House Course 59-20

Web and Mobile Applications
Week 2: Xcode Intro / ToDo App

Davis Gossage
Jesse Hu

Week 2

- Swift Playground Review
- The Xcode Environment
- A Command Line App
- A To-do List App

Updates

- Parse.com...

An Important Announcement.

Parse's hosted services will be fully retired on January 28, 2017. We're proud that we've been able to help so many of you build great mobile apps. Read more on this announcement and what this means for your app here. Thank you for using Parse.

What it means

- Parse is shutting down its platform next year :-(
  - “need to focus our resources elsewhere”
- Still an awesome platform, and they open sourced it!
  - Means we get to learn more about MongoDB and Node.js which the server runs on top of

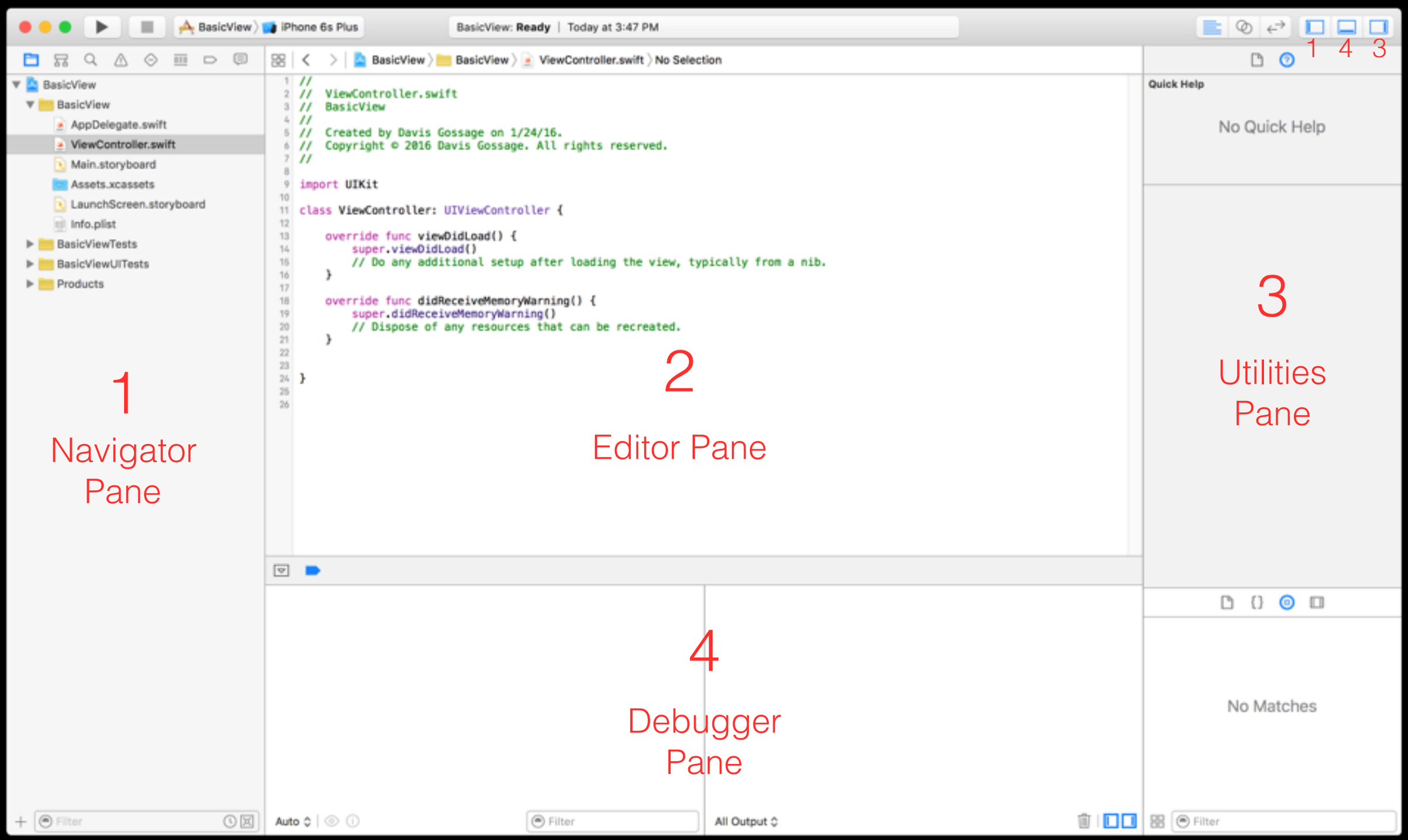
Updates - Swift

- ++ and -- operators will be removed from Swift 3.0
- Why?
  - It is shorthand for i += 1, doesn't save that many characters
  - No reason to use ++i in a C-style for loop when you have Swift's for-in loop
  - Unnecessarily raises the barrier to learning Swift
- https://github.com/apple/swift-evolution/blob/master/proposals/0004-remove-pre-post-inc-decrement.md

