the length of an `NSString` to see if it is blank is cool because it works whether the string is the empty string (i.e. `@""`) or is `nil`.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

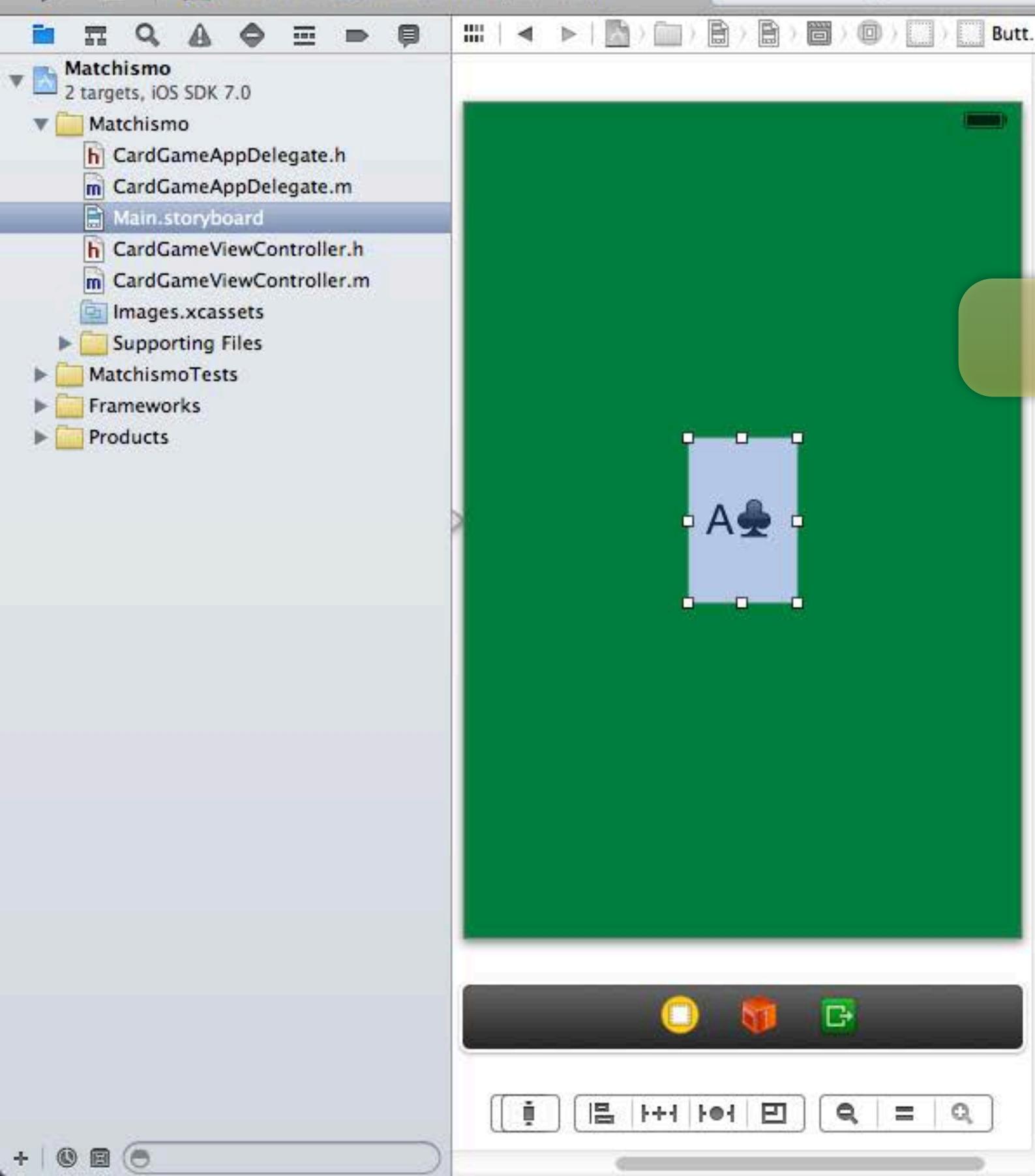
Products

A♣

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length])  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                             forState:UIControlStateNormal];  
    [sender setTitle:@"" forState:UIControlStateNormal];  
}  
@end
```

... (note that we don't have to type the [, since Xcode automatically adds it when we type]) ...

Stanford CS193p
Fall 2013



```
//  
//  CardGameViewController.m  
//  Matchismo  
//  
//  Created by CS193p Instructor.  
//  Copyright (c) 2013 Stanford University.  
//  All rights reserved.  
  
#import "CardGameViewController.h"  
... is non-zero (front is showing), then ...  
@implementation CardGameViewController  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    }  
    @end
```

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

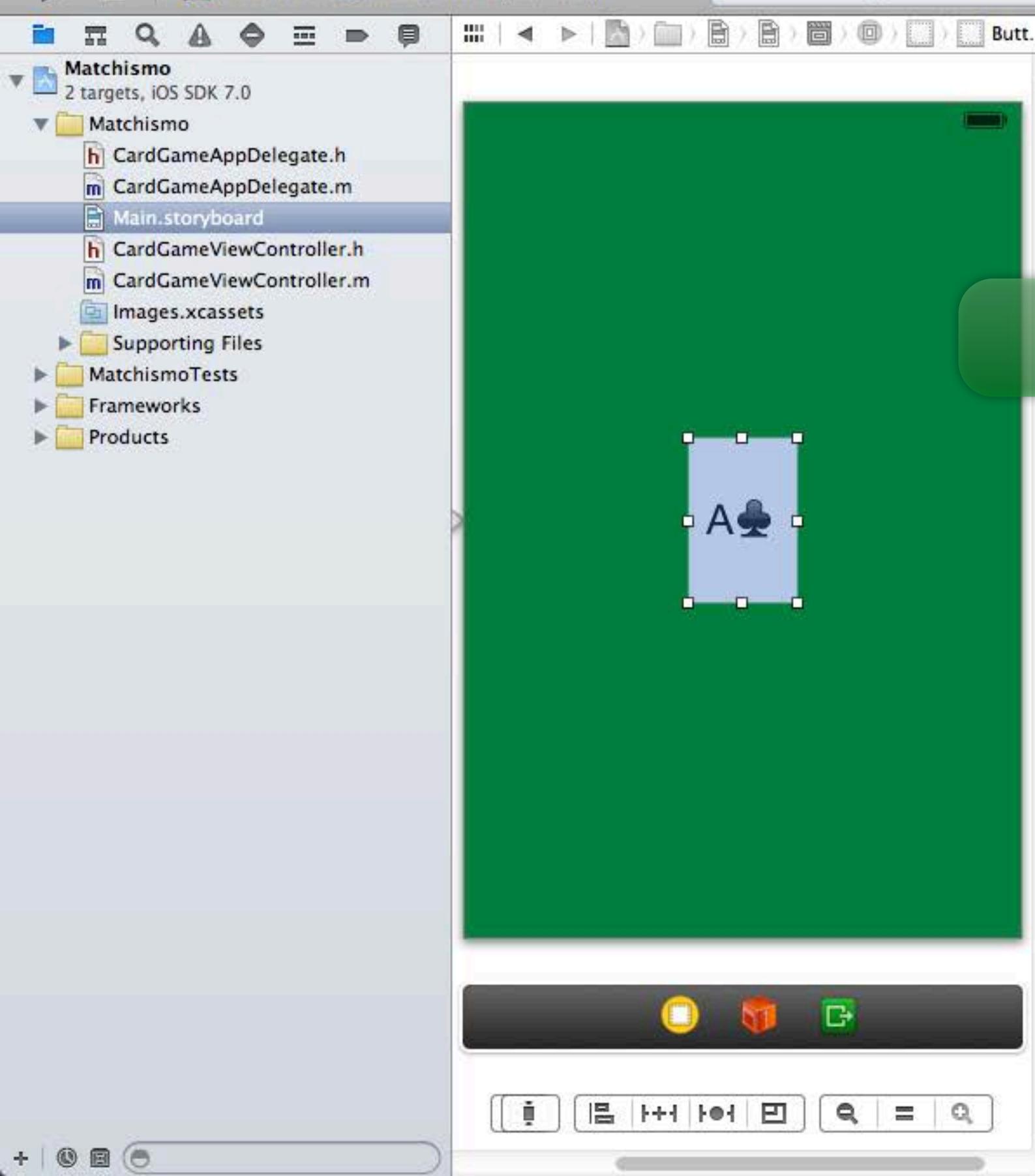
Products

A♣

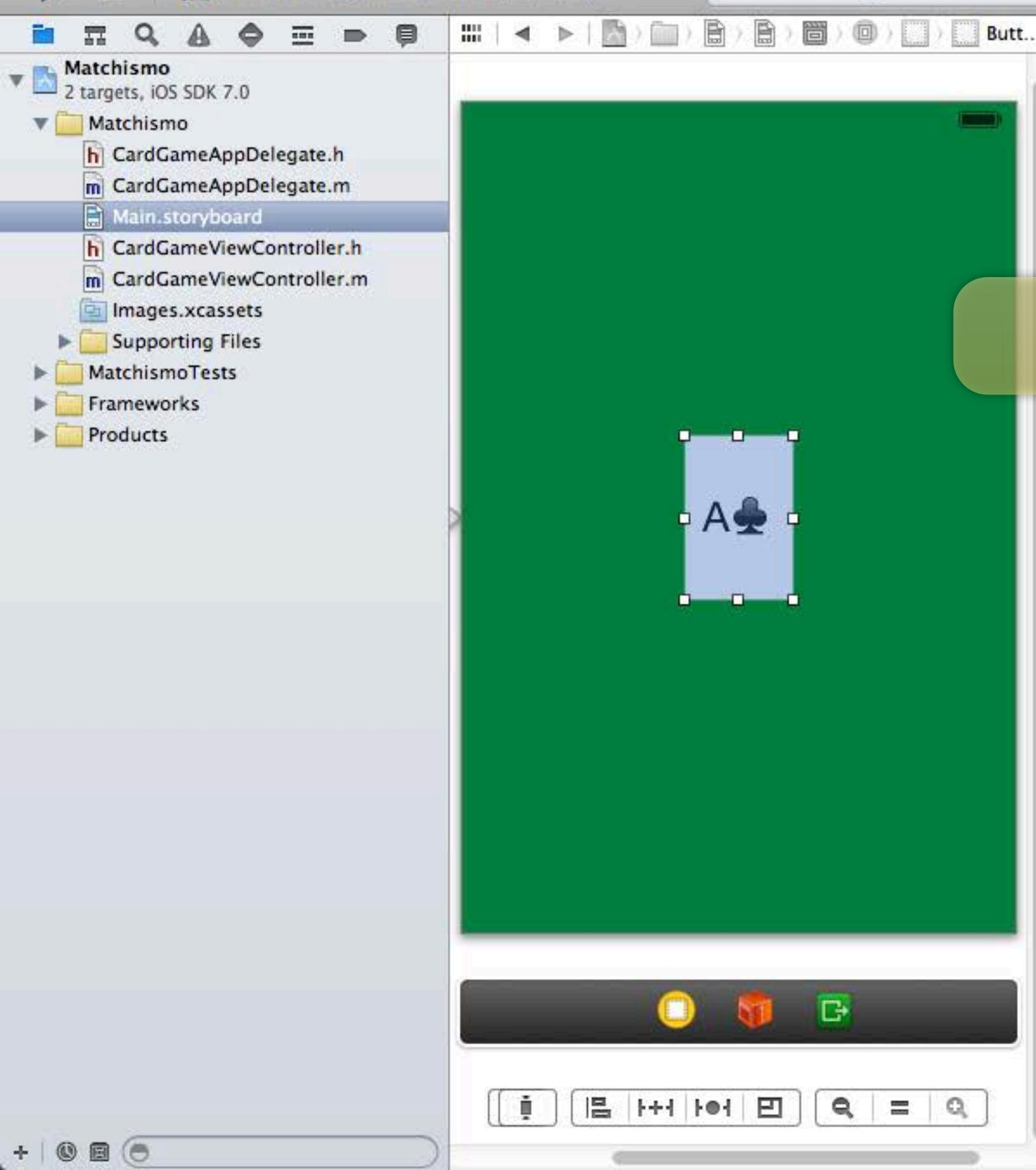
```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
    }  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
        forState:UIControlStateNormal];  
    [sender setTitle:@"" forState:UIControlStateNormal];  
}  
  
@end
```

Move this code inside
the curly braces.

Stanford CS193p
Fall 2013



```
//  
//  CardGameViewController.m  
//  Matchismo  
//  
//  Created by CS193p Instructor.  
//  Copyright (c) 2013 Stanford University.  
//  All rights reserved.  
  
#import "CardGameViewController.h"  
... flip the card over to show it's back ...  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    }  
}  
@end
```



```
//  
//  CardGameViewController.m  
//  Matchismo  
//  
//  Created by CS193p Instructor.  
//  Copyright (c) 2013 Stanford University.  
//  All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
    }  
}  
@end
```

Copy/paste this code into the else ...

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

... and change cardback to cardfront ...

Stanford CS193p
Fall 2013

Matchismo
2 targets, iOS SDK 7.0

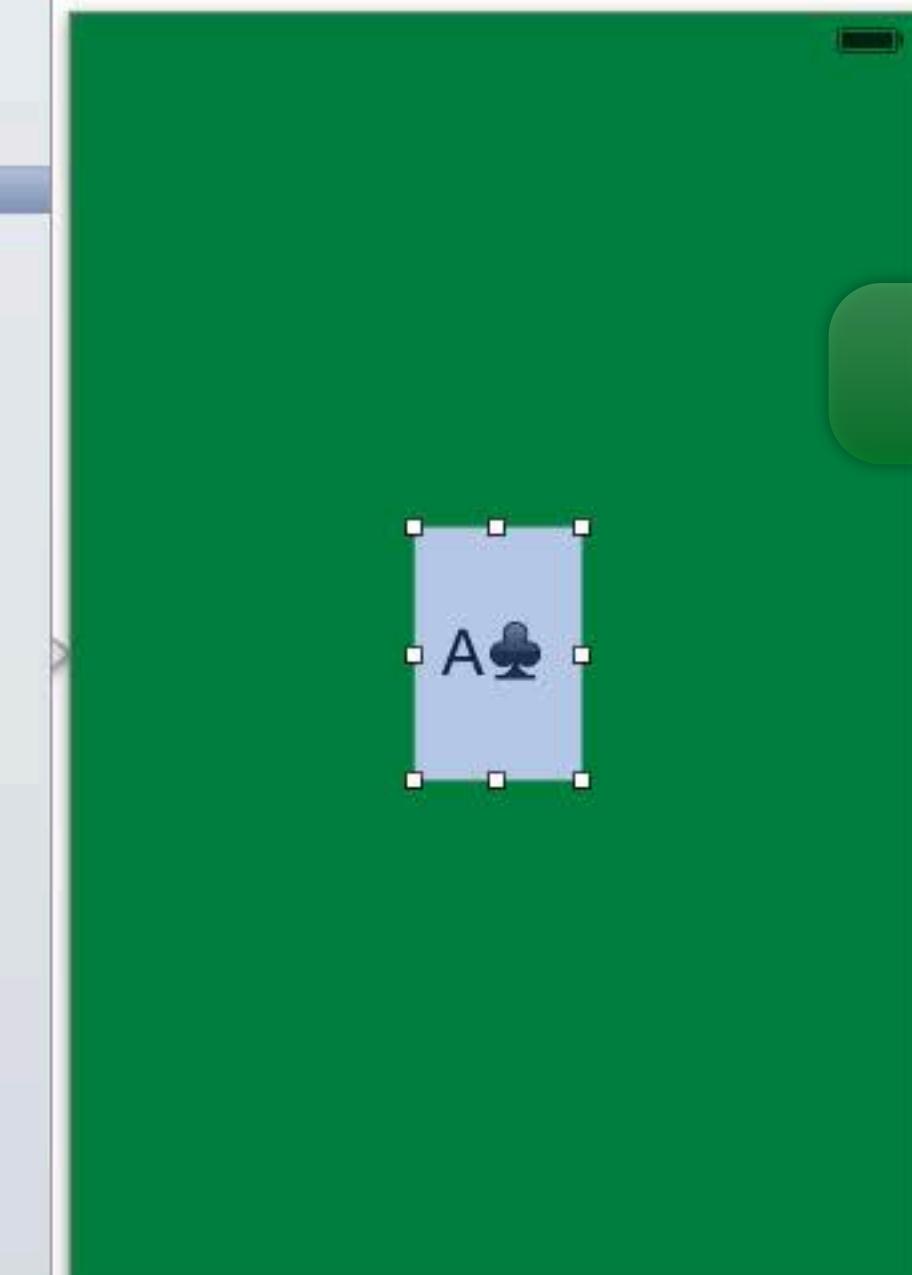
Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files

MatchismoTests

Frameworks

Products



```
//  
//  CardGameViewController.m  
//  Matchismo  
//  
//  Created by CS193p Instructor.  
//  Copyright (c) 2013 Stanford University.  
//  All rights reserved.  
  
#import "CardGameViewController.h"  
... flip back over to the A♣ (the front).  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
}  
@end
```

... and change @"" to @"A♣".

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

Now let's Run again ...

```
//  
// CardGameViewController.m  
// Matchismo  
//  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Matchismo

Carrier

Touch once ...

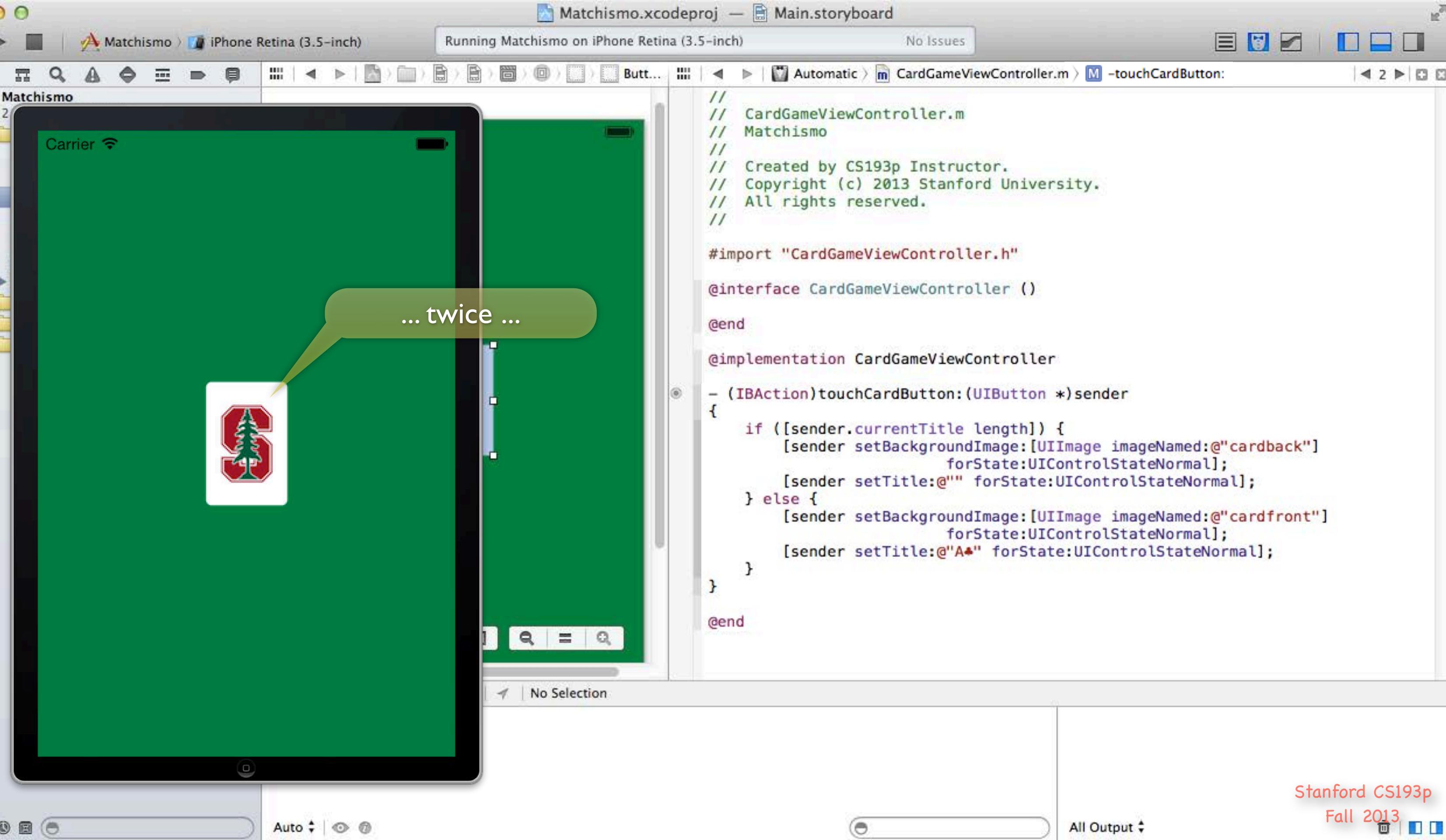
A ♣

```
//  
// CardGameViewController.m  
// Matchismo  
//  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

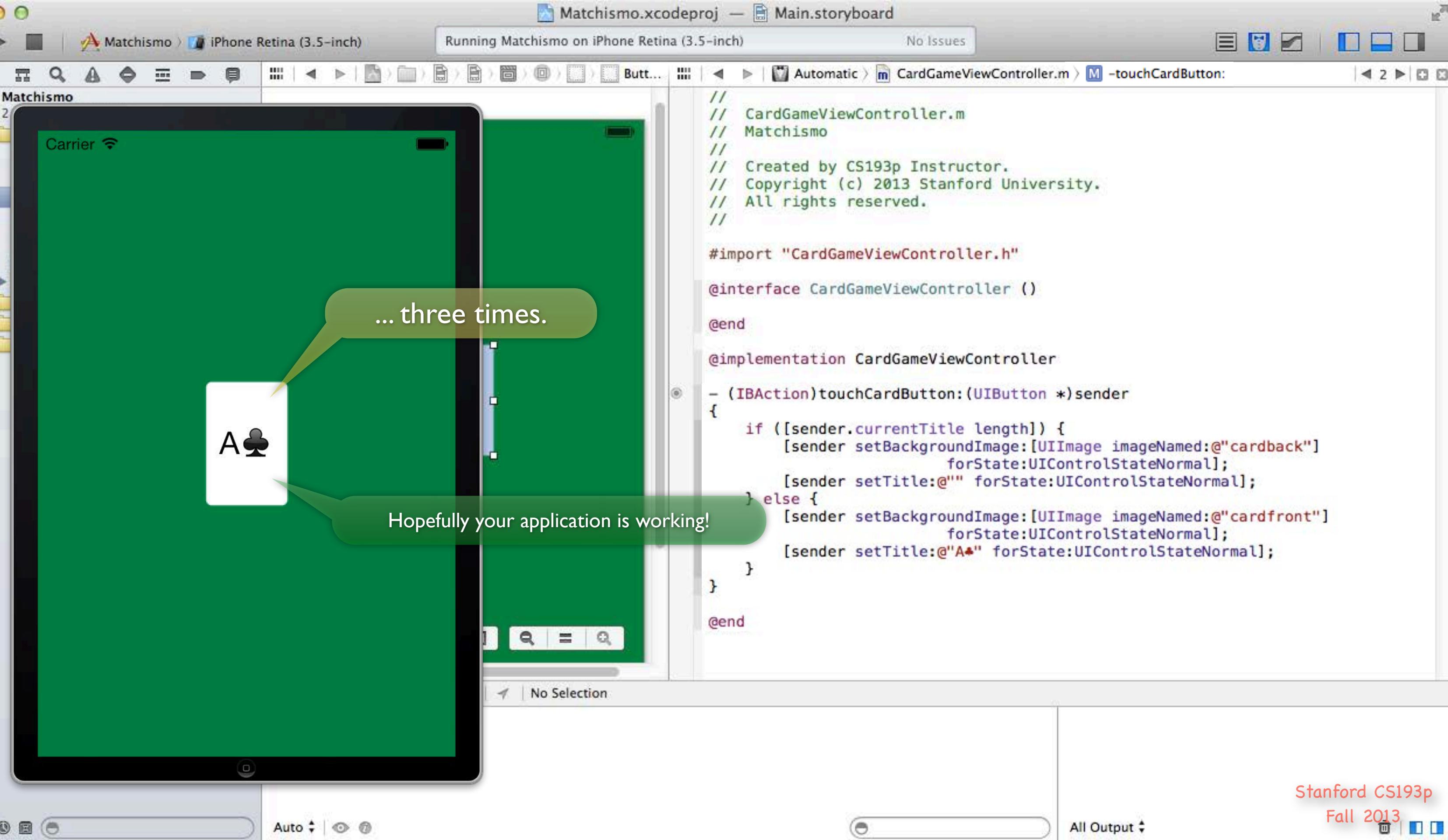
No Selection

Auto All Output

Stanford CS193p Fall 2013



Stanford CS193p
Fall 2013



Stanford CS193p

Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Matchismo 2 targets, iOS SDK 7.0

Main.storyboard

CardGameAppDelegate.h

CardGameAppDelegate.m

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Stop!

// CardGameViewController.m
// Matchismo
// Created by CS193P Instructors
// Copyright (c) 2013 Stanford University.
// All rights reserved.

#import "CardGameViewController.h"

@interface CardGameViewController ()

@end

@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
 if ([sender.currentTitle length]) {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
 forState:UIControlStateNormal];
 [sender setTitle:@"" forState:UIControlStateNormal];
 } else {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]
 forState:UIControlStateNormal];
 [sender setTitle:@"A♣" forState:UIControlStateNormal];
 }
}

Let's take a little timeout to talk about documentation.

There are numerous ways to transition to the documentation, but an easy one is to use the ALT key.

Hold down the ALT key and hover your mouse over something like currentTitle.

A dashed line should appear underneath and the cursor should be a question mark.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic > CardGameViewController.m -touchCardButton:

Main.storyboard

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

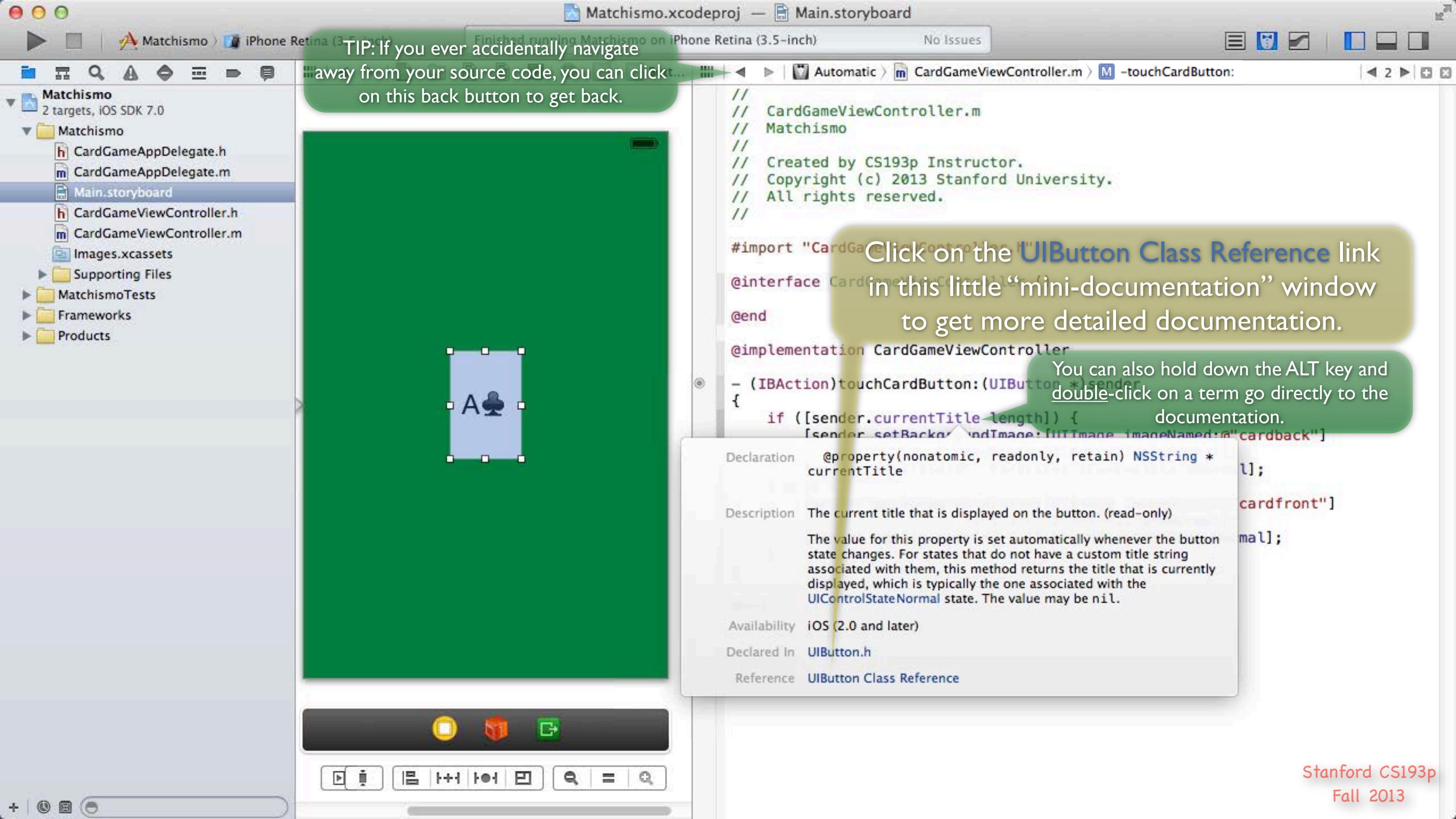
A Club card (A♣) is displayed on the iPhone Retina (3.5-inch) simulator screen.

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
@implementation CardGameViewController  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
        Declaration @property(nonatomic, readonly, retain) NSString *  
        currentTitle  
  
        Description The current title that is displayed on the button. (read-only)  
The value for this property is set automatically whenever the button state changes. For states that do not have a custom title string associated with them, this method returns the title that is currently displayed, which is typically the one associated with the UIControlStateNormal state. The value may be nil.  
  
Availability iOS (2.0 and later)  
Declared In UIButton.h  
Reference UIButton Class Reference  
    }  
}  
    1];  
    cardfront"]  
    mal];
```

ALT-clicking on `currentTitle` will bring up this “mini-documentation” in-line.

retain is the same as strong.
We'll cover the readonly directive next lecture.

Stanford CS193p
Fall 2013



TIP: If you ever accidentally navigate away from your source code, you can click on this back button to get back.

Click on the [UIButton Class Reference](#) link in this little “mini-documentation” window to get more detailed documentation.

You can also hold down the ALT key and double-click on a term go directly to the documentation.



Overview

This is the Documentation window.

Important: This is a preliminary document for an API or technology in development. Although this document has been reviewed for technical accuracy, it is not final. This Apple confidential information is for use only by registered members of the applicable Apple Developer program. Apple is supplying this confidential information to help you plan for the adoption of the technologies and programming interfaces described herein. This information is subject to change, and software implemented according to this document should be tested with final operating system software and final documentation. Newer versions of this document may be provided with future seeds of the API or technology.

An instance of the `UIButton` class implements a button on the touch screen. A button intercepts touch events and sends an action message to a target object when tapped. Methods for setting the target and action are inherited from `UIControl`. This class provides methods for setting the title, image, and other appearance properties of a button. By using these accessors, you can specify a different appearance for each button state.

For information about basic view behaviors, see [View Programming Guide for iOS](#).

For more information about appearance and behavior configuration, see "Buttons".

Tasks

Creating Buttons

+ `buttonWithType:`

Configuring the Button Title

`titleLabel` *property*

`reversesTitleShadowWhenHighlighted` *property*

- `setTitle:forState:`

- `setAttributedTitle:forState:`

- `setTitleColor:forState:`

- `setTitleShadowColor:forState:`

- `titleColorForState:`

- `titleForState:`

- `attributedTitleForState:`

- `titleShadowColorForState:`

Configuring Button Presentation

You should explore what is here.

It is substantial.

Being able to maneuver through the documentation is critical to success in iOS Development.

Provide Feedback

UIButton Class Reference	
Overview	
Tasks	
Creating Buttons	Configuring the Button Title
Configuring Button Presentation	Configuring Edge Insets
Getting the Current State	Getting Dimensions
Deprecated Properties	
Properties	
<code>adjustsImageWhenDisabled</code>	<code>adjustsImageWhenHighlighted</code>
<code>buttonType</code>	<code>contentEdgeInsets</code>
<code>currentAttributedTitle</code>	<code>currentBackgroundImage</code>
<code>currentImage</code>	<code>currentTitle</code>
<code>currentTitleColor</code>	<code>currentTitleShadowColor</code>
<code>imageEdgeInsets</code>	<code>imageView</code>
<code>reversesTitleShadowWhenHighlighted</code>	<code>reversesTitleShadowWhenHighlighted</code>
<code>showsTouchWhenHighlighted</code>	<code>tintColor</code>
<code>tintColor</code>	<code>titleEdgeInsets</code>
<code>titleLabel</code>	<code>titleLabel</code>
Class Methods	
<code>buttonWithType:</code>	
Instance Methods	
<code>attributedTitleForState:</code>	<code>backgroundImageForState:</code>
<code>backgroundImageForState:</code>	<code>backgroundRectForBounds:</code>
<code>backgroundRectForBounds:</code>	<code>contentRectForBounds:</code>
<code>contentRectForBounds:</code>	<code>imageForState:</code>

Search documentation

Available in iOS 6.0 and later.

Declared In

UIButton.h

setBackgroundImage:forState:

Sets the background image to use for the specified button state.

```
- (void)setBackgroundImage:(UIImage *)image forState:(UIControlState)state
```

Parameters*image*

The background image to use for the specified state.

state

The state that uses the specified image. The values are described in UIControlState.

Discussion

In general, if a property is not specified for a state, the default is to use the UIControlStateNormal value. If the UIControlStateNormal value is not set, then the property defaults to a system value. Therefore, at a minimum, you should set the value for the normal state.

Availability

Available in iOS 2.0 and later.

See Also[- backgroundImageForState:](#)**Related Sample Code**[Accessory](#)[AddMusic](#)[UICatalog](#)**Declared In**

UIButton.h

setImage:forState:

Sets the image to use for the specified state.

```
- (void)setImage:(UIImage *)image forState:(UIControlState)state
```

Parameters

For example, scroll down to
setBackgroundImage:forState:.

[Provide Feedback](#)

	UIButton Class Reference
Overview	
Tasks	
Creating Buttons	
Configuring the Button Title	
Configuring Button Presentation	
Configuring Edge Insets	
Getting the Current State	
Getting Dimensions	
Deprecated Properties	
Properties	
adjustsImageWhenDisabled	
adjustsImageWhenHighlighted	
buttonType	
contentEdgeInsets	
currentAttributedTitle	
currentBackgroundImage	
currentImage	
currentTitle	
currentTitleColor	
currentTitleShadowColor	
imageEdgeInsets	
imageView	
reversesTitleShadowWhenHighlighted	
showsTouchWhenHighlighted	
tintColor	
titleEdgeInsets	
titleLabel	
Class Methods	
buttonWithType:	
Instance Methods	
attributedTitleForState:	
backgroundImageForState:	
backgroundRectForBounds:	
contentRectForBounds:	
imageForState:	



Search documentation



UIImage Class Reference

Next

And get detailed class overviews.

Overview

Important: This is a preliminary document for an API or technology in development. Although this document has been reviewed for technical accuracy, it is not final. This Apple confidential information is for use only by registered members of the applicable Apple Developer program. Apple is supplying this confidential information to help you plan for the adoption of the technologies and programming interfaces described herein. This information is subject to change, and software implemented according to this document should be tested with final operating system software and final documentation. Newer versions of this document may be provided with future seeds of the API or technology.

A `UIImage` object is a high-level way to display image data. You can create images from files, from Quartz image objects, or from raw image data you receive. The `UIImage` class also offers several options for drawing images to the current graphics context using different blend modes and opacity values.

Image objects are immutable, so you cannot change their properties after creation. This means that you generally specify an image's properties at initialization time or rely on the image's metadata to provide the property value. It also means that image objects are themselves safe to use from any thread. The way you change the properties of an existing image object is to use one of the available convenience methods to create a copy of the image but with the custom value you want.

Because image objects are immutable, they also do not provide direct access to their underlying image data. However, you can get an `NSData` object containing either a PNG or JPEG representation of the image data using the `UIImagePNGRepresentation` and `UIImageJPEGRepresentation` functions.

The system uses image objects to represent still pictures taken with the camera on supported devices. To take a picture, use the `UIImagePickerController` class. To save a picture to the Saved Photos album, use the `UIImageWriteToSavedPhotosAlbum` function.

Images and Memory Management

In low-memory situations, image data may be purged from a `UIImage` object to free up memory on the system. This purging behavior affects only the image data stored internally by the `UIImage` object and not the object itself. When you attempt to draw an image whose data has been purged, the image object automatically reloads the data from its original file. This extra load step, however, may incur a small performance penalty.

You should avoid creating `UIImage` objects that are greater than 1024 x 1024 in size. Besides the large amount of memory such an image would consume, you may run into problems when using the image as a texture in OpenGL ES or when drawing the image to a view or layer. This size restriction does not apply if you are performing code-based manipulations, such as resizing an image larger than 1024 x 1024 pixels by d to a bitmap-backed graphics context. In fact, you may need to resize an image in this manner (or break it into several smaller images) in order to

UIImage Class Reference

Overview

Images and Memory Management
Supported Image Formats

Tasks

Cached Image Loading Routines
Creating New Images
Initializing Images
Image Attributes
Drawing Images

Properties

alignmentRectInsets
capInsets
CGImage
CIImage
duration
imageOrientation
images
renderingMode
resizingMode
scale
size

Class Methods

animatedImageNamed:duration:
animatedImageWithImages:duration:
animatedResizableImageNamed:ca...
animatedResizableImageNamed:ca...
imageNamed:
imageWithCGImage:
imageWithCGImage:scale:orientation:
imageWithCIImage:
imageWithCIImage:scale:orientation:
imageWithContentsOfFile:
imageWithData:
imageWithData:scale:
Provide Feedback

Instance Methods

NSString

Top Hit

[C NSString](#)

API Reference

[C Appendix A: Deprecated NSString Methods](#)[NSString Class Reference](#)[C Appendix A: Deprecated NSString UIKit Additions Methods](#)[NSString UIKit Additions Reference](#)[f CFStringConvertEncodingToNSStringEncoding](#)[f CFStringConvertNSStringEncodingToEncoding](#)[C NSString](#)

SDK Guides

[NSString](#)[Appendix A: Old-Style ASCII Property...](#)[Strings Are Represented by Instances of the NSString Class](#)[Objects Can Represent Primitive Values](#)[NSString from C Strings and Data](#)[Creating Strings](#)[Technical Q&A QA1235](#)[Converting to Precomposed Unicode](#)

Sample Code

[Birthdays](#)[IDEs](#)[LLDBCustomDataFormatter](#)[Languages & Utilities](#)[Show All Results](#)

Supported

Table 1 lists the

Table 1 Supported

Format

Tagged Image

Joint Photogr...

Graphic Inter...

Portable Netw...

Windows Bitm...

Windows Icon

Windows Cursor

XWindow bitmap

.cur

.xbm

You can also search for classes, methods,
or just general topics of interest.

Note: Windows Bitmap Format (BMP) files that are formatted as RGB-565 are converted to ARGB-1555 when they are loaded.

Tasks

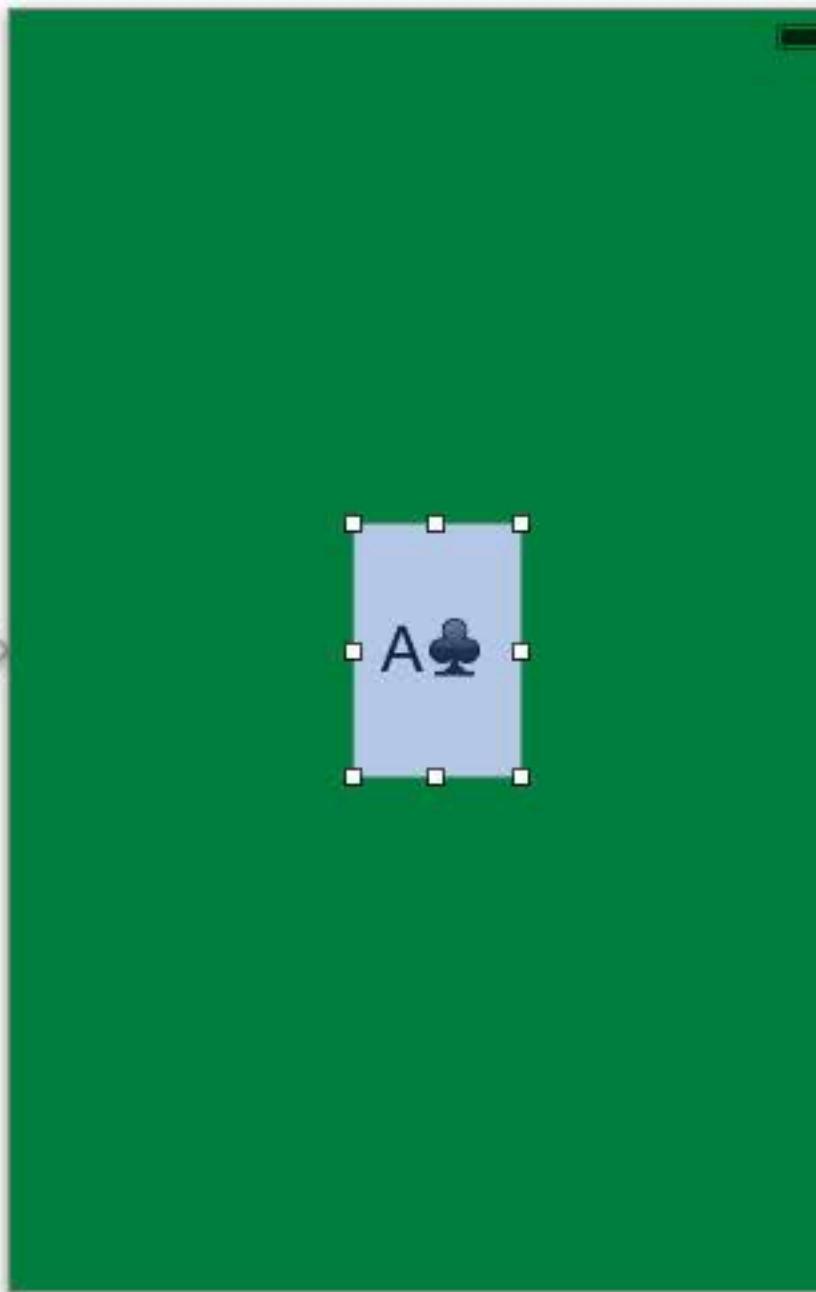
Cached Image Loading Routines

[+ imageNamed:](#)

Creating New Images

[+ imageWithContentsOfFile:](#)[Provide Feedback](#)

- ☰ ⓘ
- UImage Class Reference
- Overview
 - Images and Memory Management
 - Supported Image Formats
- Tasks
 - Cached Image Loading Routines
 - Creating New Images
 - Initializing Images
 - Image Attributes
 - Drawing Images
- Properties
 - alignmentRectInsets
 - capInsets
 - CGImage
 - CIImage
 - duration
 - imageOrientation
 - images
 - renderingMode
 - resizingMode
 - scale
 - size
- Class Methods
 - animatedImageNamed:duration:
 - animatedImageWithImages:duration:
 - animatedResizableImageNamed:ca...
 - animatedResizableImageNamed:ca...
 - imageNamed:
 - imageWithCGImage:
 - imageWithCGImage:scale:orientation:
 - imageWithCIImage:
 - imageWithCIImage:scale:orientation:
 - imageWithContentsOfFile:
 - imageWithData:Stanford CS193p
 - imageWithData:scale:Fall 2013
- Instance Methods



//
// CardGameViewController.m
// Matchismo
// Created by CS193p Team on 10/2/13.
// Copyright (c) 2013 Stanford University.
// All rights reserved.
//
#import "CardGameViewController.h"

@interface CardGameViewController ()

@end

@implementation CardGameViewController

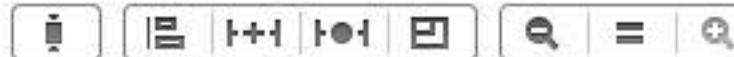
- (IBAction)touchCardButton:(UIButton *)sender