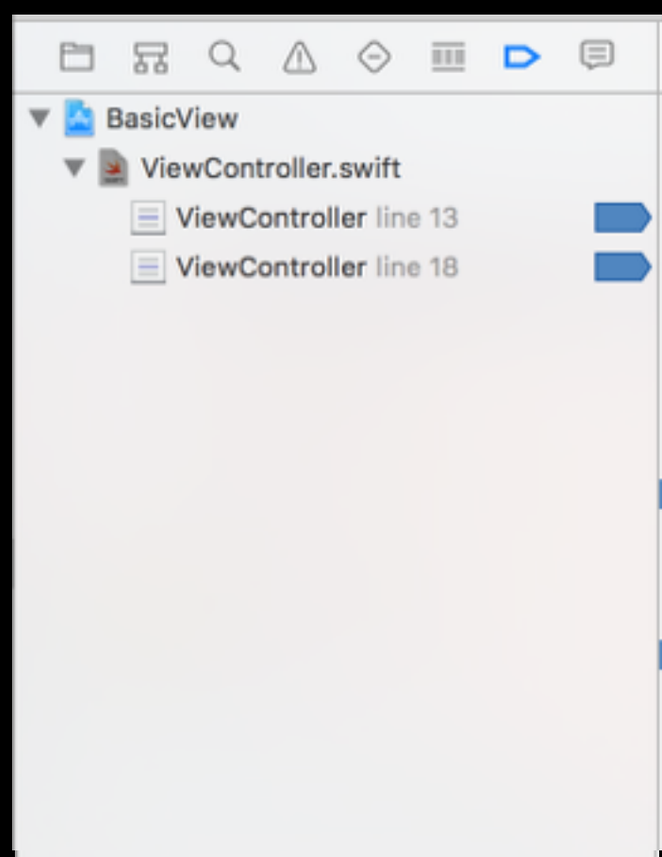
Command line app

- File -> New -> Project
- OS X -> Application -> Command Line Tool
- Run
- Debug

Xcode IDE

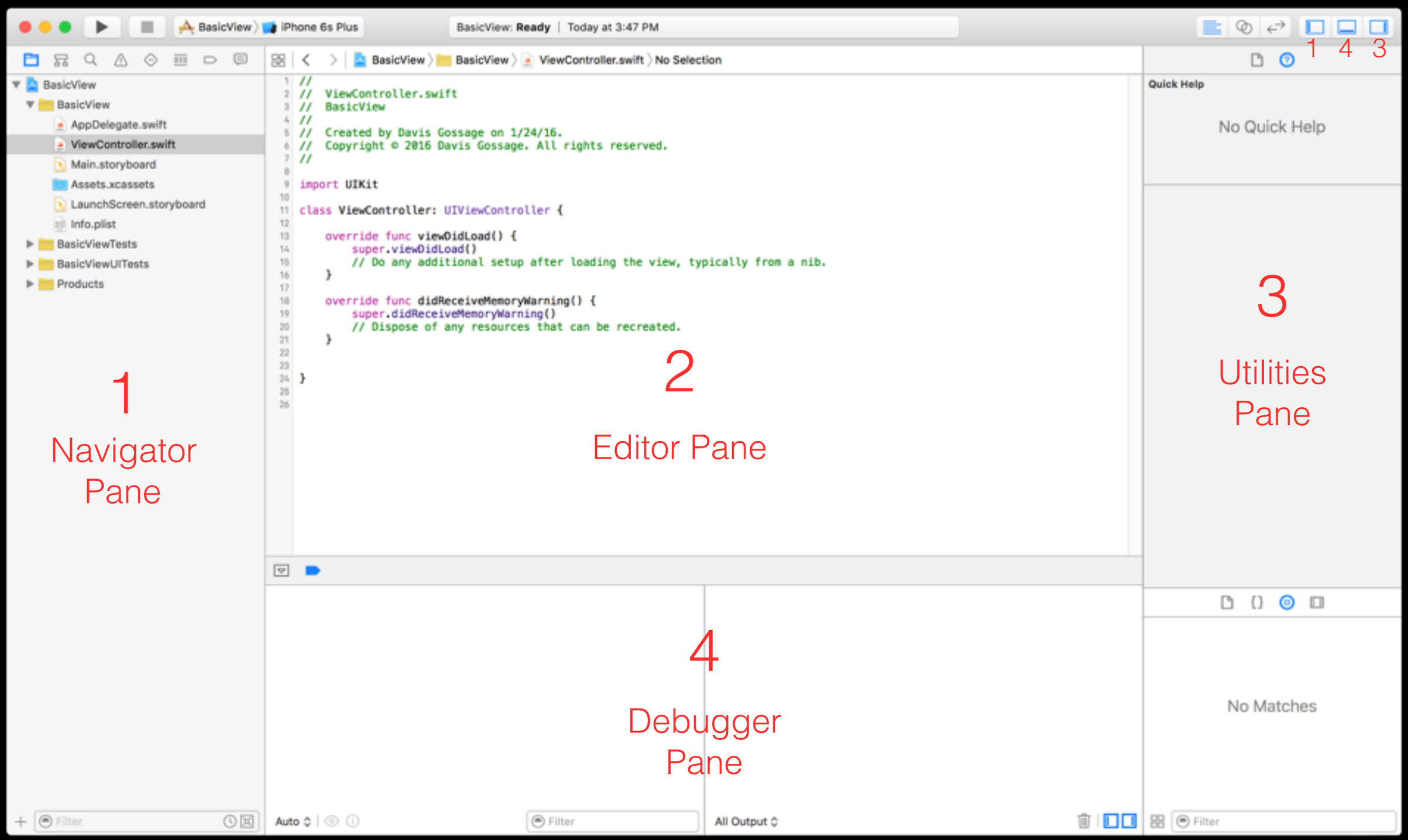

hide/show
1 4 3
BasicView
iPhone 6s Plus
BasicView: Ready | Today at 3:47 PM
BasicView
BasicView
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
BasicViewTests
BasicViewUITests
Products
BasicView 〉 BasicView 〉 ViewController.swift 〉 No Selection
//
// ViewController.swift
// BasicView
//
// Created by Davis Gossage on 1/24/16.
// Copyright © 2016 Davis Gossage. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view, typically from a nib.
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
}
Quick Help
No Quick Help
No Matches
1
Navigator Pane
2
Editor Pane
3
Utilities Pane
4
Debugger Pane
Auto
Filter
All Output

Navigator Pane



- Project (Cmd-1) - project files
- Symbol (Cmd-2) - lists all classes and methods
- Search (Cmd-3) - global text search
- Issue (Cmd-4) - warnings and errors in project
- Test (Cmd-5) - unit tests in project
- Debug (Cmd-6) - manages debug session
- Breakpoints (Cmd-7) - lists all breakpoints
- Report (Cmd-8) - shows logged events