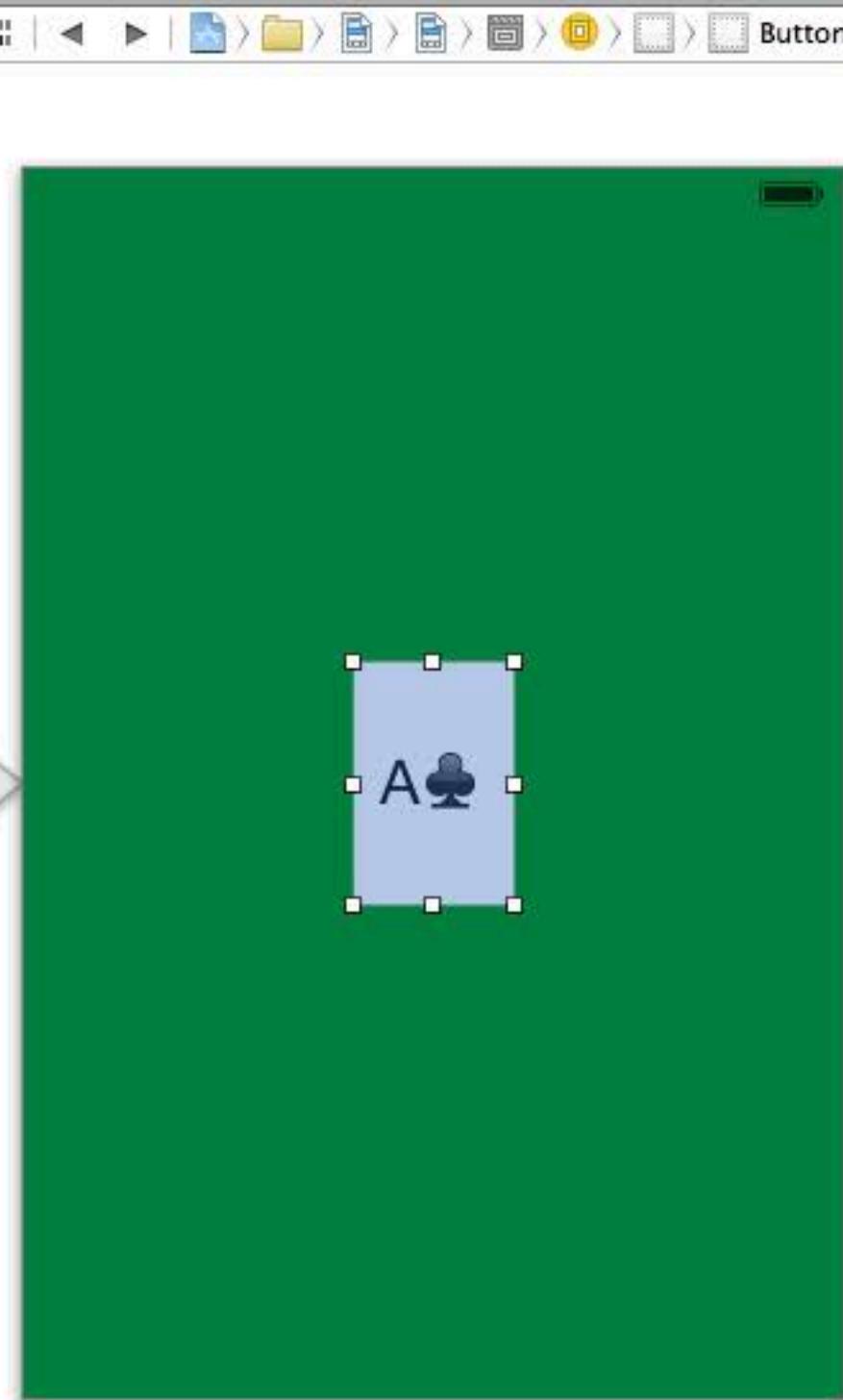
{
 if ([sender.currentTitle length]) {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
 forState:UIControlStateNormal];
 [sender setTitle:@"" forState:UIControlStateNormal];
 } else {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]
 forState:UIControlStateNormal];
 [sender setTitle:@"A♣" forState:UIControlStateNormal];
 }
}

@end

So we've learned how our Controller can *react* to the user manipulating an element of our MVC's View. Next lets learn how the Controller can *proactively* communicate with an element of the View.

Hide the Navigator ...





```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

... and bring back the Utilities area.

Button

Type System

State Config Default

Title Plain

A♣

Font System 24.0

Text Color Black Color

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

Left Right

Control

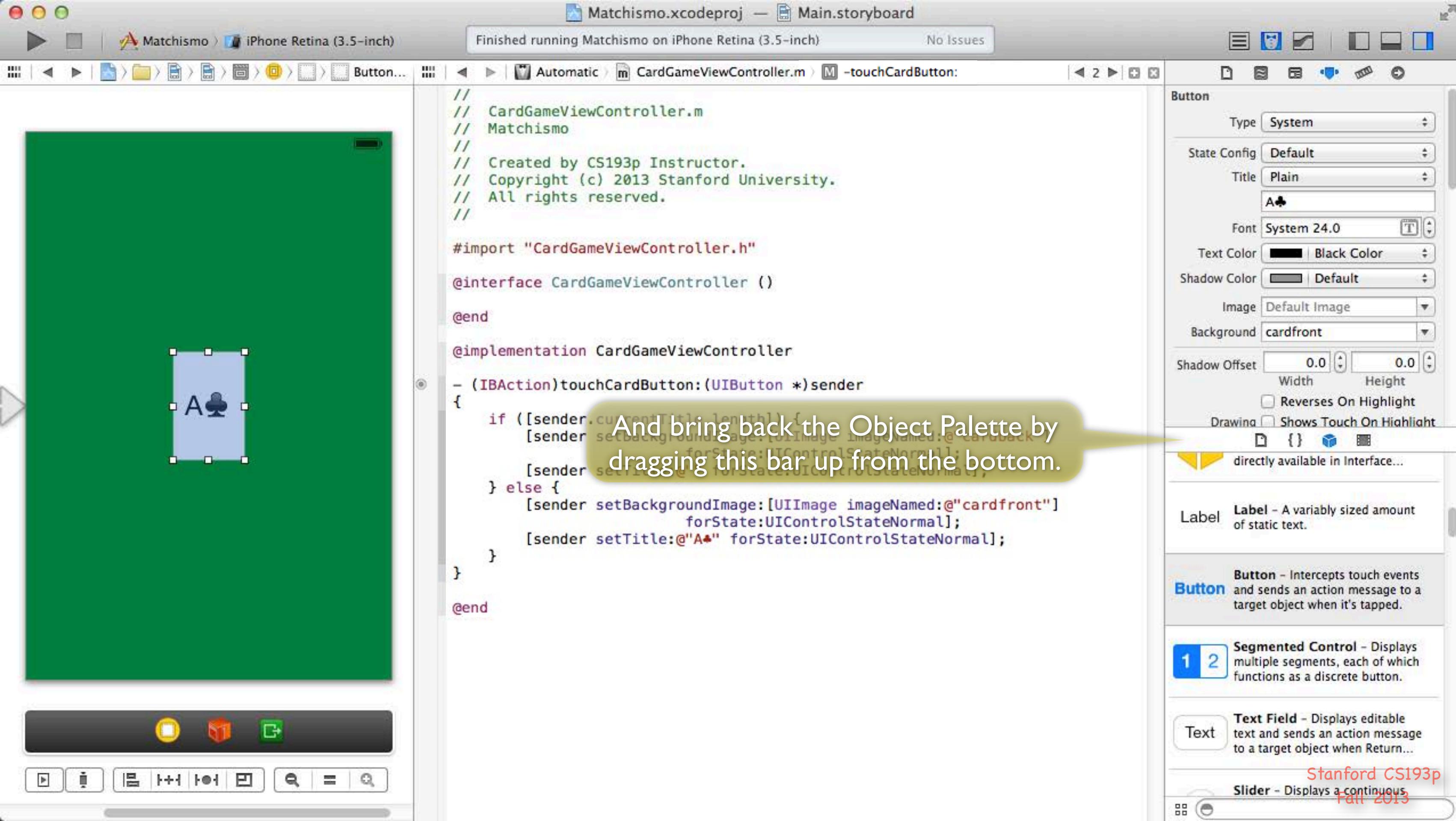
Alignment

Horizontal

Vertical

Content Selected Enabled

Highlighted



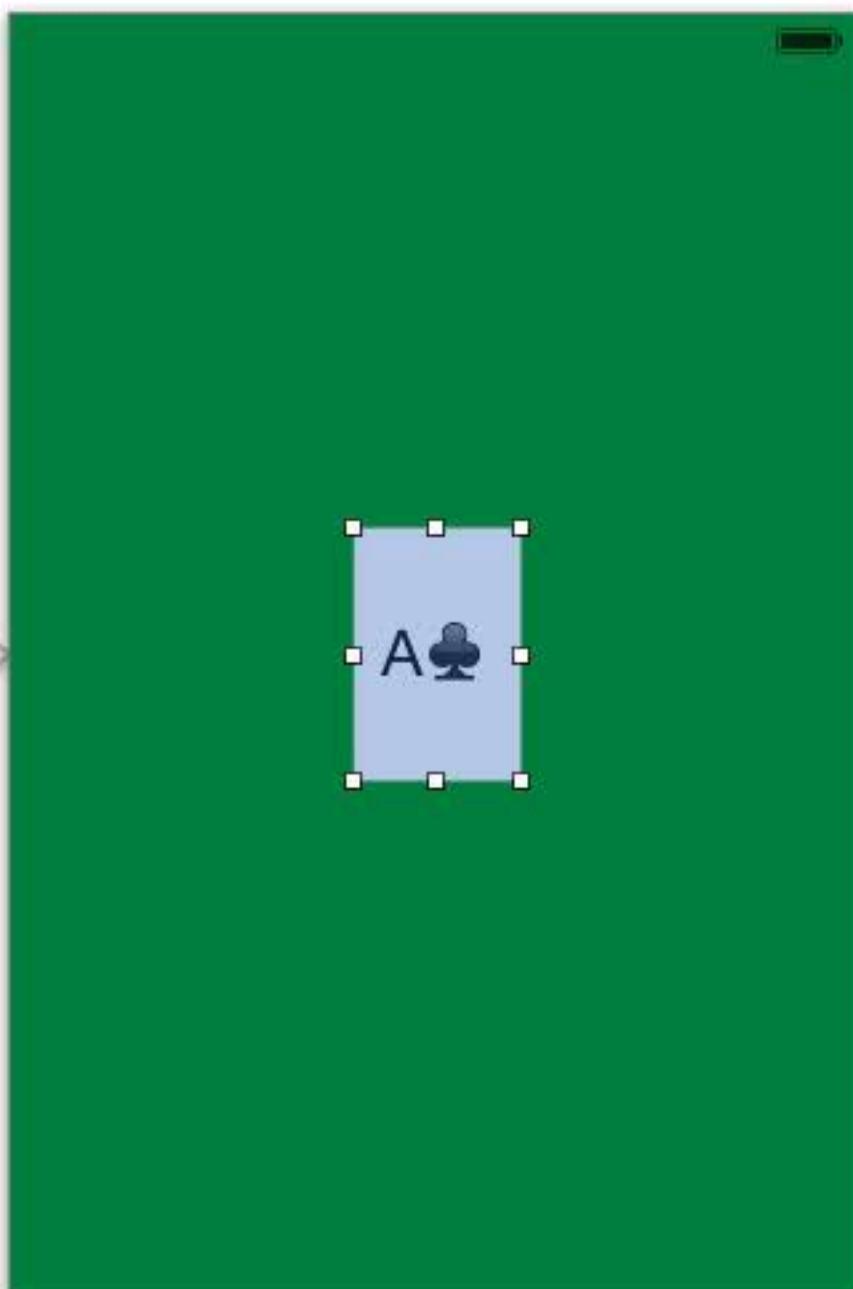
Matchismo > iPhone Retina (3.5-inch)



Button... ▶

Automatic CardGameViewController.m -touchCardButton:

◀ ▶ + X



We're going to have a bit of text in our UI which shows how many times we've flipped the card.
A **UILabel** is the UIKit class we want (it displays small bits of uneditable text).

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

#import "CardGameViewController.h"

@interface CardGameViewController : UIViewController
@end

@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
    if ([sender.currentTitle length]) {
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
                           forState:UIControlStateNormal];
        [sender setTitle:@"" forState:UIControlStateNormal];
    } else {
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]
                           forState:UIControlStateNormal];
        [sender setTitle:@"A♣" forState:UIControlStateNormal];
    }
}

@end
```

Find **UILabel** in the Object Palette (it's right above **UIButton**).

Button

Type System
State Config Default
Title Plain
A♣
Font System 24.0
Text Color Black Color
Shadow Color Default
Image Default Image
Background cardfront
Shadow Offset 0.0 0.0
Width Height
 Reverses On Highlight
Drawing Shows Touch On Highlight
directly available in Interface...

Label

Label - A variably sized amount of static text.

Button

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

Text Field - Displays editable text and sends an action message to a target object when Return...

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

Label

View

Drag one into the lower left corner of the View.
Use the blue guidelines to place it.

Button

Type System

State Config Default

Title Plain

A♣

Font System 24.0

Text Color Black Color

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Shows Touch On Highlight

{} { } { }

directly available in Interface...

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control - Displays multiple segments, each of which functions as a discrete button.

Text Text - Displays editable text and sends an action message to a target object when Return... is pressed.

Slider Slider - Displays a continuous

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

The Attributes Inspector has now changed to show attributes of the Label.

Put this away by dragging it down.

Label

Text Plain

Label

Color Black Color

Font System 17.0

Alignment

Lines 1

Behavior Enabled

Baseline Align Baselines

Line Breaks Truncate Tail

Autoshrink Fixed Font Size

Tighten Letter Spacing

directly available in Interface...

Label Label – A variably sized amount of static text.

Button Button – Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control Segmented Control – Displays multiple segments, each of which functions as a discrete button.

Text Text Field – Displays editable text and sends an action message to a target object when Return... is pressed.

Slider Slider – Displays a continuous slider.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

Label

Text Plain

Color Black Color

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

Shadow Default

Shadow Offset 0 -1

Horizontal Vertical

View

Mode Left

Tag 0

Interaction User Interaction Enabled
 Multiple Touch

Alpha 1

Background Default

Tint Default

Drawing Opaque Hidden
 Clear Graphics Context
 Clip Subviews

Stanford CS193p Fall 2013

Grab a resize handle and make this **UILabel** very wide (again, use the blue guidelines).

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

Label

Text Plain

Color Black Color

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

Shadow Default

Shadow Offset 0 -1

Horizontal Vertical

View

Mode Left

Tag 0

Interaction User Interaction Enabled
 Multiple Touch

Alpha 1

Background Default

Tint Default

Drawing Opaque Hidden
 Clear Graphics Context
 Clip Subviews

Stanford CS193p Fall 2013

A♣

Label

Now double-click on it to change the text ...

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

Label

Text: Plain

Color: Black Color

Font: System 17.0

Alignment: Center

Lines: 1

Behavior: Enabled

Baseline: Align Baselines

Line Breaks: Truncate Tail

Autoshrink: Fixed Font Size

Shadow: Default

Shadow Offset: 0 -1

View

Mode: Left

Tag: 0

Interaction: User Interaction Enabled

Alpha: 1

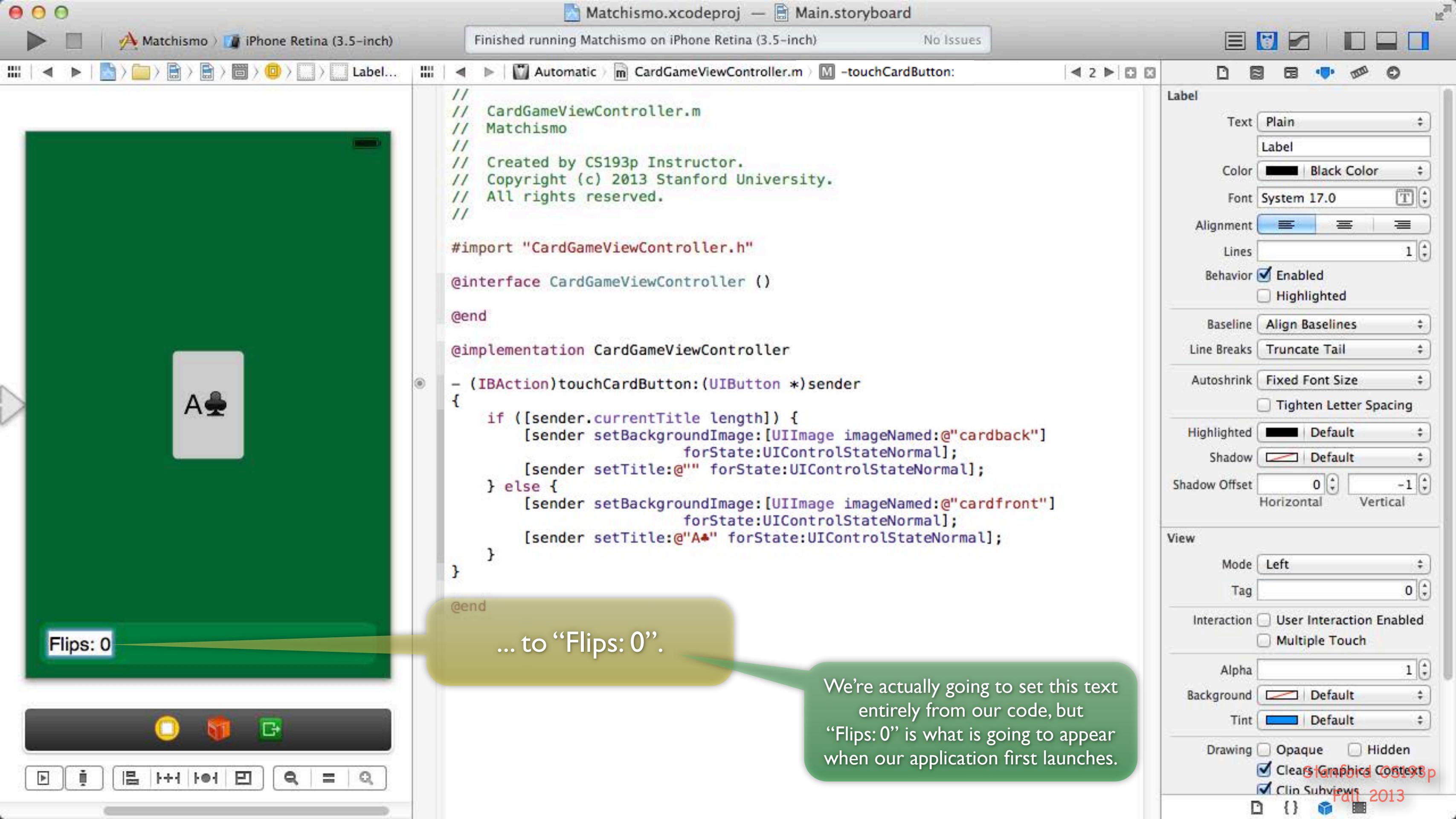
Background: Default

Tint: Default

Drawing: Opaque

Clin Subviews: Checked

Stanford CS193p Fall 2013



... to “Flips: 0”.

We’re actually going to set this text entirely from our code, but “Flips: 0” is what is going to appear when our application first launches.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

Label

Text: Plain

Flips: 0

Color: Black Color

Font: System 17.0

Alignment: Center

Lines: 1

Behavior: Enabled

Baseline: Align Baselines

Line Breaks: Truncate Tail

Autoshrink: Fixed Font Size

Shadow: Default

Shadow Offset: 0 -1

Mode: Left

Tag: 0

Interaction: User Interaction Enabled

Background: Default

Tint: Default

Drawing: Opaque

Clears Graphics Context

Clin Subviews

Stanford CS193p

Fall 2013

A♣

Flips: 0

Now we have to connect this label to our Controller.
We do this by dragging to our code again
(but to the **@interface** instead of the **@implementation**).

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
@end
```

Label

Text Plain

Flips: 0

Color Black Color

Font System 17.0

Alignment

Lines 1

Behavior Enabled

Baseline Align Baselines

Line Breaks Truncate Tail

Autoshrink Fixed Font Size

Tighten Letter Spacing

Highlighted Default

Shadow Default

Shadow Offset 0 -1

View

Mode Left

Tag 0

User Interaction Enabled

Multiple Touch

Alpha 1

Background Default

Tint Default

Opaque

Hidden

Clears Graphics Context

Clin Subviews

A ♠

Flips: 0

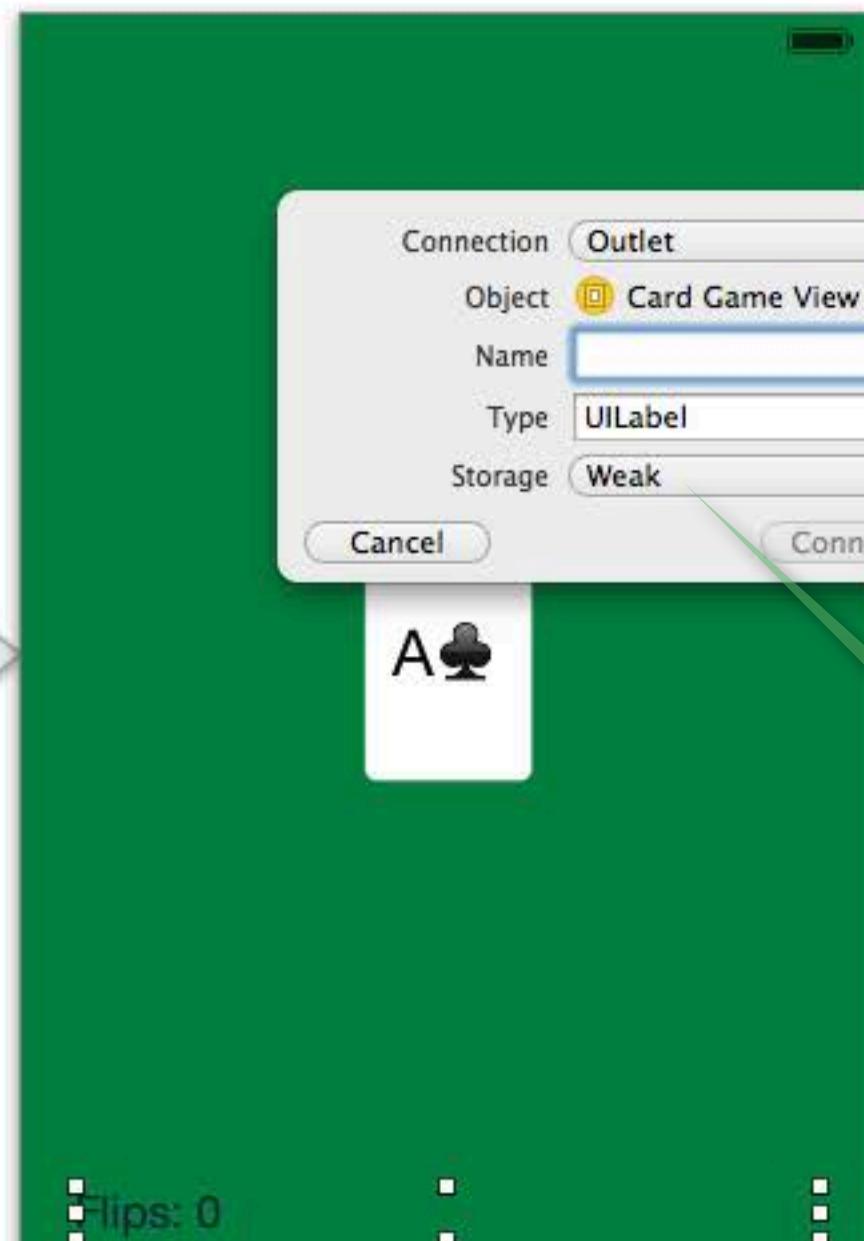
Hold down the CTRL key while dragging the mouse from the label to the code (then let go). Since we're creating a **@property** here, we drag somewhere between the **@interface** and the **@end**.

This **@property** we're going to create is called an “outlet property” or “outlet” for short.

Stanford CS193p Fall 2013

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController ()  
  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
  
@end
```

The `@property` created by this process is `weak` because the MVC's View already keeps a `strong` pointer to the `UILabel`, so there's no need for the Controller to do so as well. And if the `UILabel` ever left the View, the Controller most likely wouldn't want a pointer to it anyway (but if you did want to keep a pointer to it even if it left the View, you could change this to `strong` (very rare)).



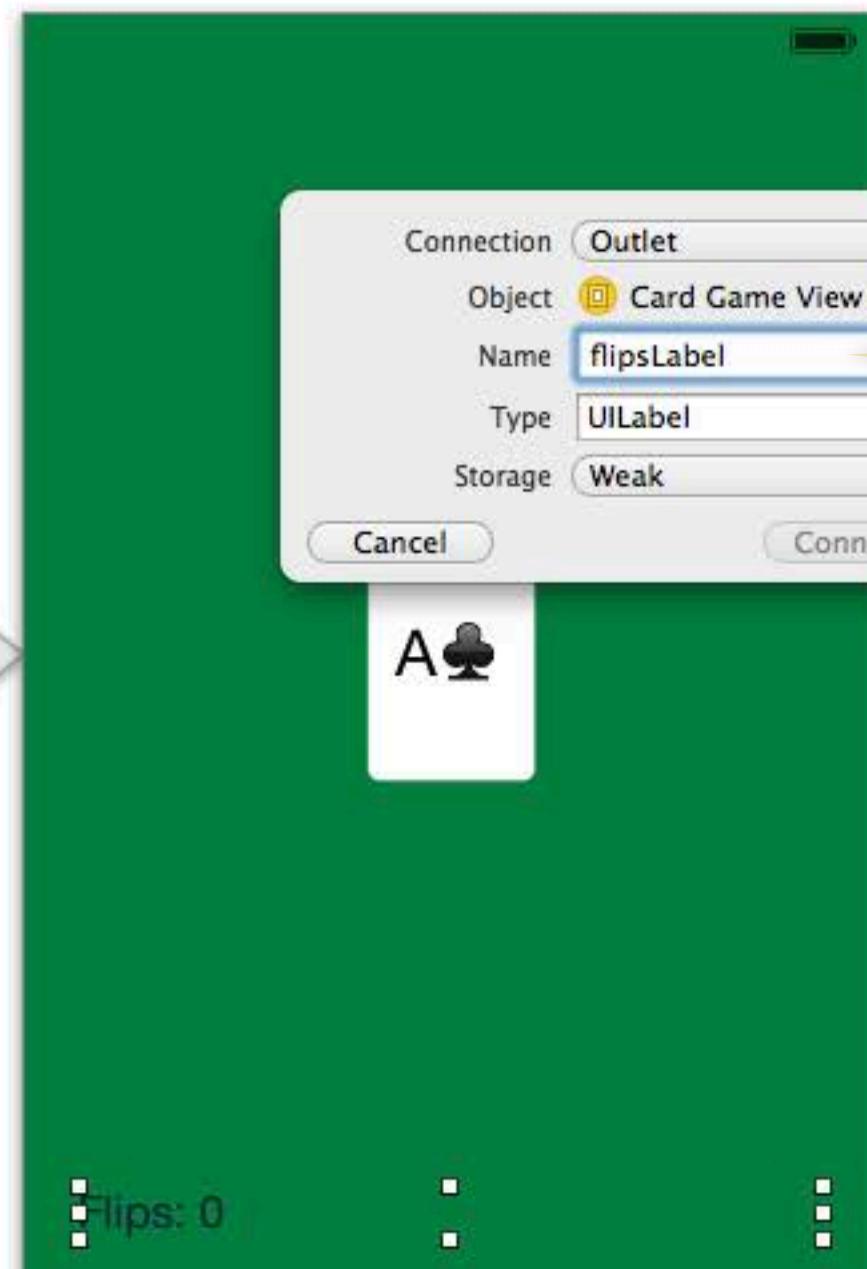
Label
Text Plain
Flips: 0
Color Black Color
Font System 17.0
Alignment
Lines 1
Behavior <input checked="" type="checkbox"/> Enabled <input type="checkbox"/> Highlighted
Baseline Align Baselines
Line Breaks Truncate Tail
Autoshrink Fixed Font Size
<input type="checkbox"/> Tighten Letter Spacing
Highlighted Default
Shadow Default
Shadow Offset 0 -1

View
Mode Left
Tag 0
Interaction <input type="checkbox"/> User Interaction Enabled <input type="checkbox"/> Multiple Touch
Alpha 1
Background Default
Tint Default
Drawing <input type="checkbox"/> Opaque <input type="checkbox"/> Hidden <input checked="" type="checkbox"/> Clear Graphics Context <input checked="" type="checkbox"/> Clip Subviews

```
//  
// CardGameViewController.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "CardGameViewController.h"  
  
@interface CardGameViewController : UIViewController  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
@end
```

We'll call this **UILabel** `flipsLabel`.

Then press Connect.



Label

Text Plain

Flips: 0

Color Black Color

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

Shadow Default

Shadow Offset 0 -1

Horizontal Vertical

View

Mode Left

Tag 0

Interaction User Interaction Enabled Multiple Touch

Alpha 1

Background Default

Tint Default

Drawing Opaque Hidden

Clear Graphics Context Clip Subviews

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo > iPhone Retina (3.5-inch)

//
// 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;
@end

@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
 if ([sender.currentTitle length]) {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
 forState:UIControlStateNormal];
 [sender setTitle:@"" forState:UIControlStateNormal];
 } else {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]
 forState:UIControlStateNormal];
 [sender setTitle:@"A♦" forState:UIControlStateNormal];
 }
}
@end

Label

Text: Plain
Flips: 0
Color: Black Color
Font: System 17.0
Alignment: Center
Lines: 1
Behavior: Enabled
Baseline: Align Baselines
Line Breaks: Truncate Tail
Autoshrink: Fixed Font Size
Tighten Letter Spacing:
Highlighted: Default
Shadow: Default
Shadow Offset: 0 -1
Horizontal Vertical
View
Mode: Left
Tag: 0
User Interaction Enabled:
Multiple Touch:
Alpha: 1
Background: Default
Tint: Default
Drawing: Opaque: Hidden:
Clears Graphics Context:
Cling Subviews:

A♣

Flips: 0

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Label

Text: Plain
Flips: 0
Color: Black Color
Font: System 17.0
Alignment: Center
Lines: 1
Behavior: Enabled (checked), Highlighted (unchecked)
Baseline: Align Baselines
Line Breaks: Truncate Tail
Autoshrink: Fixed Font Size (checked), Tighten Letter Spacing (unchecked)
Highlighted: Default
Shadow: Default
Shadow Offset: 0 (Horizontal), -1 (Vertical)
View
Mode: Left
Tag: 0
Interaction: User Interaction Enabled (unchecked), Multiple Touch (unchecked)
Alpha: 1
Background: Default
Tint: Default
Drawing: Opaque (unchecked), Hidden (unchecked), Clear Graphics Context (checked), Clip Subviews (checked)

Stanford CS193p
Fall 2013

```
// 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;  
@end

@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
    if ([sender.currentTitle length]) {
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
                           forState:UIControlStateNormal];
        [sender setTitle:@"" forState:UIControlStateNormal];
    } else {
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]
                           forState:UIControlStateNormal];
        [sender setTitle:@"A♣" forState:UIControlStateNormal];
    }
}
@end
```



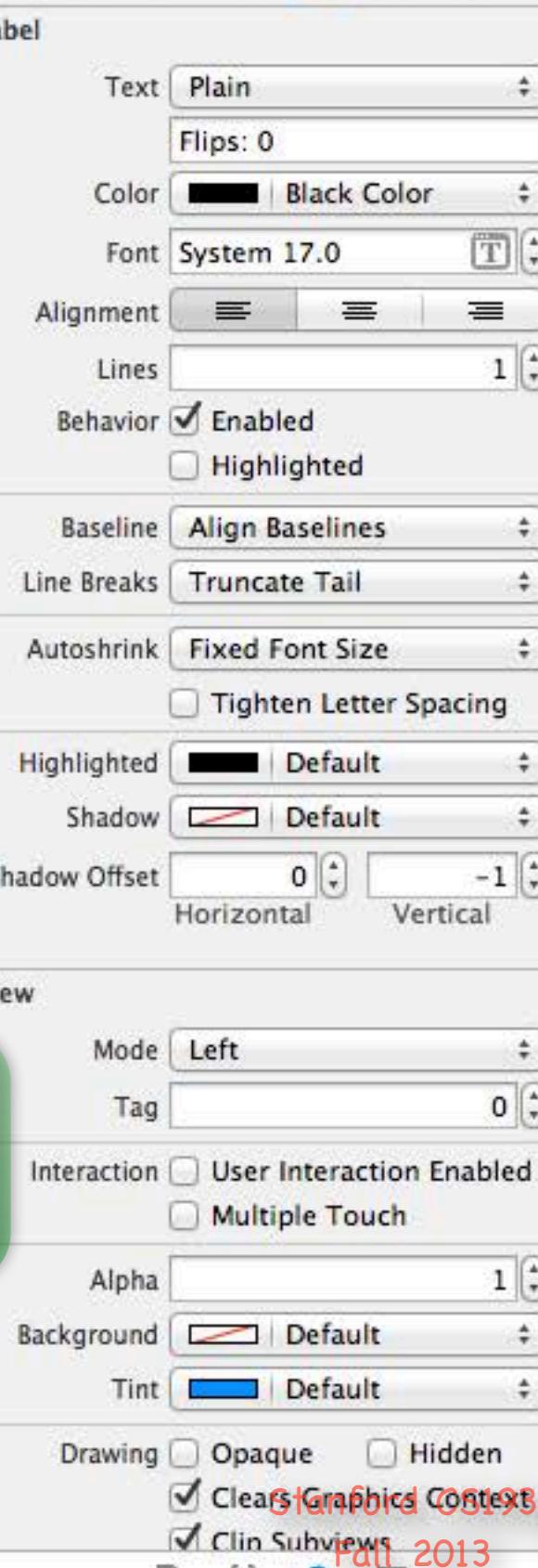
It is also possible to see the connections between your View and Controller by right-clicking on an element in your View.

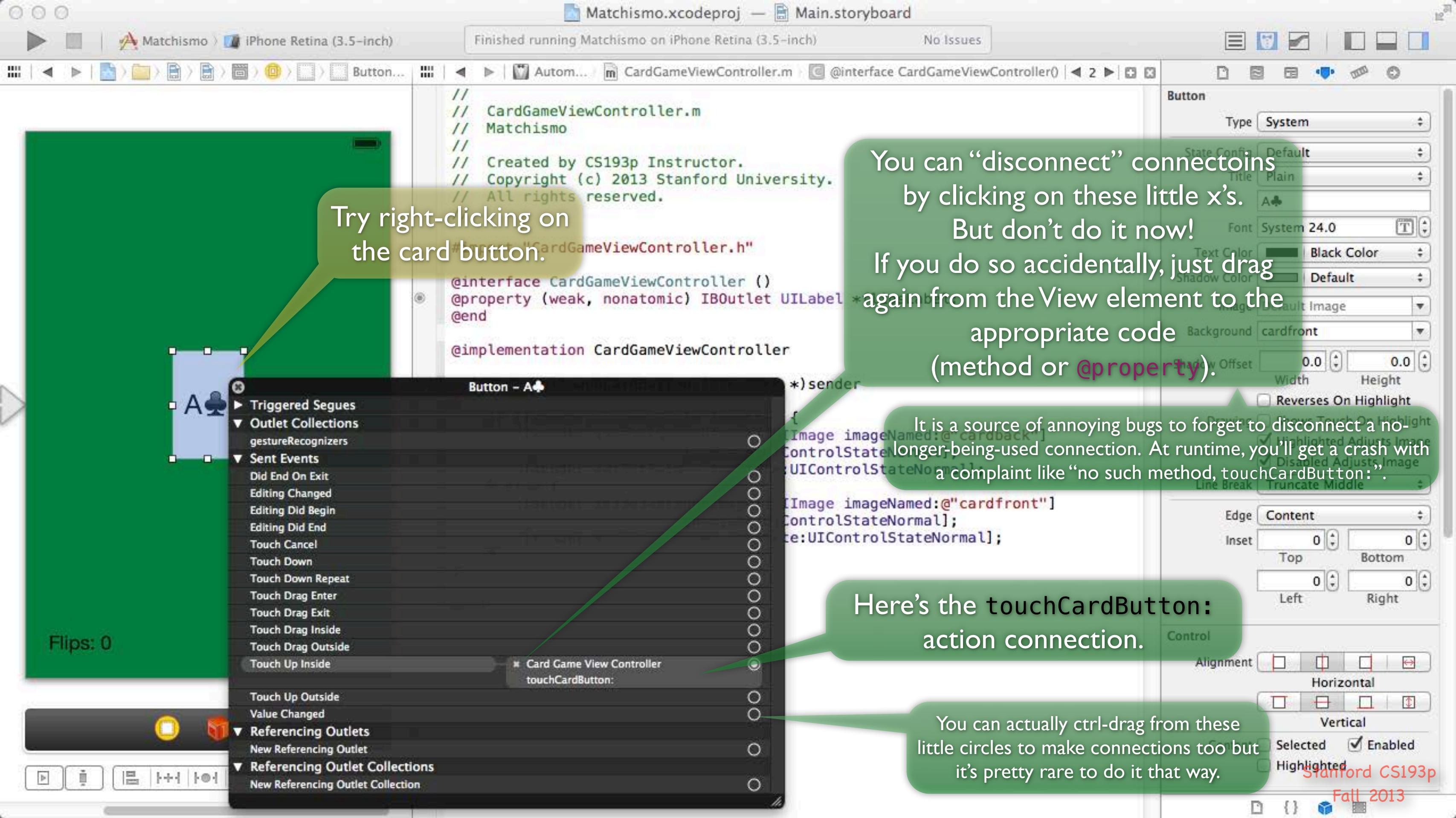
For example, right-click on the Flips:0 label.

Flips: 0



Here are all the connections to/from this **UILabel**.
Notice the outlet **flipsLabel**.





Matchismo > iPhone Retina (3.5-inch)



Button - A♣

Triggered Segues

Outlet Collections

gestureRecognizers

Sent Events

- Did End On Exit
- Editing Changed
- Editing Did Begin
- Editing Did End
- Touch Cancel
- Touch Down
- Touch Down Repeat
- Touch Drag Enter
- Touch Drag Exit
- Touch Drag Inside
- Touch Drag Outside
- Touch Up Inside
- Touch Up Outside
- Value Changed

Referencing Outlets

New Referencing Outlet

Referencing Outlet Collections

New Referencing Outlet Collection

* Card Game View Controller

touchCardButton:

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

#import "CardGameViewController.h"

@interface CardGameViewController : UIViewController
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;
@end

@implementation CardGameViewController

*) sender

```

    {
        [Image imageNamed:@"cardback"]
        [ControlStateNormal];
        [UIControlStateNormal];

        [Image imageNamed:@"cardfront"]
        [ControlStateNormal];
        [UIControlStateNormal];
    }
}

```

If you mouse over something in this dialog,
it will show what it is connected to.

Mouse over this connection ...

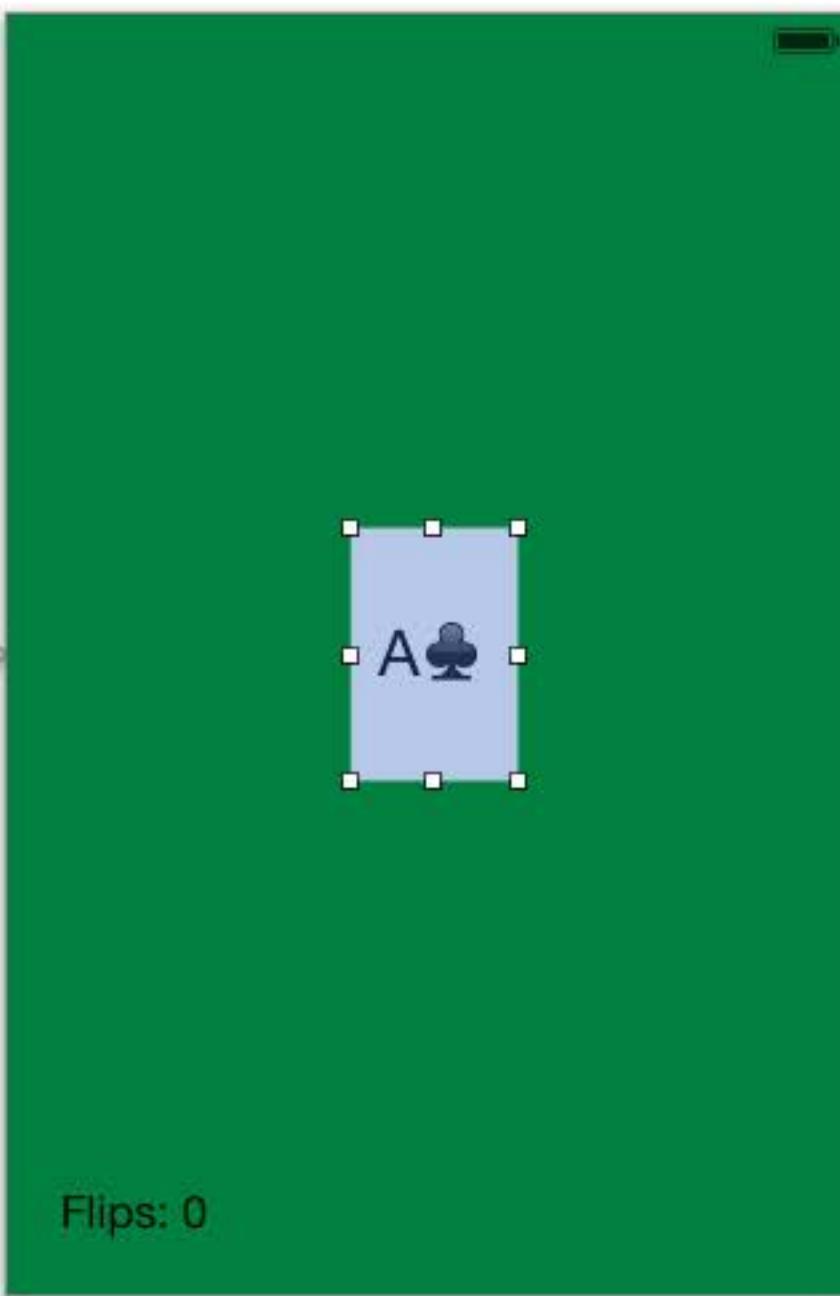
... and you'll see that the entire View highlights.

This mouse over mechanism shows
connections to the Controller by
highlighting the entire View.