Xcode IDE

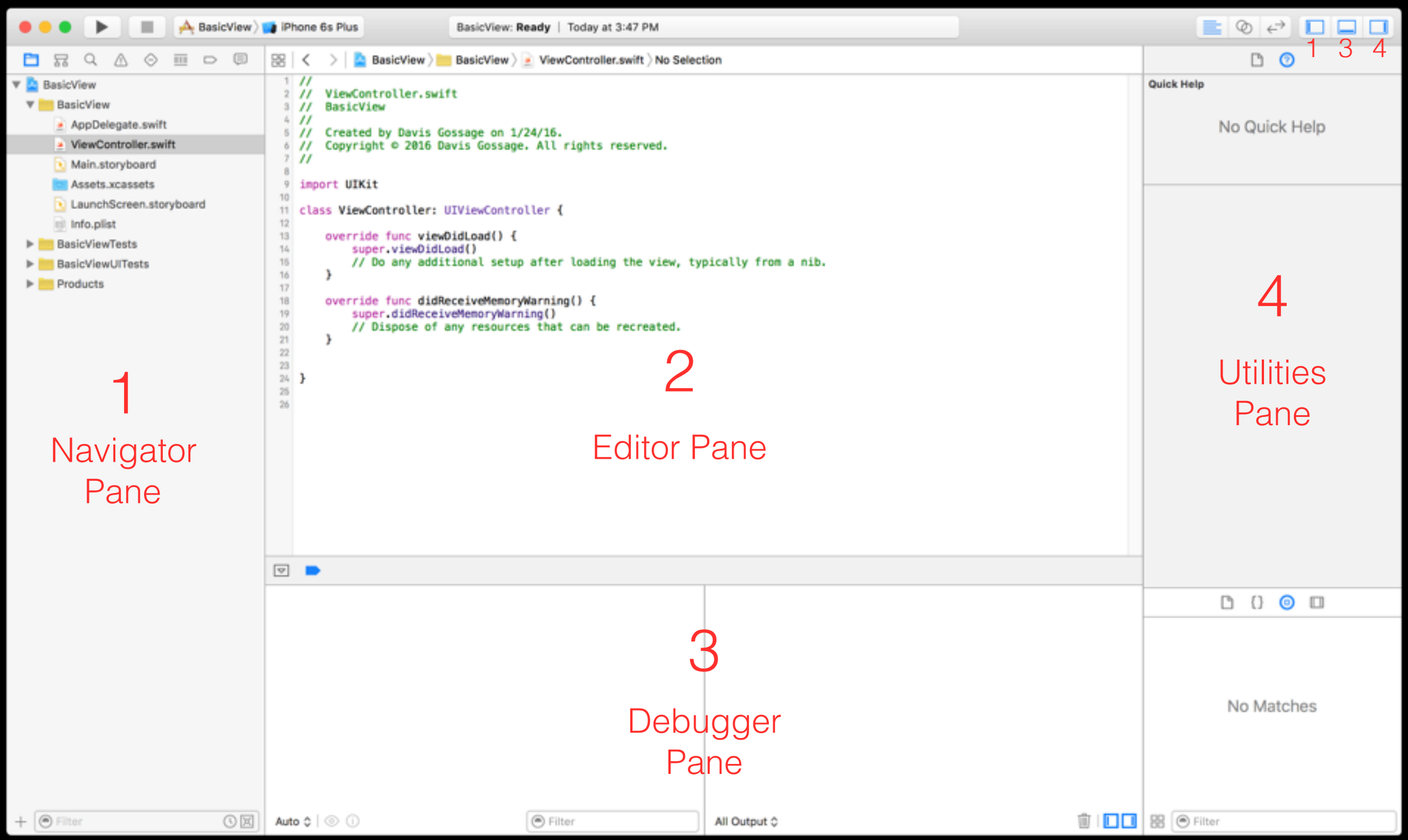

hide/show
1 4 3
BasicView
iPhone 6s Plus
BasicView: Ready | Today at 3:47 PM
BasicView
BasicView
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
BasicViewTests
BasicViewUITests
Products
BasicView 〉 BasicView 〉 ViewController.swift 〉 No Selection
//
// ViewController.swift
// BasicView
//
// Created by Davis Gossage on 1/24/16.
// Copyright © 2016 Davis Gossage. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view, typically from a nib.
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
}
Quick Help
No Quick Help
No Matches
1
Navigator Pane
2
Editor Pane
3
Utilities Pane
4
Debugger Pane
Auto
Filter
All Output

Editor Pane

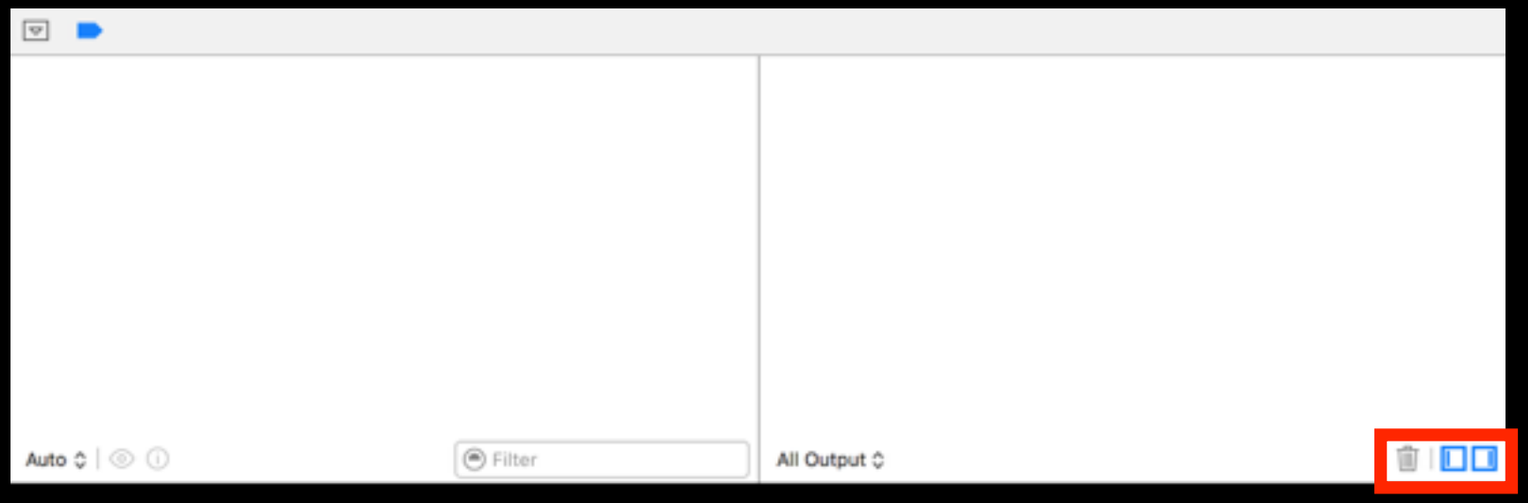
- Code editing and interface design pane
- Jump bar allows for navigating

- The editor pane can be split to include an assistant editor
- Entire project can be tabbed to maintain different states (Cmd-T)

Xcode IDE


hide/show
1 3 4
BasicView
iPhone 6s Plus
BasicView: Ready | Today at 3:47 PM
BasicView
BasicView
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
BasicViewTests
BasicViewUITests
Products
BasicView 〉 BasicView 〉 ViewController.swift 〉 No Selection
//
// ViewController.swift
// BasicView
//
// Created by Davis Gossage on 1/24/16.
// Copyright © 2016 Davis Gossage. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view, typically from a nib.
}
override func didReceiveMemoryWarning() {
super.didReceiveMemoryWarning()
// Dispose of any resources that can be recreated.
}
}
Quick Help
No Quick Help
No Matches
1
Navigator Pane
2
Editor Pane
3
Debugger Pane
4
Utilities Pane
Auto
Filter
All Output

Debugging Pane



- Right panel shows debugging output (print statements)
- Left side will show variables when debug mode is entered (breakpoint reached)
- Either of these views can be collapsed