Content Selected Enabled
 Highlighted

Stanford CS193p

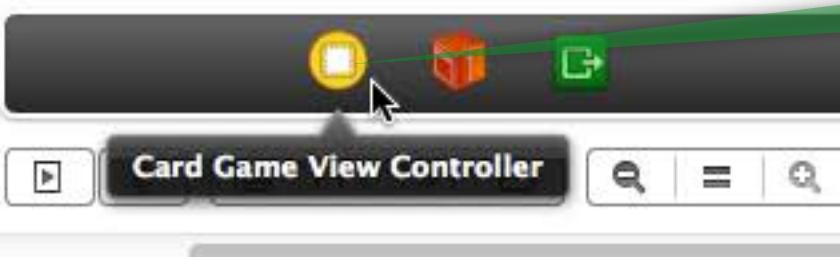
Fall 2013



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;  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
@end
```

You can also “right-click” on your Controller itself (and thus see all connections) by using this icon.



Card Game View Controller



Button

Type: System

State Config: Default

Title: Plain

A ♠

Font: System 24.0

Text Color: Black Color

Shadow Color: Default

Image: Default Image

Background: cardfront

Shadow Offset: 0.0 0.0

Width: 0.0 Height: 0.0

Reverses On Highlight

Drawing: Shows Touch On Highlight Highlighted Adjusts Image Disabled Adjusts Image

Line Break: Truncate Middle

Edge: Content

Inset: 0 0 Top: 0 Bottom: 0

Left: 0 Right: 0

Control

Alignment: Horizontal

Vertical

Content: Selected Enabled

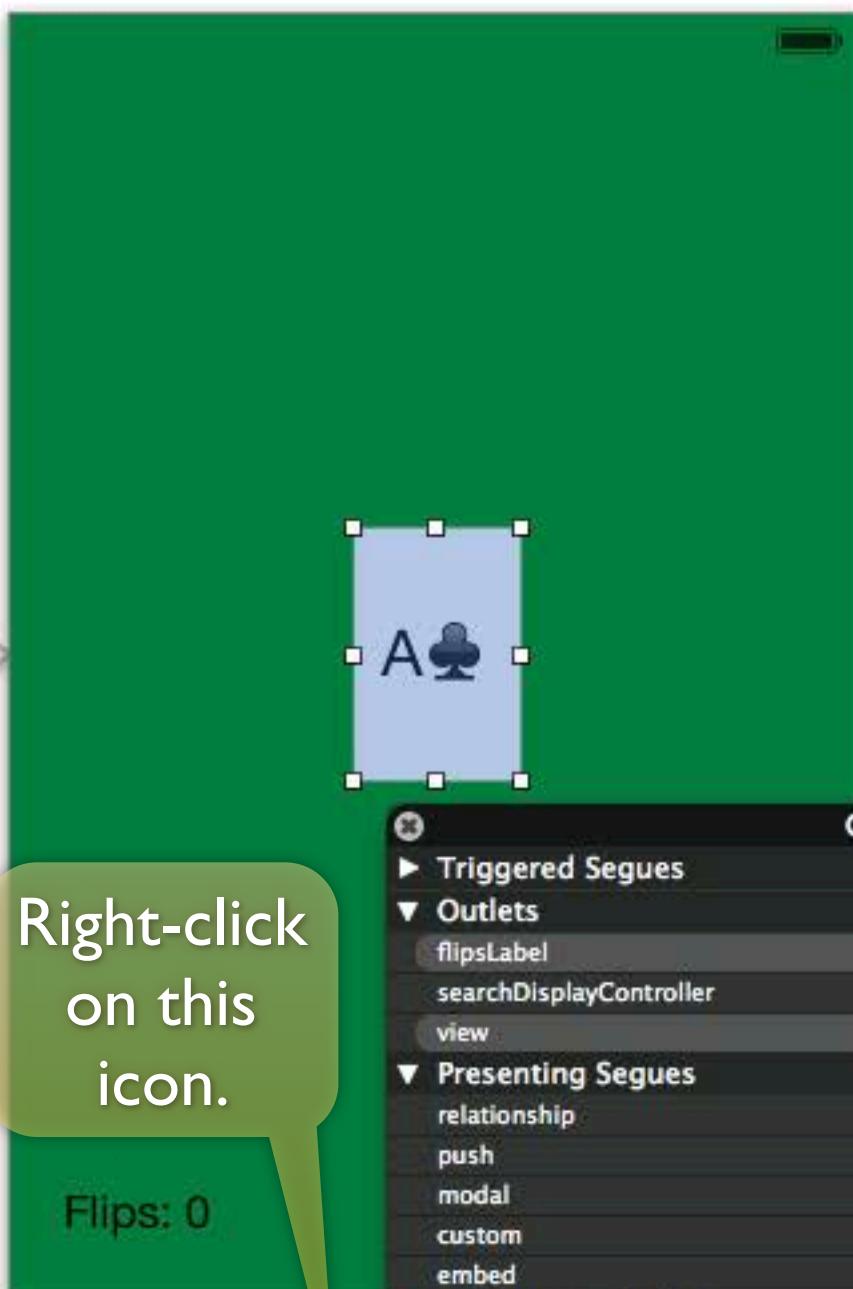
Highlighted

Stanford CS193p

Fall 2013



Matchismo | iPhone Retina (3.5-inch) | Autom... | CardGameViewController.m | @interface CardGameViewController() | 2 | + | X



Right-click
on this
icon.



```

// 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;  

@end

@implementation CardGameViewController  

- (IBAction)touchCardButton:(UIButton *)sender  

{  

    if ([sender.currentTitle length]) {  

        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  

                           forState:UIControlStateNormal];  

        [sender setTitle:@"" forState:UIControlStateNormal];  

    } else {  

        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  

                           forState:UIControlStateNormal];  

        [sender setTitle:@"Flips: 0" forState:UIControlStateNormal];  

    }
}

```

This title bar says that we are looking at
the connections to/from our Controller.

Card Game View Controller

Flips: 0

Triggered Segues

Outlets

flipsLabel

* Label - Flips: 0

searchDisplayController

view

* View

Presenting Segues

relationship

push

modal

custom

embed

Referencing Outlets

New Referencing Outlet

Referencing Outlet Collections

New Referencing Outlet Collection

Received Actions

touchCardButton:

* Button - A♣

Touch Up Inside

Button

Type System

State Config Default

Title Plain

A♣

Font System 24.0

Text Color Black Color

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

0 0

Left Right

Control

Alignment

Horizontal

Vertical

Content Selected

Enabled

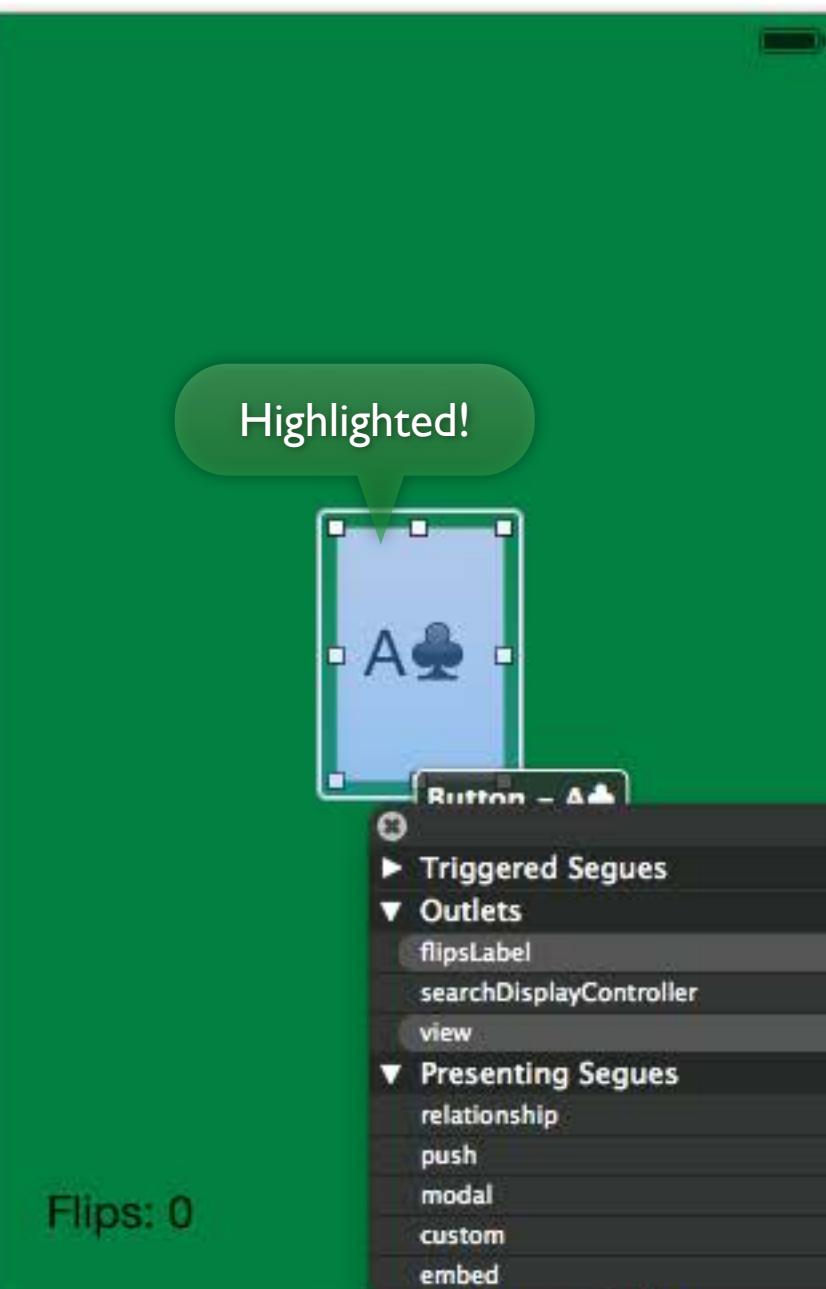
Highlighted

Fall 2013

Stanford CS193p



Matchismo > iPhone Retina (3.5-inch)



```

// 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;  

@end

@implementation CardGameViewController  

- (IBAction)touchCardButton:(UIButton *)sender  

{  

    if ([sender.currentTitle length]) {  

        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  

                           forState:UIControlStateNormal];  

        [sender setTitle:@"" forState:UIControlStateNormal];  

    } else {  

        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  

                           forState:UIControlStateNormal];  

        [sender setTitle:@"A♣" forState:UIControlStateNormal];  

    }  

}

```

Button

Type: System

State Config: Default

Title: Plain

Image: A♣

Font: System 24.0

Text Color: Black Color

Shadow Color: Default

Background: cardfront

Shadow Offset: 0.0 0.0

Width: 0.0 Height: 0.0

Reverses On Highlight

Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break: Truncate Middle

Edge: Content

Inset: 0 0

Top: 0 Bottom: 0

Left: 0 Right: 0

Control

Alignment: Horizontal

Vertical

Content: Selected Enabled

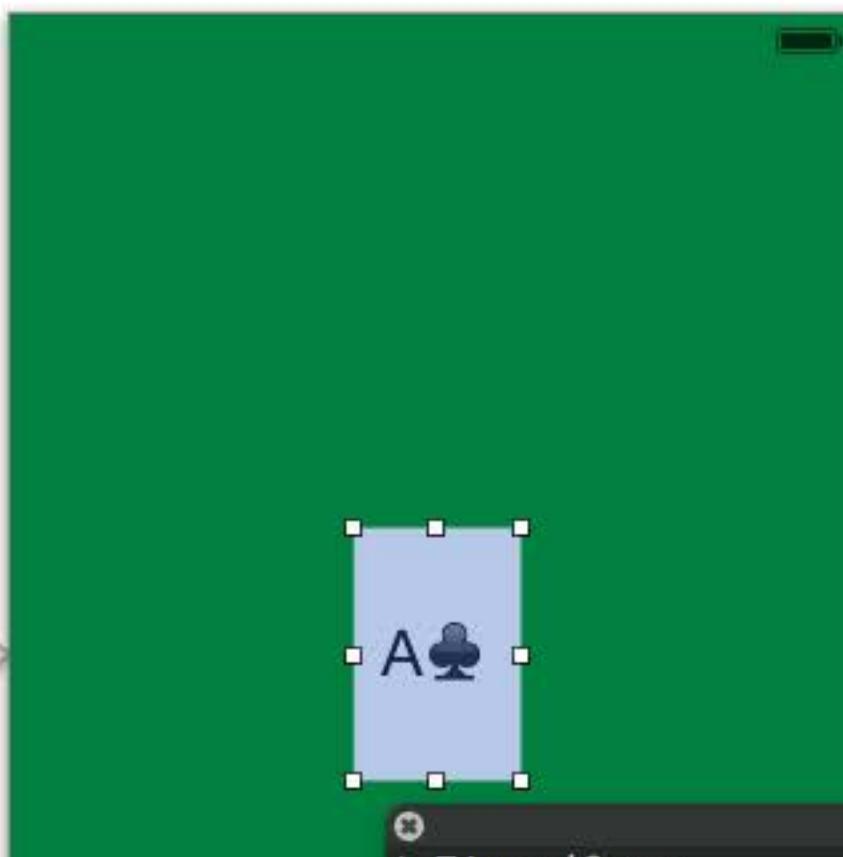
Highlighted

Mouse over.

Stanford CS193p
Fall 2013



Matchismo > iPhone Retina (3.5-inch) | Button... | Autom... | CardGameViewController.m | @interface CardGameViewController()



Highlighted!

Flips: 0

Card Game View Controller

Triggered Segues

Outlets

flipsLabel

searchDisplayController

view

Presenting Segues

relationship

push

modal

custom

embed

Referencing Outlets

New Referencing Outlet

Referencing Outlet Collections

New Referencing Outlet Collection

Received Actions

touchCardButton:

* Button - A ♠
Touch Up Inside

```

// 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;  

@end

@implementation CardGameViewController  

- (IBAction)touchCardButton:(UIButton *)sender  

{  

    if ([sender.currentTitle length]) {  

        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  

                           forState:UIControlStateNormal];  

        [sender setTitle:@"" forState:UIControlStateNormal];  

    } else {  

        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  

                           forState:UIControlStateNormal];  

        [sender setTitle:@"A ♠" forState:UIControlStateNormal];  

    }
}

```

Mouse over

This is an automatically connected outlet from
your Controller to the top-level of your View.
We'll talk about that later in the course.

Button

Type System

State Config Default

Title Plain



Font System 24.0

Text Color Black Color

Shadow Color Default

Image Default Image

Background cardfront

Shadow Offset 0.0 0.0

Width Height

Reverses On Highlight

Shows Touch On Highlight

Highlighted Adjusts Image

Disabled Adjusts Image

Line Break Truncate Middle

Edge Content

Inset 0 0

Top Bottom

0 0

Left Right

Horizontal

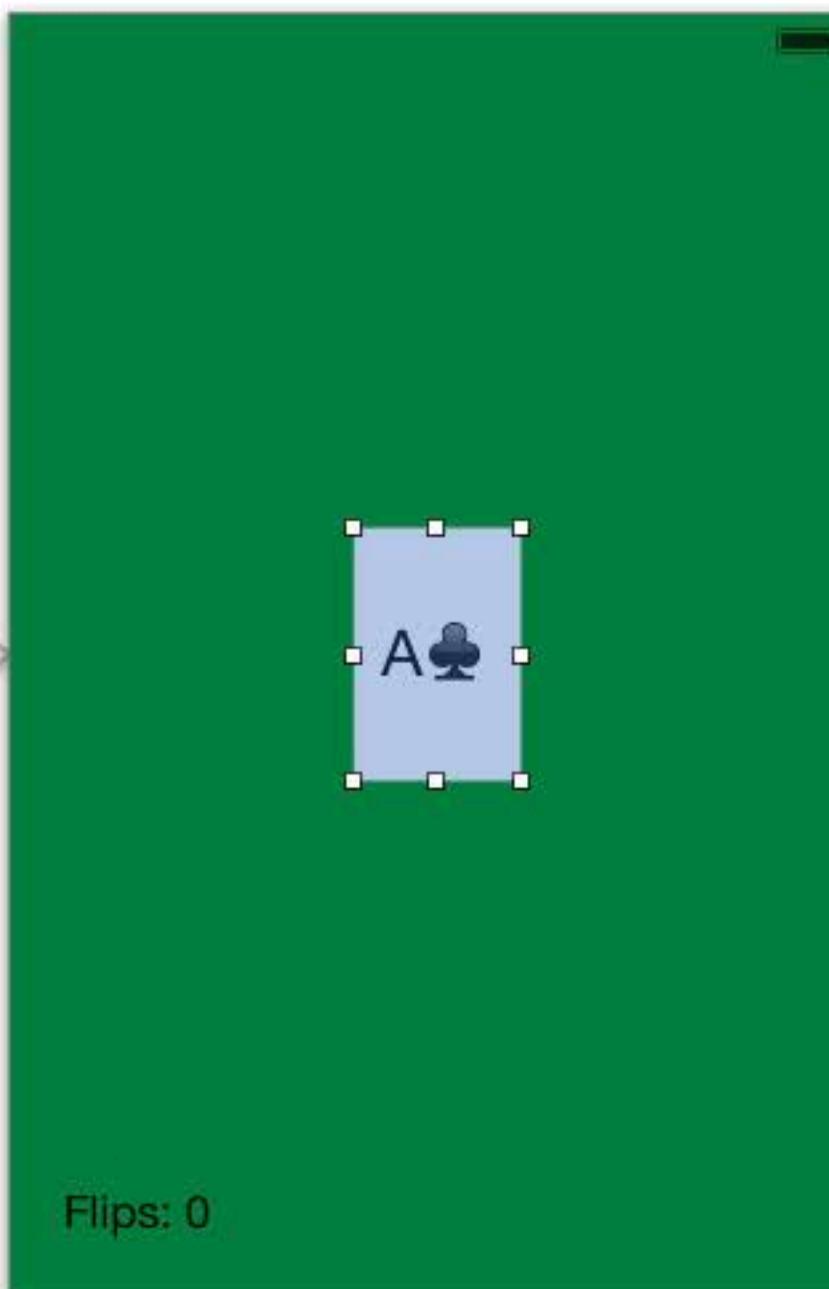
Vertical

Selected Enabled

Highlighted

Stanford CS193p

Fall 2013



```
//  
// 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;  
@end  
  
@implementation CardGameViewController  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
@end
```

Let's hide the Utilities area again to make more room for code.

Button

Type: System

State Config: Default

Title: Plain

A♦

Font: System 24.0

Text Color: Black Color

Shadow Color: Default

Image: Default Image

Background: cardfront

Shadow Offset: 0.0 0.0

Width: 0.0 Height: 0.0

Reverses On Highlight

Drawing: Shows Touch On Highlight Highlighted Adjusts Image Disabled Adjusts Image

Line Break: Truncate Middle

Edge: Content

Inset: 0 0 Top: 0 Bottom: 0

Left: 0 Right: 0

Control

Alignment: Horizontal

Vertical

Content: Selected Enabled

Highlighted

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m @interface CardGameViewController()

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

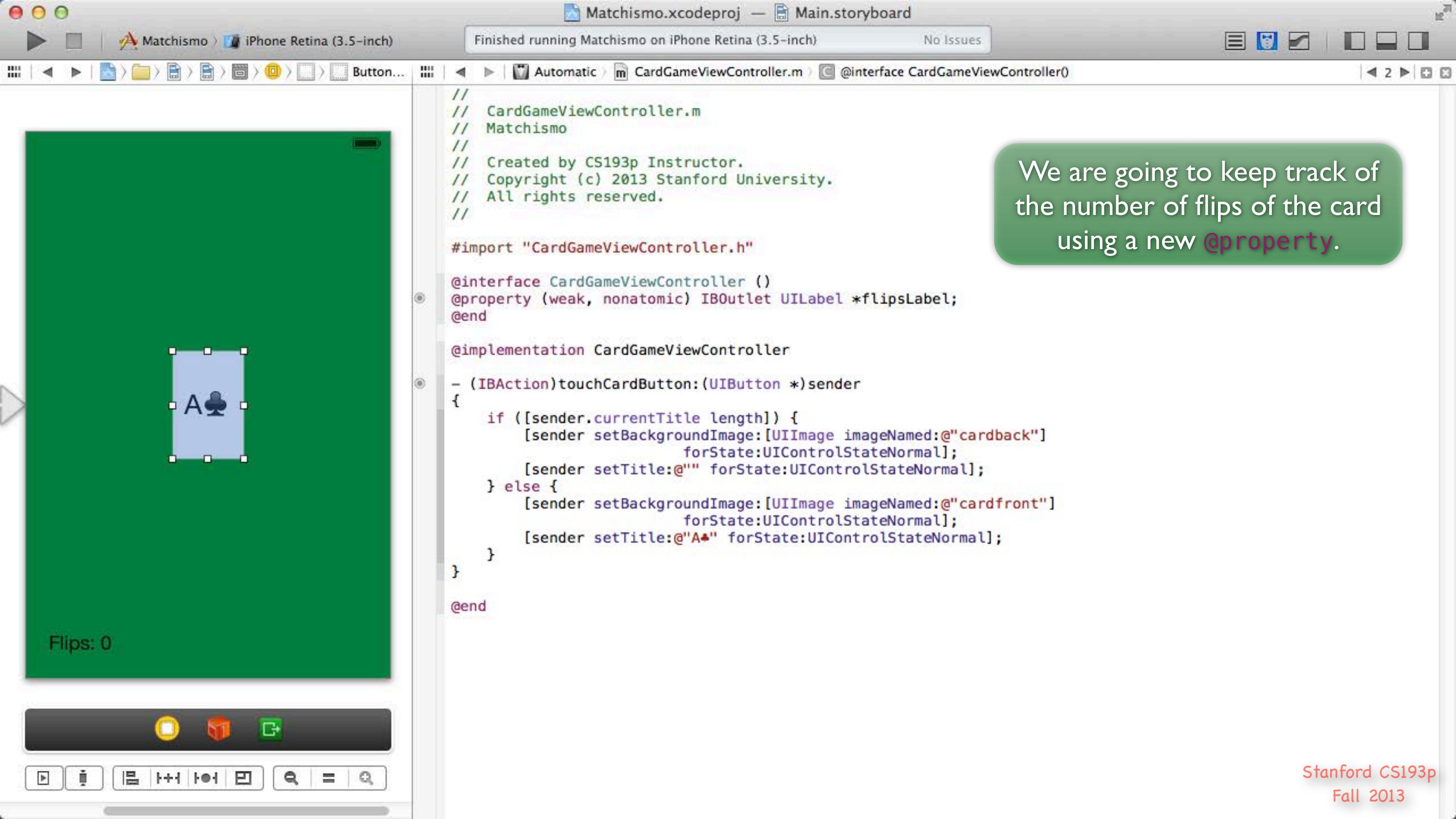
#import "CardGameViewController.h"

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

@implementation CardGameViewController

- (IBAction)touchCardButton:(UIButton *)sender
{
 if ([sender.currentTitle length]) {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
 forState:UIControlStateNormal];
 [sender setTitle:@"" forState:UIControlStateNormal];
 } else {
 [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]
 forState:UIControlStateNormal];
 [sender setTitle:@"A♣" forState:UIControlStateNormal];
 }
}
@end

We are going to keep track of the number of flips of the card using a new `@property`.



A screenshot of the Xcode IDE. The top navigation bar shows the project name 'Matchismo' and the target 'iPhone Retina (3.5-inch)'. The status bar indicates the application has finished running and shows 'No Issues'. The main area displays the 'CardGameViewController.m' file. The code implements a flip action for a button. A callout bubble on the right side highlights the first line of the @interface section, which defines a new property 'flipsLabel'. The storyboard preview on the left shows a single button labeled 'A♣' with a light blue background. The bottom of the screen shows the Xcode toolbar and a search bar.

Flips: 0

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

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;  
@end  
  
@implementation CardGameViewController  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♦" forState:UIControlStateNormal];  
    }  
}  
@end
```

We'll call this **@property** `flipCount`.

Nothing special about this **@property**, it's just an integer.
We could use `NSInteger` or `NSUInteger` here,
but we're using `int`, just to show doing so.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -touchCardButton:

```
//  
// 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  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
    self.flipCount++;  
}  
@end
```

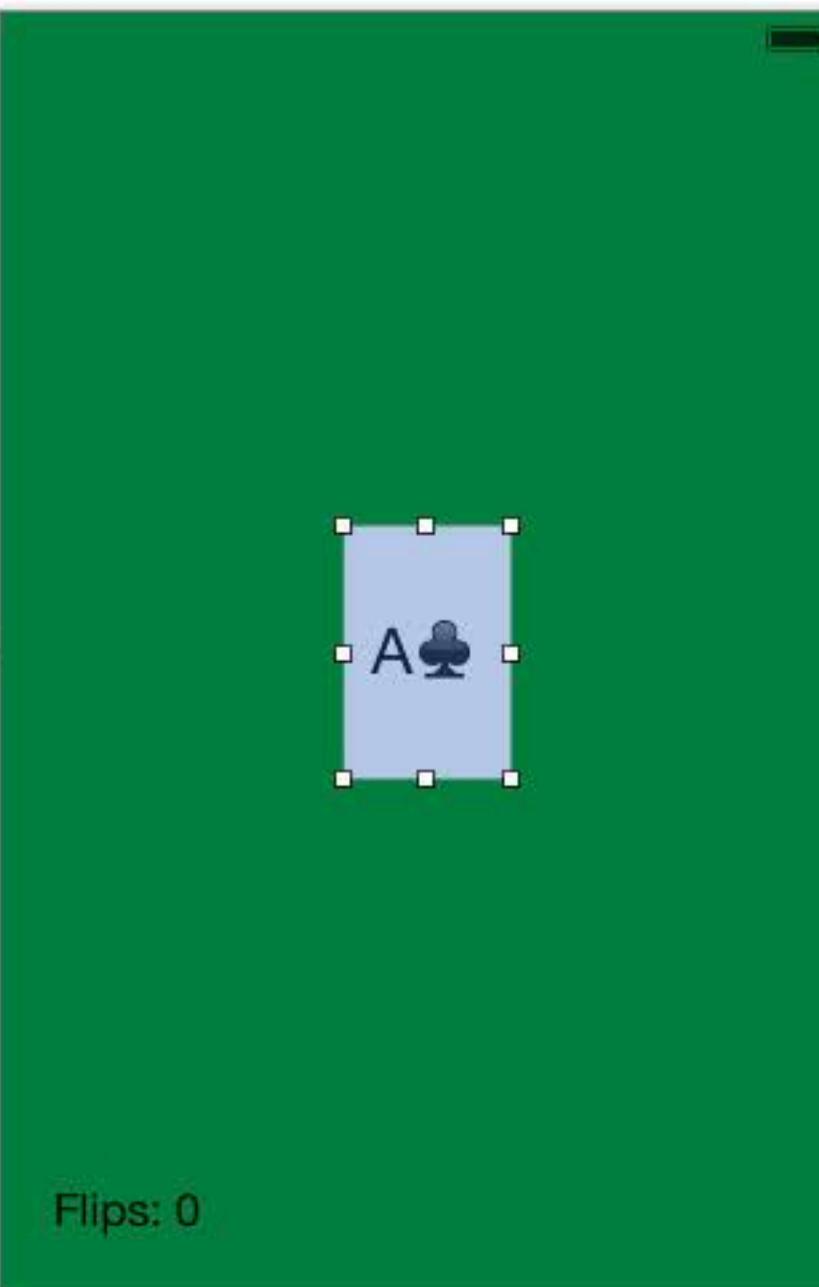
A screenshot of Xcode showing the Matchismo project. The storyboard preview on the left shows a card with 'A♣' on it. Below the preview is a label 'Flips: 0'. The code editor on the right shows the implementation of the touchCardButton method. A callout bubble points from the 'self.flipCount++' line to a explanatory text box.

And we'll just increment it each time we flip the card.

Notice that we can use `++` notation just like with a variable. This is the same as `self.flipCount = self.flipCount + 1`. In other words, `self.flipCount++` invokes both the getter and the setter for the `flipCount` `@property`.

Stanford CS193p Fall 2013

Matchismo iPhone Retina (3.5-inch)



Automatic CardGameViewController.m -setFlipCount:

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

But how can we coordinate the flipCount @property with the flipsLabel UILabel?

Easy! We'll use the setter of the flipCount @property.
This is yet another advantage of using @property instead of accessing instance variables directly.

```
#import "CardGameView.h"

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

@implementation CardGameViewController

- (void)setFlipCount:(int)flipCount
{
    _flipCount = flipCount;
}

- (IBAction)touchCardButton:(UIButton *)sender
{
    if ([sender.currentTitle length]) {
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]
                           forState:UIControlStateNormal];
        [sender setTitle:@"" forState:UIControlStateNormal];
    } else {
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]
                           forState:UIControlStateNormal];
        [sender setTitle:@"A♦" forState:UIControlStateNormal];
    }
    self.flipCount++;
}

@end
```

This is what the setter for flipCount would normally look like.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -setFlipCount:

```
//  
// 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)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length] == 0)  
        [sender setImage:[UIImage imageNamed:@"cardback"]  
            forState:UIControlStateNormal];  
    [sender setTitle:@"" forState:UIControlStateNormal];  
} else {  
    [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
        forState:UIControlStateNormal];  
    [sender setTitle:@"A♣" forState:UIControlStateNormal];  
}  
self.flipCount++;  
}  
@end
```

Flips: 0

A♣

We'll just add this one line of code to set the `text` `@property` of the `flipsLabel` to a string formatted to include the `flipCount`.

Now any time the `flipCount` `@property` changes, the `flipsLabel` `UILabel` will get updated.

Advanced thinking: note that we use `self.flipCount` here instead of just `_flipCount`. Imagine if a subclass wanted to control the value of `flipCount` by overriding the getter but still benefit from this method's display of it. Subtle.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -setFlipCount:

```
// 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];  
    NSLog(@"flipCount changed to %d", self.flipCount);  
}

- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A ♠" forState:UIControlStateNormal];  
    }  
    self.flipCount++;  
}
```

The first argument to `NSLog()` must always be an `@""`. Not any other kind of `NSString`.

Remember, the console will automatically appear at the bottom of the screen when you run.

While we're here, let's take another aside to look at a debugging technique. We can output something to the console any time we want.

We do it using the C function `NSLog()`. The first argument to `NSLog()` is a `@""` printf-like format string specifying what to output. The rest of the arguments are the values matching up with the `%`'s in the format string.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Main.storyboard

Running Matchismo on iPhone Retina (3.5-inch)

No Issues

Carrier

Okay, let's run again!

```
//  
// 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", flipCount];  
    NSLog(@"flipCount changed to %d", self.flipCount);  
}  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length] == 0){  
        [sender setBackgroundImage:[UIImage imageNamed:@"card-back.png"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"card-front.png"]  
                           forState:UIControlStateNormal];  
    }  
}
```

Flips: 0

Flips: 0

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Main.storyboard

Running Matchismo on iPhone Retina (3.5-inch)

No Issues

Matchismo > iPhone Retina (3.5-inch) | Button... | Automatic | CardGameViewController.m | -setFlipCount:

```
//  
// 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", flipCount];  
    NSLog(@"flipCount changed to %d", self.flipCount);  
}  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"card-flipped.png"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"card-normal.png"]  
            forState:UIControlStateNormal];  
        [sender setTitle:@"A ♣" forState:UIControlStateNormal];  
    }  
}
```

Carrier

Flips: 0

Flips: 1

A ♣

Touch

This should change.

And here's the output of the NSLog().

2013-00-00 14:29:14.411 Matchismo[72114:a0b] flipCount changed to 1

Matchismo.xcodeproj — Main.storyboard

Running Matchismo on iPhone Retina (3.5-inch)

No Issues

Matchismo > iPhone Retina (3.5-inch) | Button... | Automatic | CardGameViewController.m | -setFlipCount:

```
//  
// 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", flipCount];  
    NSLog(@"flipCount changed to %d", self.flipCount);  
}  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"card-flipped.png"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"card-normal.png"]  
                           forState:UIControlStateNormal];  
        [sender setTitle:@"A ♠" forState:UIControlStateNormal];  
    }  
}
```

Carrier

Touch again.

Changes again.

Flips: 0

A ♠

Flips: 2

2013-00-00 14:29:14.411 Matchismo[72114:a0b] flipCount changed to 1
2013-00-00 14:29:14.834 Matchismo[72114:a0b] flipCount changed to 2

And outputs here again.

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Automatic CardGameViewController.m -setFlipCount:

Stop.

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

Well this is all wonderful, but it's sort of boring since it only shows the A♣ all the time.

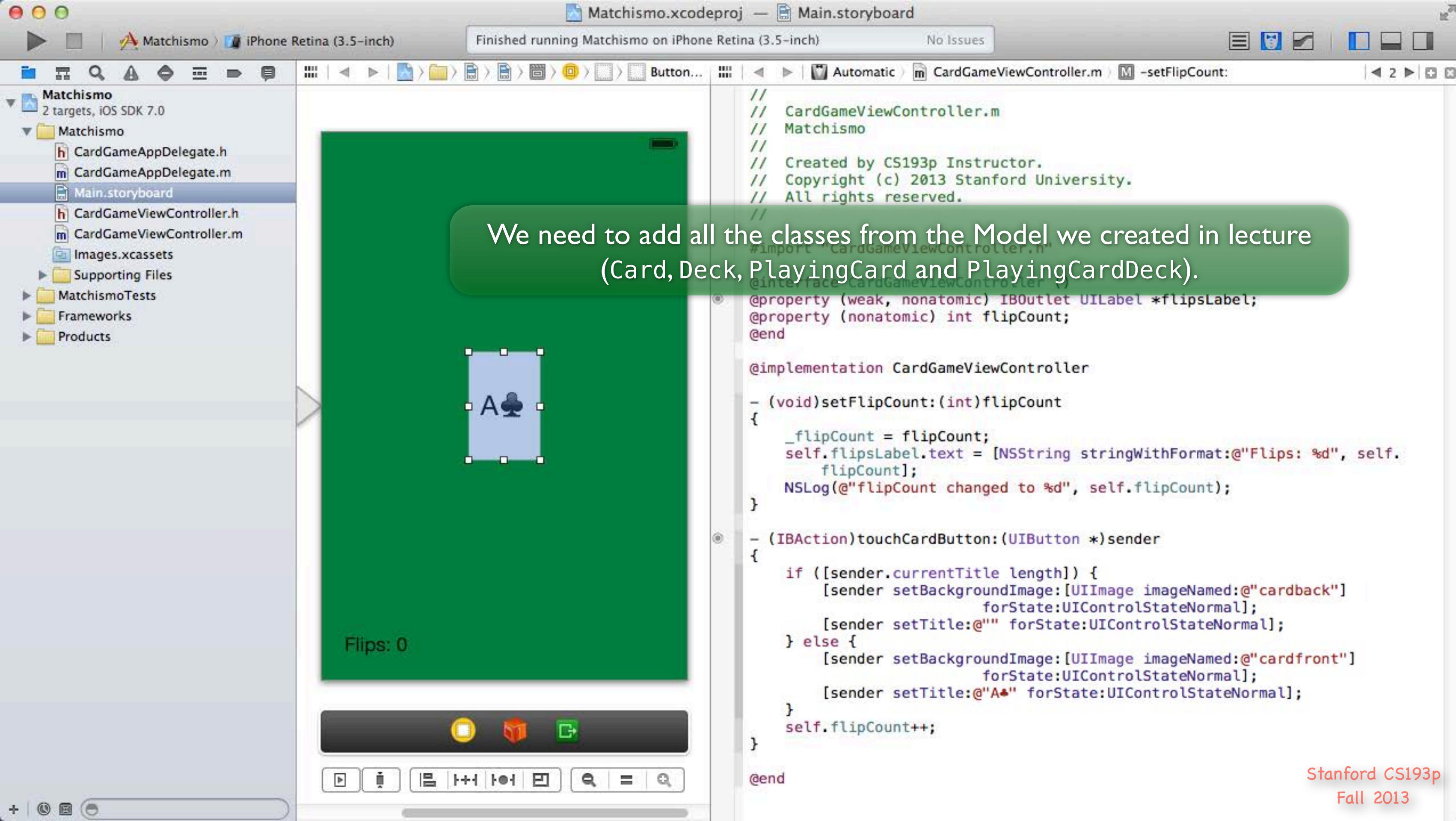
If only we had a Deck of PlayingCards to drawRandomCard from, we could make each flip show a different card.

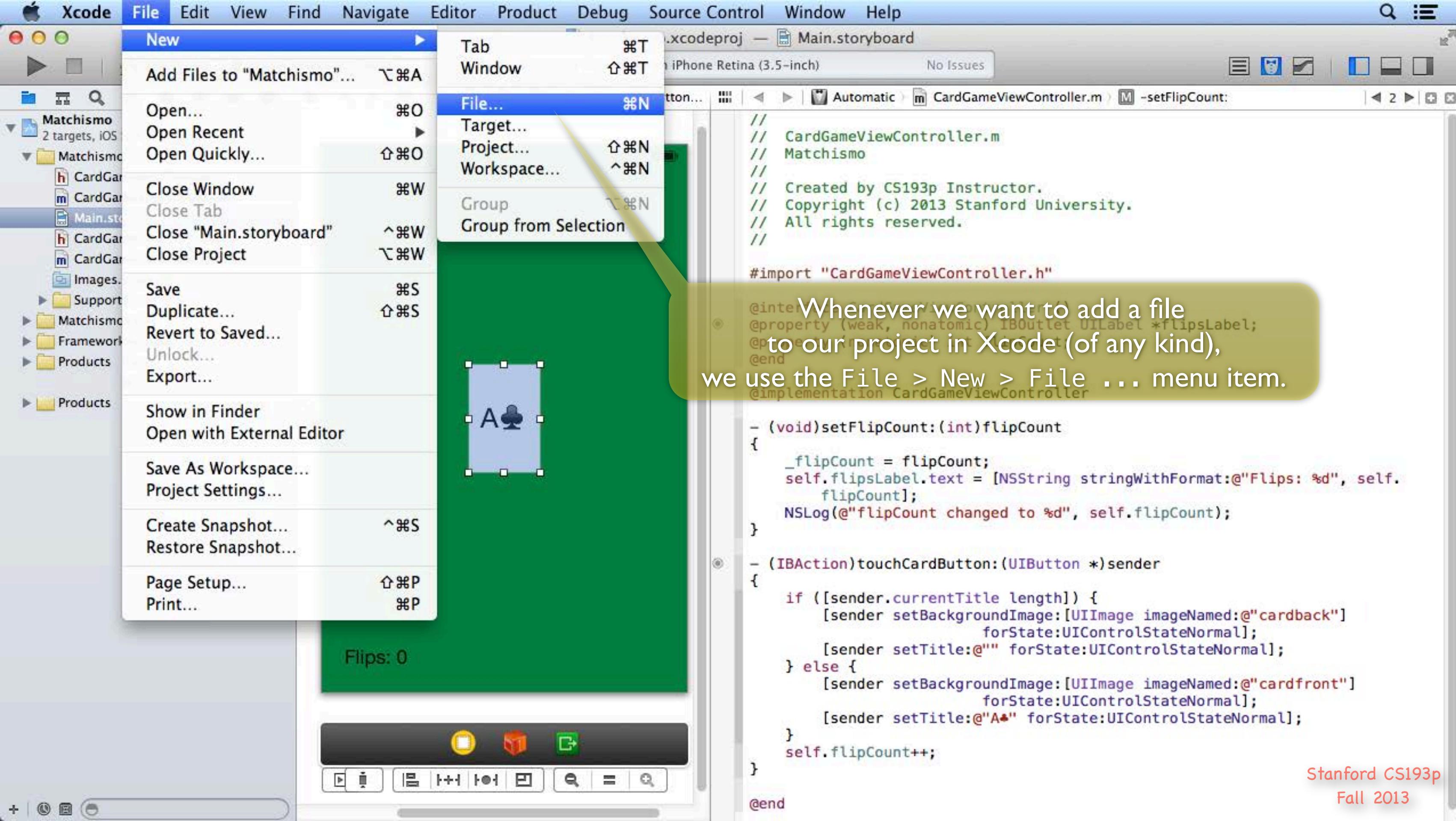
Hmmm ...

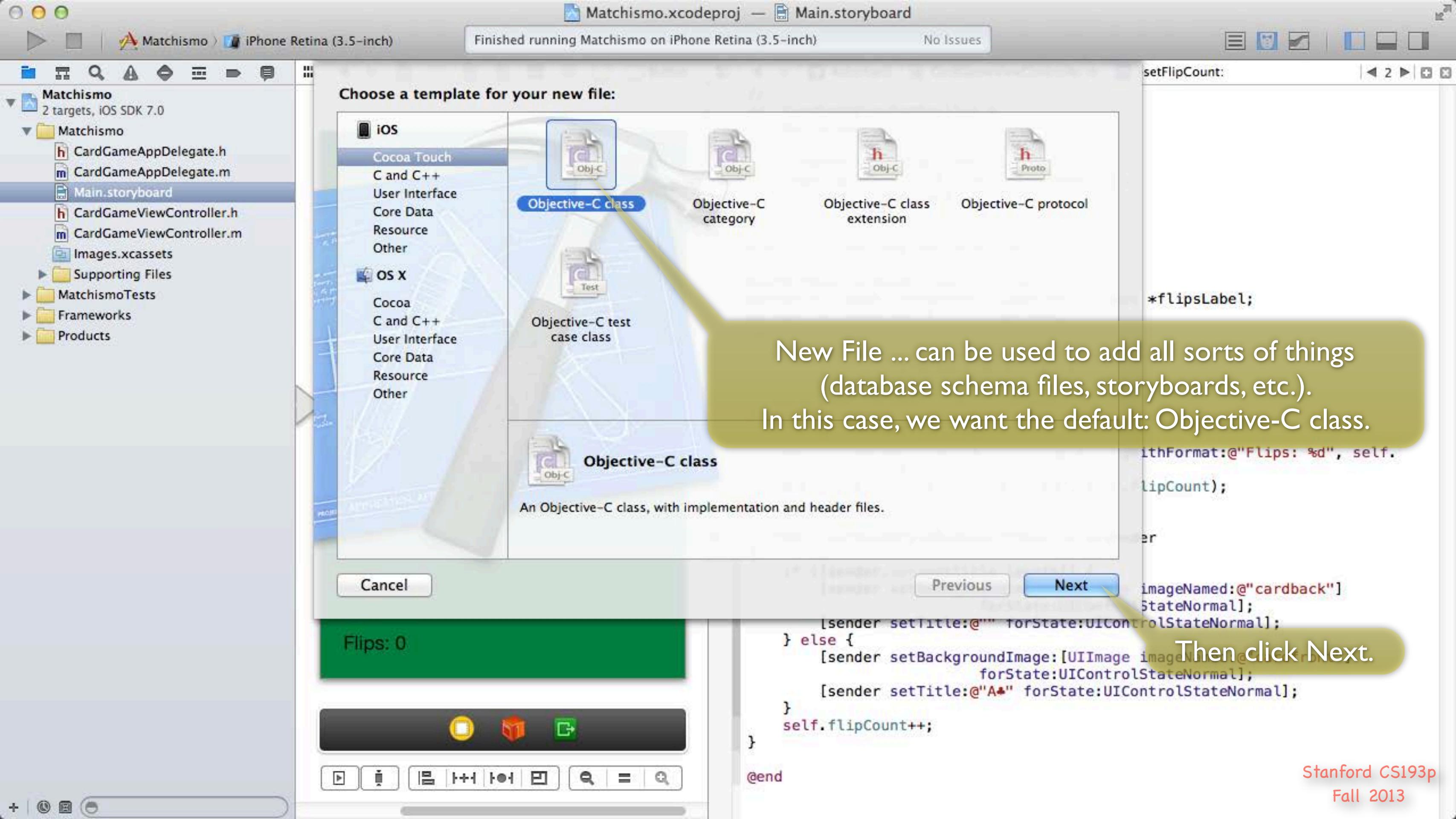
Let's start by revealing the Navigator again.