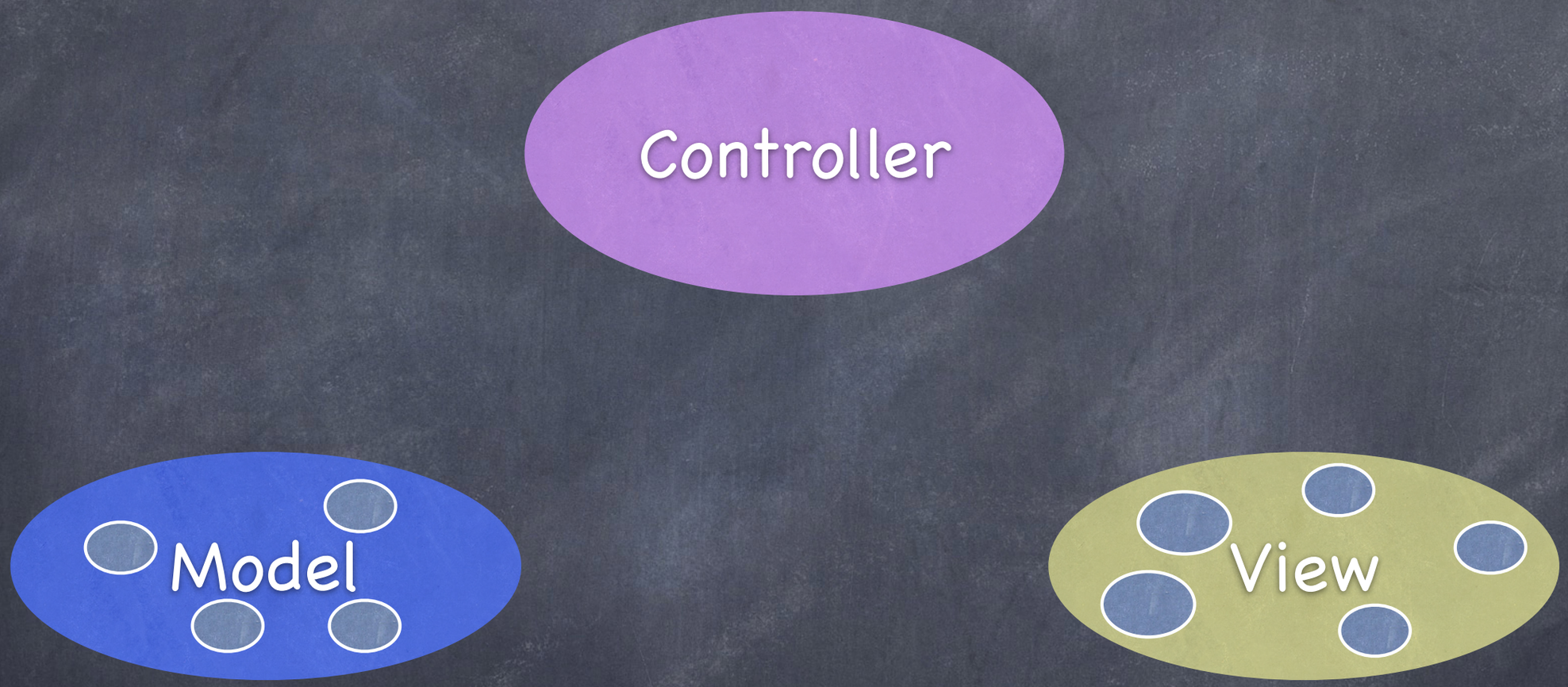
ToDo List App

- File -> New -> Project
- iOS Application -> Single View
- Name it “ToDoList”
- Files and folders in Project Navigator do not matchup directly with the project directory. The project directory should not be edited.

.storyboard

- Storyboards are composed of Scenes. A scene contains a hierarchy of **views**, often with a single **view controller**.
  - Scenes can be connected to other Scenes with **segues**.
- Clicking on .storyboard files will open Xcode's graphical editor called Interface Builder (IB)
- Interface Builder shows the document outline on the left and the canvas on the right

MVC



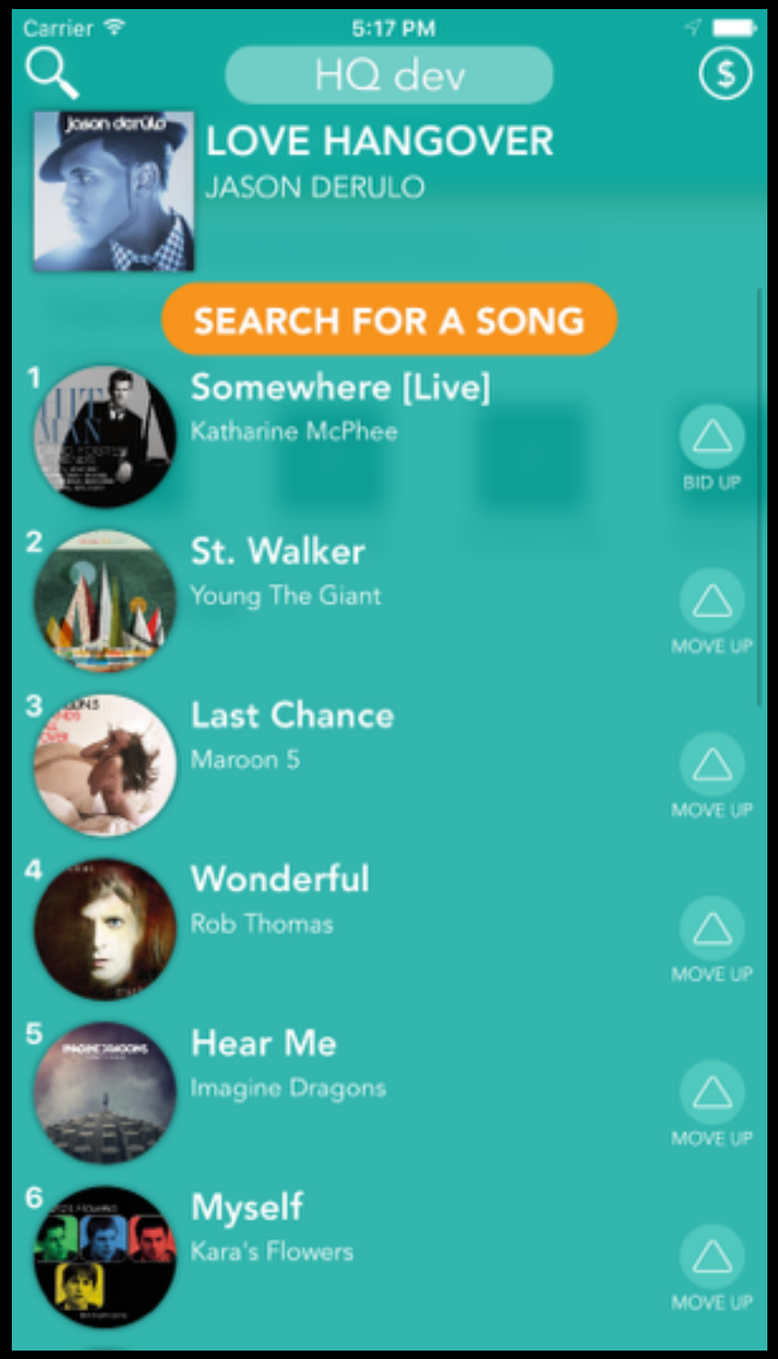
Divide objects in your program into 3 "camps."

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

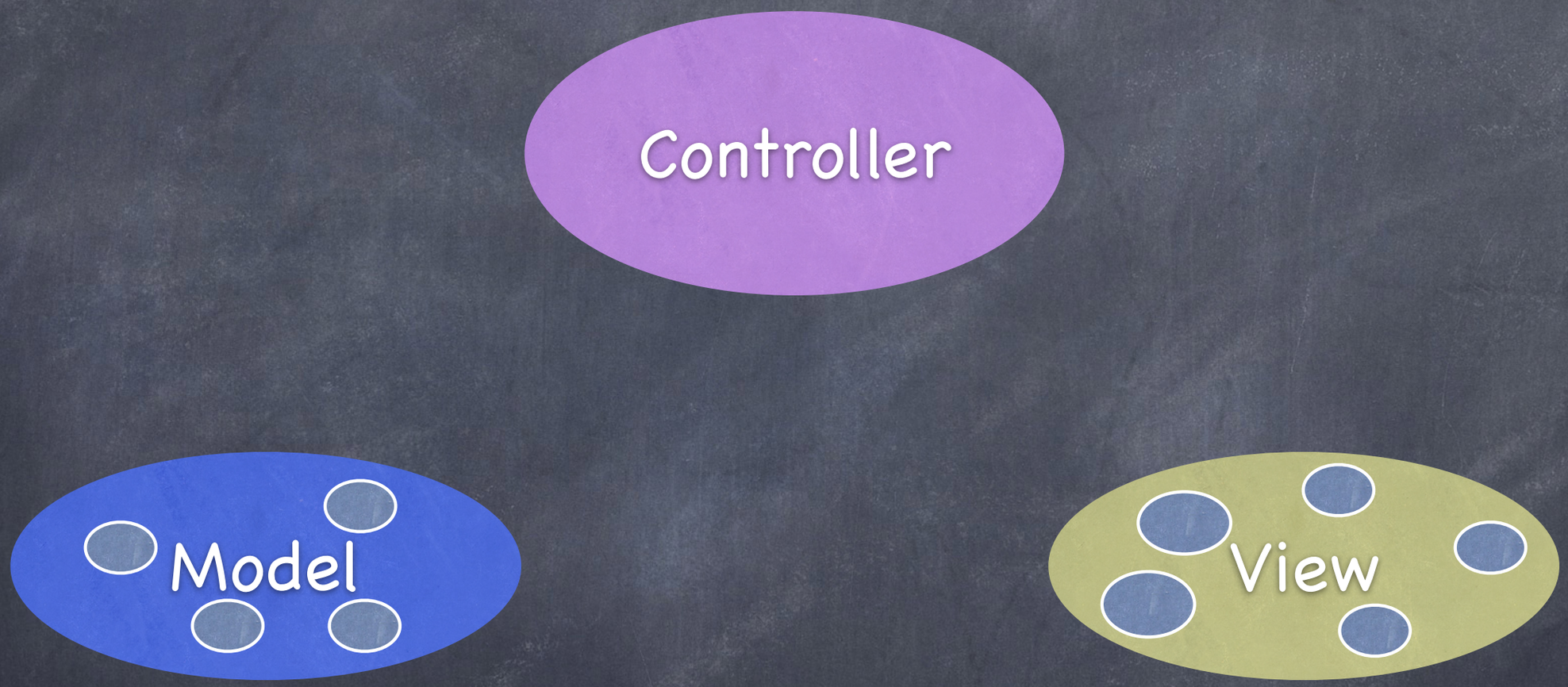
Views

- Scenes are built by creating a hierarchy of UIView objects
- Rectangles stacked on rectangles…
- UIView is the parent class, its children include:
  - **UILabel** - single line of text
  - **UITextView** - multiple lines of text
  - **UITextField** - accepts user inputted text
  - **UITableView**
- UIKit Framework Reference