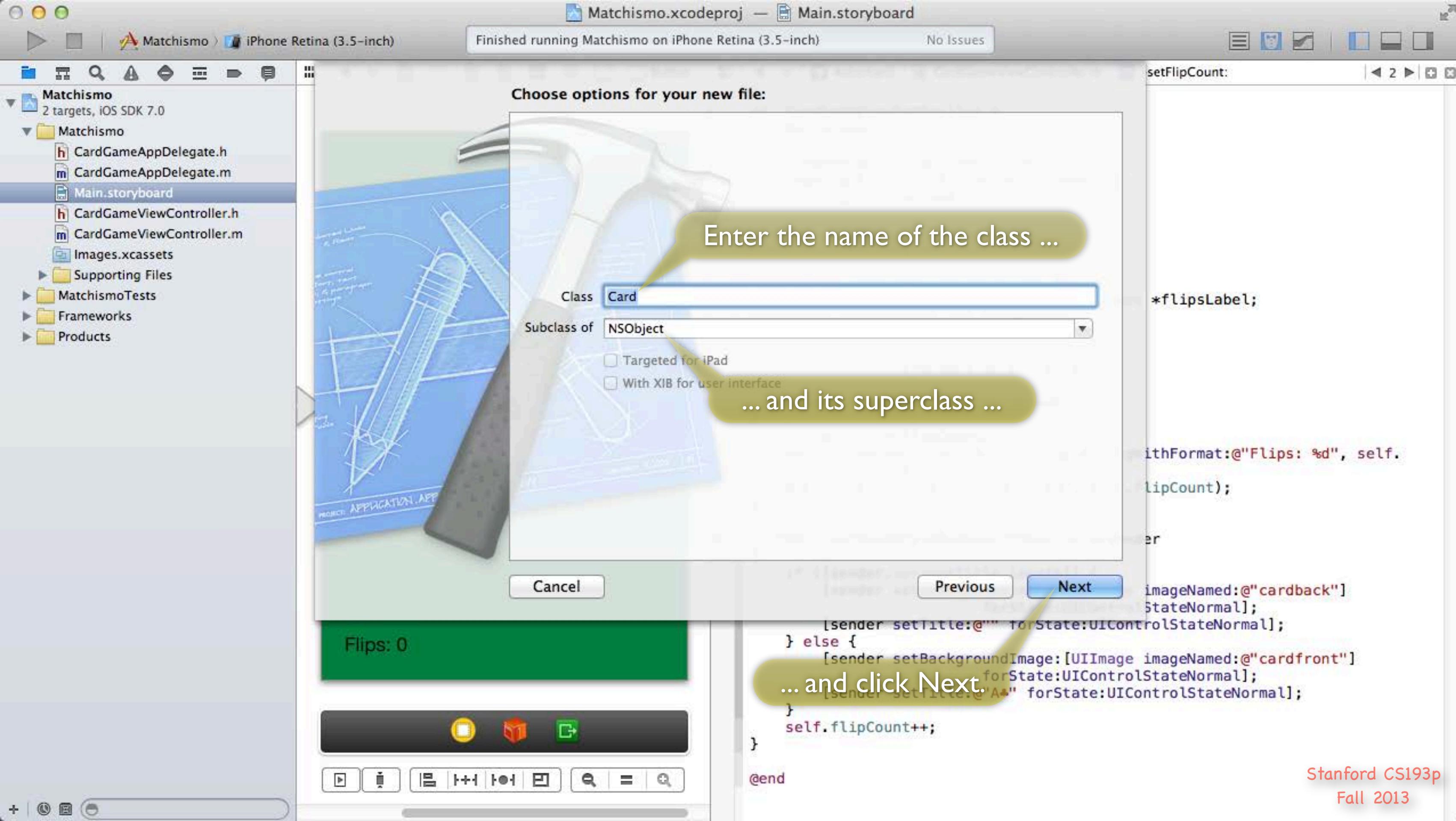
```
#import "CardGameViewController.h"  
  
@interface CardGameViewController()  
@property (weak, nonatomic) IBOutlet UILabel *flipsLabel;  
@property (nonatomic) int flipCount;  
@end  
  
@implementation CardGameViewController  
  
- (void)setFlipCount:(int)flipCount  
{  
    _flipCount = flipCount;  
    self.flipsLabel.text = [NSString stringWithFormat:@"Flips: %d", self.flipCount];  
    NSLog(@"flipCount changed to %d", self.flipCount);  
}  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{  
    if ([sender.currentTitle length]) {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"" forState:UIControlStateNormal];  
    } else {  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardfront"]  
                             forState:UIControlStateNormal];  
        [sender setTitle:@"A♣" forState:UIControlStateNormal];  
    }  
    self.flipCount++;  
}  
  
@end
```

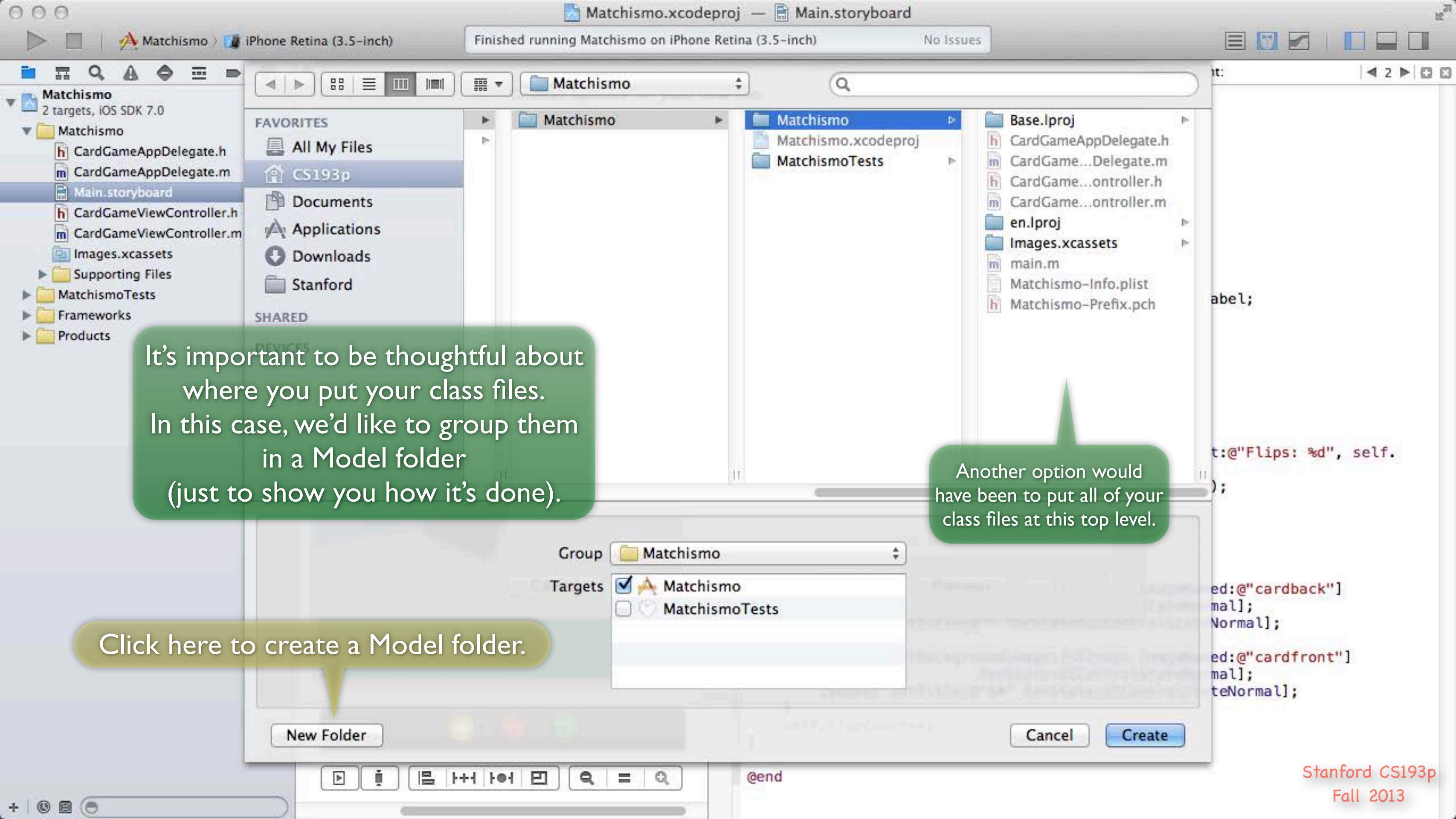
A screenshot of Xcode showing the Matchismo project. The storyboard preview on the left shows a green screen with a blue button labeled 'A♣' and a label below it saying 'Flips: 0'. The code editor on the right shows the implementation of CardGameViewController.m. A yellow callout bubble points to the 'Stop.' button in the storyboard with the text 'Well this is all wonderful, but it's sort of boring since it only shows the A♣ all the time.' Another yellow callout bubble points to the code with the text 'If only we had a Deck of PlayingCards to drawRandomCard from, we could make each flip show a different card.' A third yellow callout bubble points to the 'Hmmm ...' text with the text 'Let's start by revealing the Navigator again.' The status bar at the top says 'Matchismo' and 'iPhone Retina (3.5-inch)'.

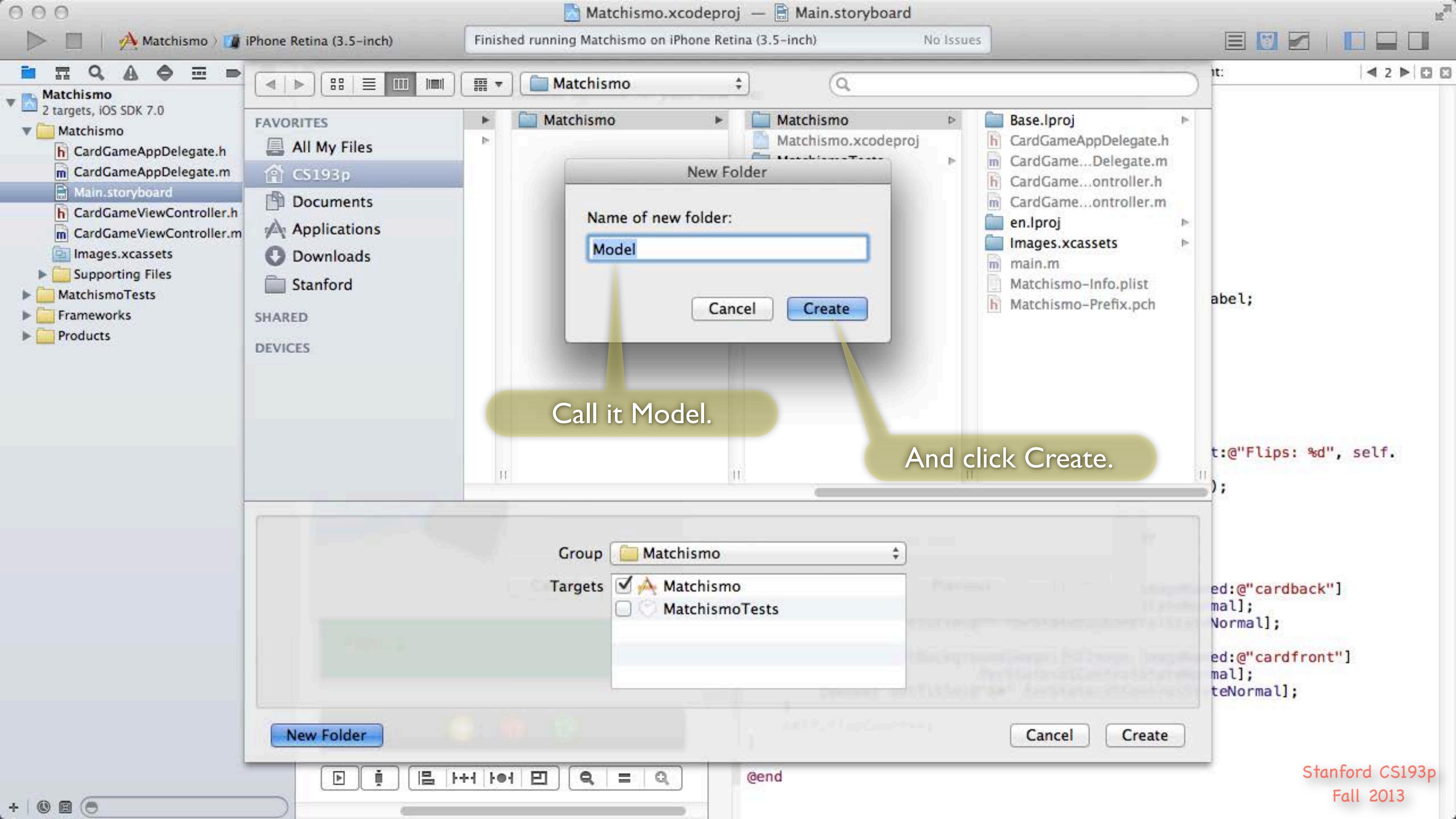


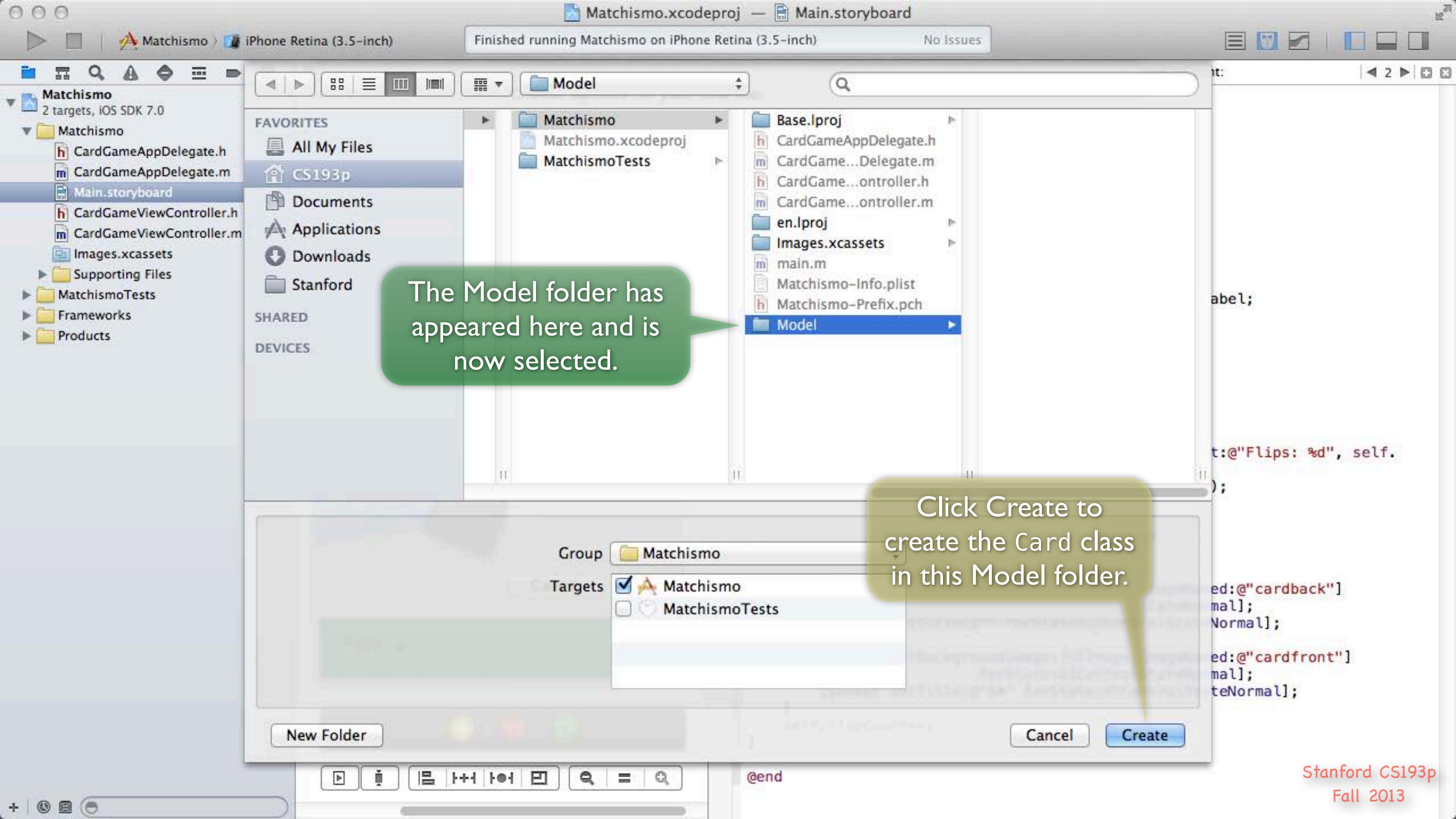












Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

Card.h

Card.m

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end
```

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

#import "Card.h"

@implementation Card

@end
```

Here is a (blank) Card class.
You will have to go through the slides from earlier and type in the implementation of Card.

It is important to *type the code in* (not copy/paste it from somewhere)
so that you gain experience with entering code in Xcode.

Stanford CS193p
Fall 2013

You can put the header file in either the left or right side of the Assistant Editor as you prefer.

You can choose which goes where by clicking on the name of the file at the top of the pane in question.

```
//  
// Card.h  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import <Foundation/Foundation.h>  
  
@interface Card : NSObject  
@end
```

```
//  
// Card.m  
// Matchismo  
// Created by CS193p Instructor.  
// Copyright (c) 2013 Stanford University.  
// All rights reserved.  
  
#import "Card.h"  
  
@implementation Card  
  
@end
```

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch)

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

Card.h

Card.m

CardGameViewController.h

CardGameViewController.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end

Manual

Counterparts (1)

Superclasses

Subclasses

Siblings

Categories

Protocols

User Interfaces

Includes

Included By

Callers

Callees

Test Classes

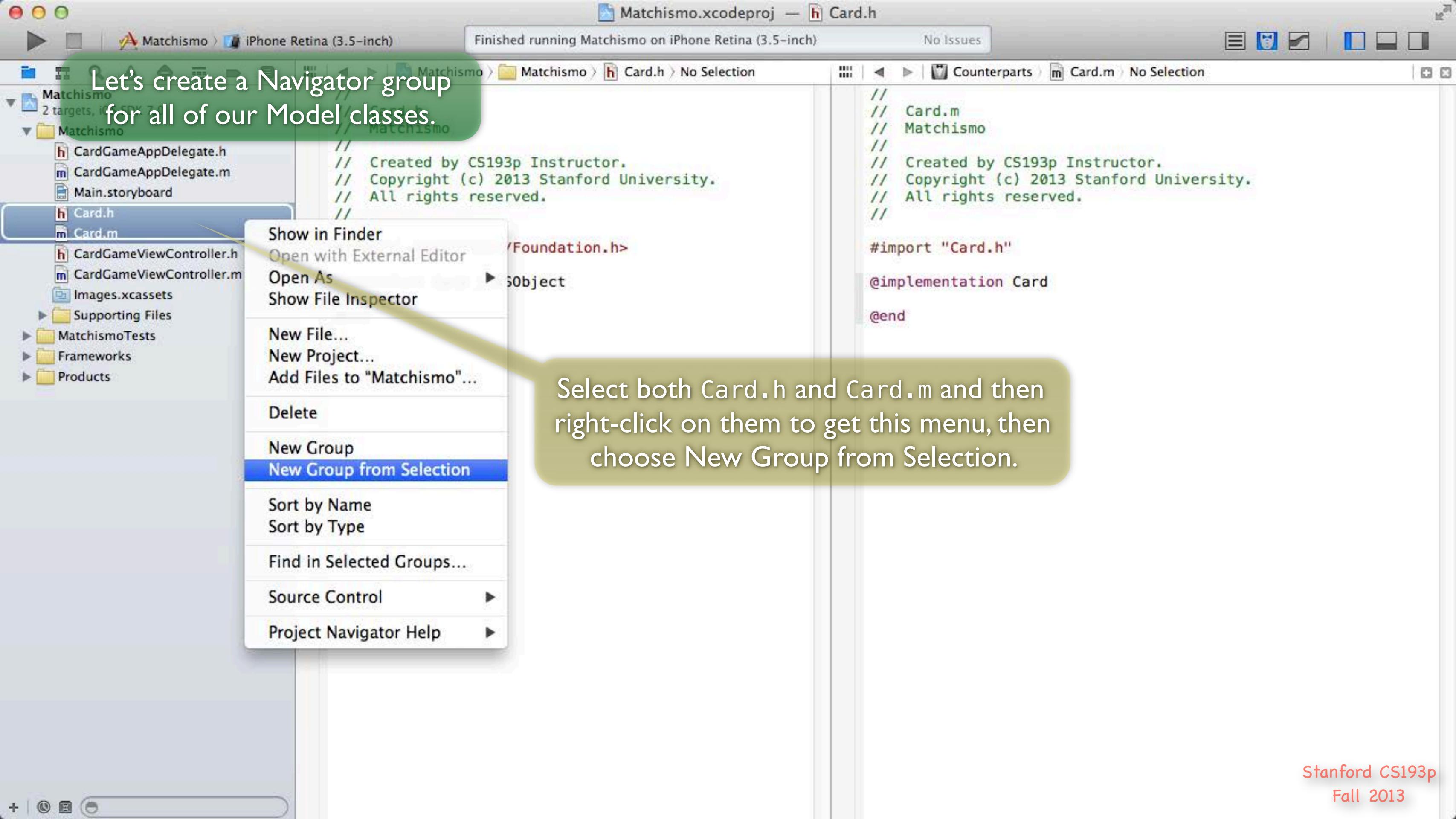
Test Callers

Preprocess

Choosing the header or implementation from the Counterparts menu at the top of the pane ensures that Xcode will continue to always match the implementation up with the interface (or vice versa), even if you change what is in the left pane.

Stanford CS193p

Fall 2013



Let's create a Navigator group for all of our Model classes.

Select both Card.h and Card.m and then right-click on them to get this menu, then choose New Group from Selection.

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- New Group
 - Card.h
 - Card.m
 - CardGameViewController.h
 - CardGameViewController.m
 - Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end

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

#import "Card.h"

@implementation Card

@end

And we'll rename the group ...

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard

Model

- Card.h
- Card.m
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end
```

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

#import "Card.h"

@implementation Card

@end
```

... to Model.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- Model**
- Card.h
- Card.m
- CardGameViewController.h
- CardGameViewController.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject
@end

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

#import "Card.h"

@implementation Card
@end
```

A group in the File Navigator can be linked to a directory in the filesystem or not, as you prefer. You control this from the File Inspector in the Utilities area.

Identity and Type

- Name Card.h
- Type Default C Header
- Absolute Path
- Relative to Group**
- Relative to Project
- Relative to Developer Directory
- Relative to Build Products
- Relative to SDK

Target Membership

- Matchismo
- MatchismoTests

Text Settings

- Text Encoding Default – Unicode (UTF-8)
- Line Endings Default – OS X / Unix (LF)
- Indent Using Spaces
- Widths Tab 4 Indent 4
- Wrap lines

Source Control

- Repository --
- Type --
- Current Branch --
- Version --
- Status No changes
- Location

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

CardGameAppDelegate.h

CardGameAppDelegate.m

Main.storyboard

Model

Card.h

Card.m

CardGameViewController.h

CardGameViewController.m

Model

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end
```

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

#import "Card.h"

@implementation Card

@end
```

You can drag things around in the File Navigator to put them in whatever order you want.

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m

Model

- Card.h
- Card.m
- Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end

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

#import "Card.h"

@implementation Card

@end
```

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Matchismo

- CardGameAppDelegate.h
- CardGameAppDelegate.m
- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m

Model

- Card.h
- Card.m

Images.xcassets

CardGameAppDelegate.h

Supporting Files

- CardGameAppDelegate.plist
- Matchismo-Info.plist
- InfoPlist.strings
- main.m
- Matchismo-Prefix.pch

MatchismoTests

Frameworks

Products

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

#import "Card.h"

@interface Card : NSObject

@end
```

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

#import "Card.h"

@implementation Card

@end
```

For example, often we'll drag the AppDelegate.[mh] into Supporting Files group since we rarely edit them.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Main.storyboard CardGameViewController.h CardGameViewController.m Model Card.h Card.m Images.xcassets Supporting Files CardGameAppDelegate.h CardGameAppDelegate.m Matchismo-Info.plist InfoPlist.strings main.m Matchismo-Prefix.pch

Card.h

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end
```

Card.m

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

#import "Card.h"

@implementation Card

@end
```

If you explicitly click on Card.h in the File Navigator, it will show it in the left pane of the Assistant Editor.

And Card.m will automatically appear in the right pane as long as Counterparts is selected here.

Stanford CS193p
Fall 2013

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Main.storyboard CardGameViewController.h CardGameViewController.m Model Card.h Card.m Images.xcassets Supporting Files MatchismoTests Frameworks Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@end

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

#import "Card.h"

@implementation Card

@end

Close Supporting Files folder.

Stanford CS193p Fall 2013

Matchismo.xcodeproj — Card.h

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo 2 targets, iOS SDK 7.0

Main.storyboard CardGameViewController.h CardGameViewController.m Model Card.h Card.m Images.xcassets Supporting Files MatchismoTests Frameworks Products

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

#import <Foundation/Foundation.h>

@interface Card : NSObject

@property (strong, nonatomic) NSString *contents;
@property (nonatomic, getter=isChosen) BOOL chosen;
@property (nonatomic, getter=isMatched) BOOL matched;

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

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

#import "Card.h"

@implementation Card

- (int)match:(NSArray *)otherCards
{
    int score = 0;

    for (Card *card in otherCards) {
        if ([card.contents isEqualToString:self.contents]) {
            score = 1;
        }
    }

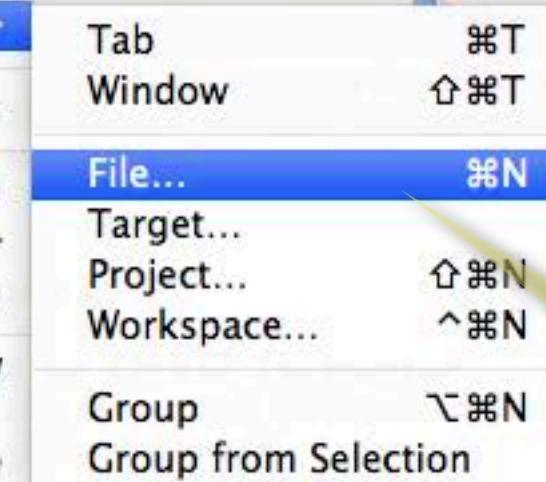
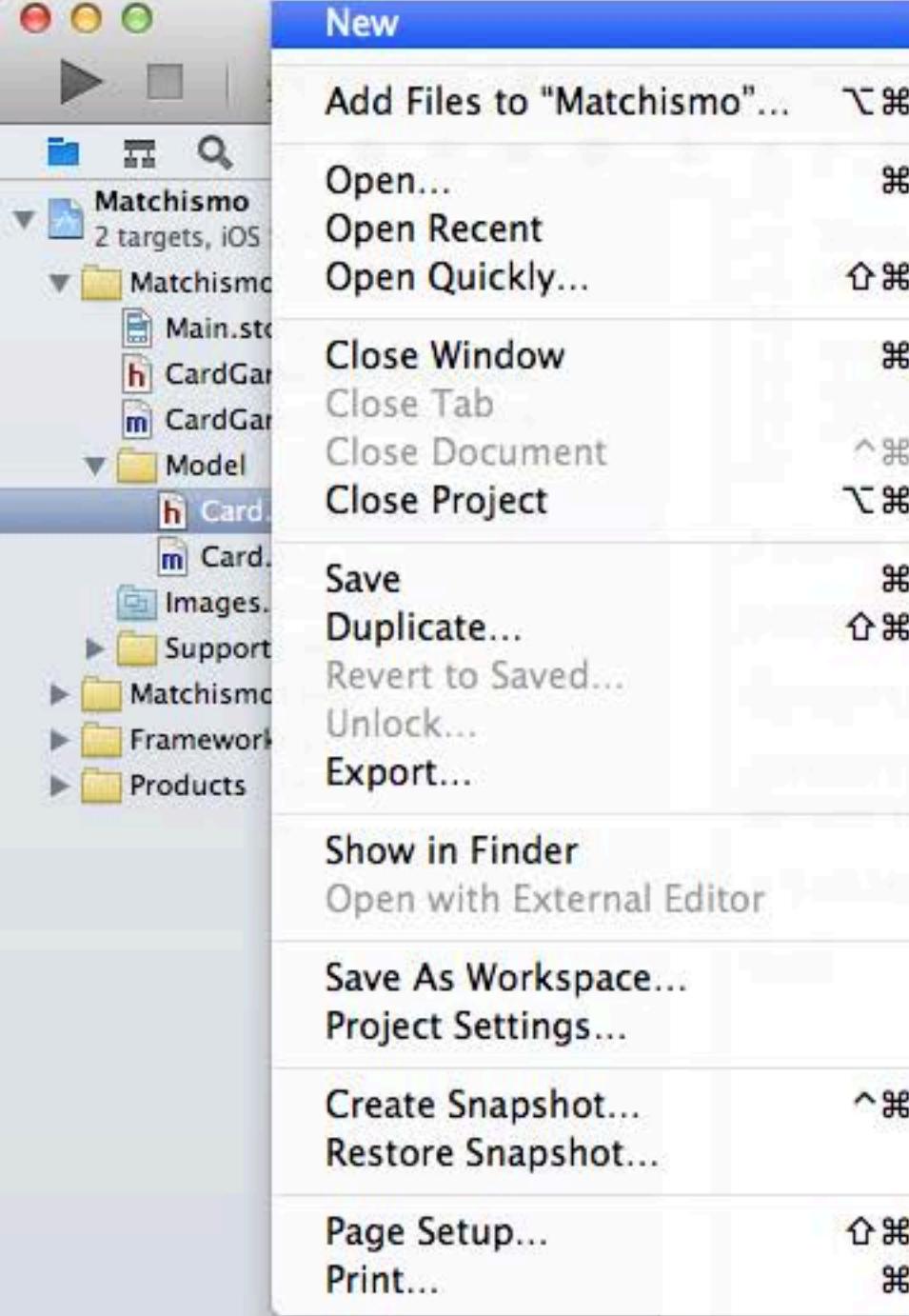
    return score;
}

@end
```

Type in the code for Card. [mh].

Now let's move on to creating templates for the Deck, PlayingCard and PlayingCardDeck classes.

Stanford CS193p
Fall 2013



```
ismo.xcodeproj — Card.h
iPhone Retina (3.5-inch) No Issues
Card.h @interface Card : NSObject
{
    NSString *contents;
    BOOL chosen;
    BOOL matched;
}

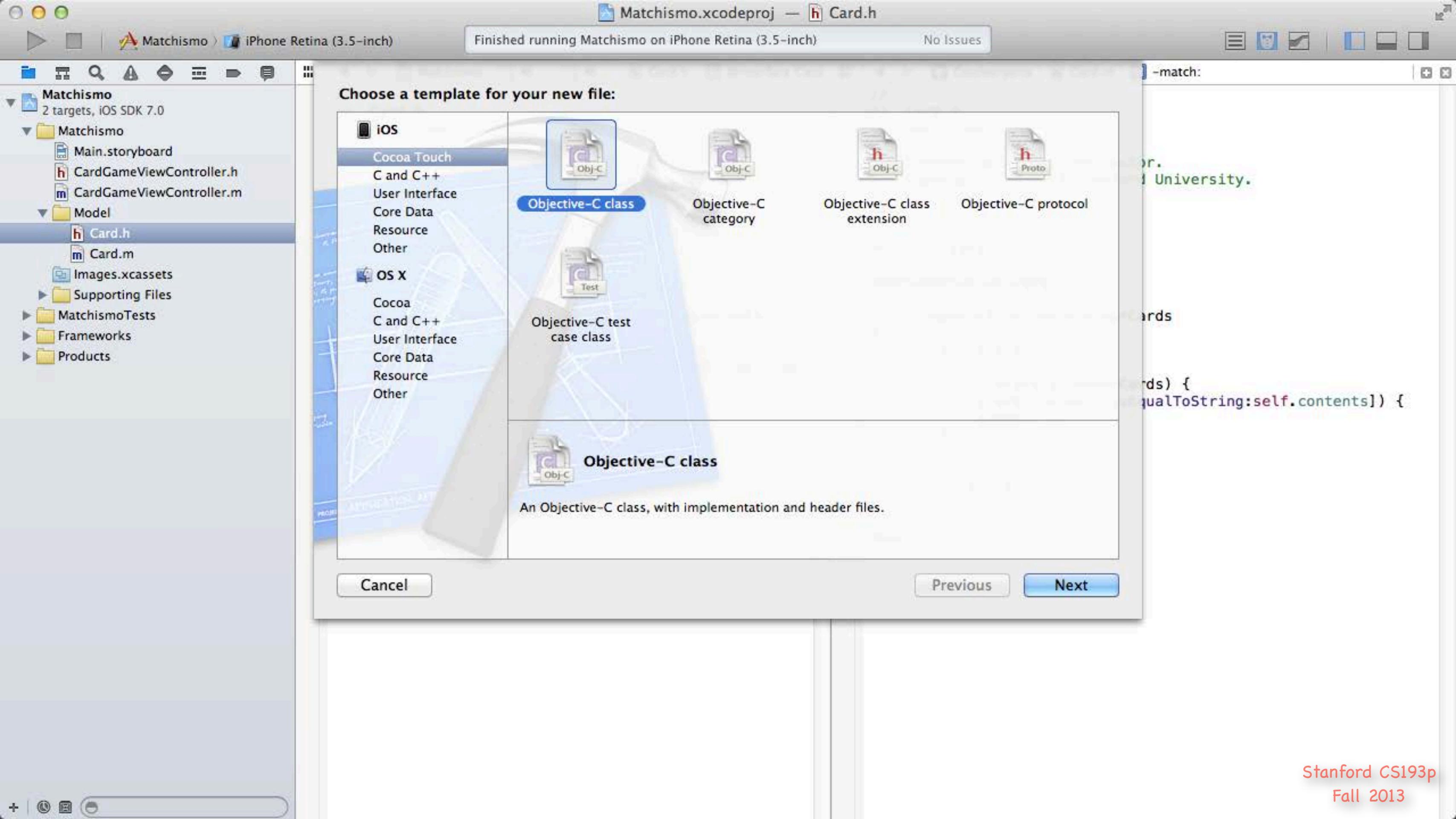
- (int)match:(NSArray *)otherCards;
{
    int score = 0;

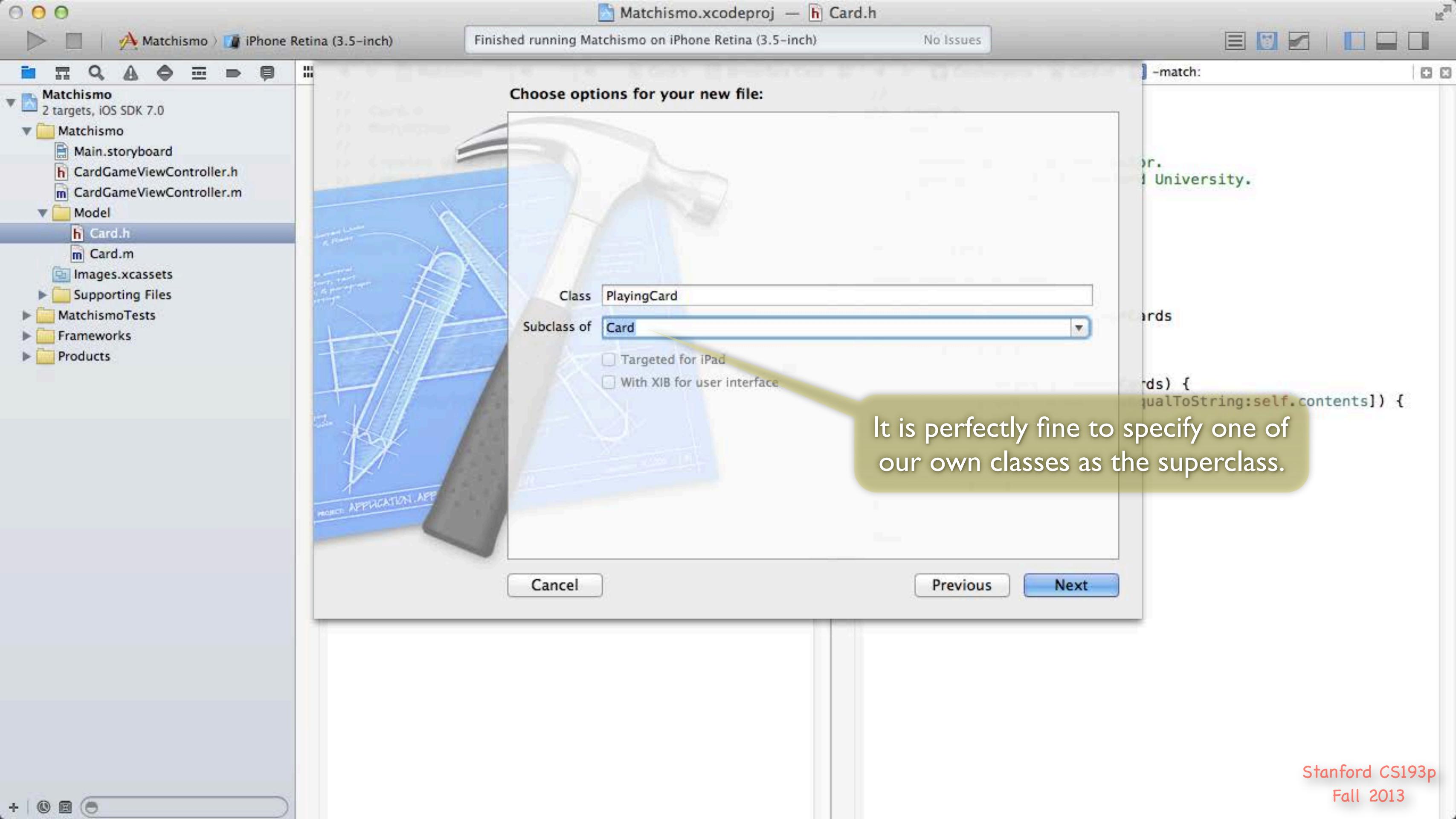
    for (Card *card in otherCards) {
        if ([card.contents isEqualToString:self.contents]) {
            score = 1;
        }
    }

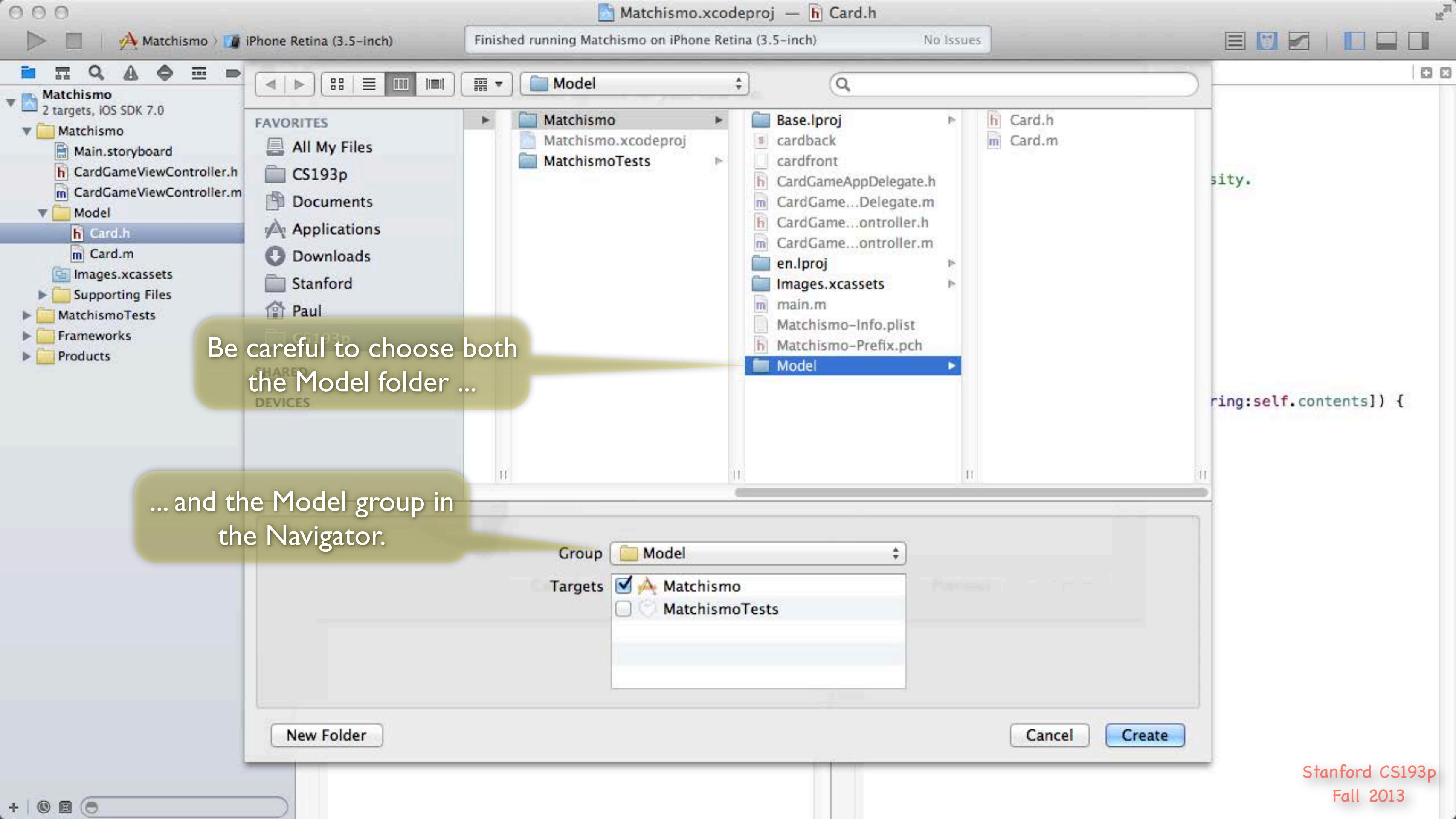
    return score;
}

@end
```

File > New > File ... again for PlayingCard.







Matchismo
2 targets, iOS SDK 7.0

Matchismo

- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Model
 - Card.h
 - Card.m
 - PlayingCard.h
 - PlayingCard.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

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

#import "Card.h"

@interface PlayingCard : Card

@end
```

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

#import "PlayingCard.h"

@implementation PlayingCard

@end
```

Hopefully PlayingCard. [mh] appeared
in your Model group!

Drag it in if not.

Matchismo
2 targets, iOS SDK 7.0

Matchismo
Main.storyboard
CardGameViewController.h
CardGameViewController.m
Model
Card.h
Card.m
PlayingCard.h
PlayingCard.m
Images.xcassets
Supporting Files
MatchismoTests
Frameworks
Products

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

#import "PlayingCard.h"

@implementation PlayingCard

- (NSString *)contents
{
    NSArray *rankStrings = [PlayingCard rankStrings];
    return [rankStrings[self.rank] stringByAppendingString:suit];
}

@synthesize suit = _suit;

+ (NSArray *)validSuits
{
    return @[@"♠", @"♦", @"♥", @"♦"];
}

- (void)setSuit:(NSString *)suit
{
    if ([[PlayingCard validSuits] containsObject:suit]) {
        _suit = suit;
    }
}

- (NSString *)suit
{
    return _suit ? _suit : @"?";
}

+ (NSArray *)rankStrings
{
    return @[@"?", @"A", @"2", @"3", @"4", @"5", @"6",
             @"7", @"8", @"9", @"10", @"J", @"Q", @"K"];
}
```

Type in the code for PlayingCard. [mh].

All the code doesn't fit here, so use the other lecture slides to enter this code.

Matchismo.xcodeproj — Deck.h

Matchismo > iPhone Retina (3.5-inch)

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Matchismo

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Model

Card.h

Card.m

PlayingCard.h

PlayingCard.m

Deck.h

Deck.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

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

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

@interface Deck : NSObject

- (void)addCard:(Card *)card atTop:(BOOL)atTop;
- (void)addCard:(Card *)card;

- (Card *)drawRandomCard;

@end
```

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

#import "Deck.h"

@interface Deck()
@property (strong, nonatomic) NSMutableArray *cards; // of Card
@end

@implementation Deck

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

- (void)addCard:(Card *)card atTop:(BOOL)atTop
{
    if (atTop) {
        [self.cards insertObject:card atIndex:0];
    } else {
        [self.cards addObject:card];
    }
}

- (void)addCard:(Card *)card
{...}

- (Card *)drawRandomCard
{
    Card *randomCard = nil;

    if ([self.cards count]) {
        unsigned index = arc4random() % [self.cards count];
        randomCard = self.cards[index];
        [self.cards removeObjectAtIndex:index];
    }
}
```

All the code doesn't fit here, so use the other lecture slides to enter this code.

Repeat for Deck. [mh].

Stanford CS193p
Fall 2013

Matchismo
2 targets, iOS SDK 7.0

Matchismo

- Main.storyboard
- CardGameViewController.h
- CardGameViewController.m
- Model
 - Card.h
 - Card.m
 - PlayingCard.h
 - PlayingCard.m
 - Deck.h
 - Deck.m
 - PlayingCardDeck.h
 - PlayingCardDeck.m
- Images.xcassets
- Supporting Files
- MatchismoTests
- Frameworks
- Products

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

#import "Deck.h"

@interface PlayingCardDeck : Deck

@end
```

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

//
//import "PlayingCardDeck.h"
//import "PlayingCard.h"

@implementation PlayingCardDeck

- (instancetype)init
{
    self = [super init];

    if (self) {
        for (NSString *suit in [PlayingCard validSuits]) {
            for (NSUInteger rank = 1; rank <= [PlayingCard maxRank]; rank++) {
                PlayingCard *card = [[PlayingCard alloc] init];
                card.rank = rank;
                card.suit = suit;
                [self addCard:card];
            }
        }
    }
    return self;
}

@end
```

Repeat for PlayingCardDeck. [mh].

Matchismo.xcodeproj — Main.storyboard

Finished running Matchismo on iPhone Retina (3.5-inch) No Issues

Main.storyboard Card Game View Controller View Automatic CardGameViewController.m No Selection

Matchismo 2 targets, iOS SDK 7.0

Matchismo

Main.storyboard

CardGameViewController.h

CardGameViewController.m

Model

Card.h

Card.m

PlayingCard.h

PlayingCard.m

Deck.h

Deck.m

PlayingCardDeck.h

PlayingCardDeck.m

Images.xcassets

Supporting Files

MatchismoTests

Frameworks

Products

Click here to go back to the View.

A ♣

Flips: 0

We did all this so that we could have each card not be A ♣.
Your homework is to make each flip draw a new random card.
One of the first things you'll want is a `@property` for a Deck.

Good luck!

```
//  
// 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];  
    NSLog(@"flipCount changed to %d", self.flipCount);  
}  
  
- (IBAction)touchCardButton:(UIButton *)sender  
{ Whew!  
    if ([sender.currentTitle length]) {  
        [sender setTitle:@"" forState:UIControlStateNormal];  
        [sender setBackgroundImage:[UIImage imageNamed:@"cardback"]  
                           forState:UIControlStateNormal];  
        [sender setTitleColor:[UIColor blackColor] forState:UIControlStateNormal];  
    }  
    self.flipCount++;  
}  
@end
```

Stanford CS193p
Fall 2013