View Hierarchy



MVC



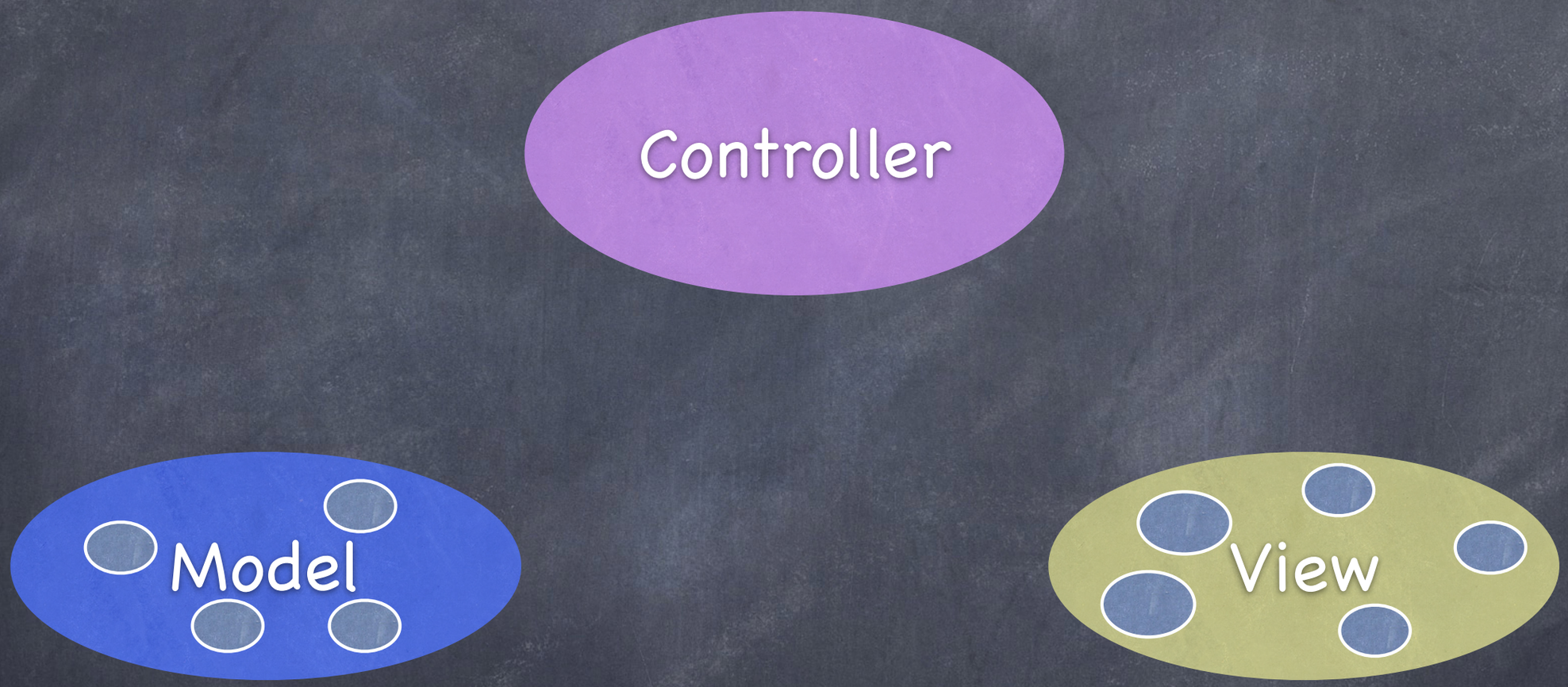
Divide objects in your program into 3 "camps."

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

View Controllers

- Manipulates views
- View Controllers in the storyboard are backed by a .swift file
  - View Controllers make connections with the views using outlets (IBOutlet) and actions (IBAction)
- UIViewController is the parent class, it's children include:
  - UINavigationController - manages a navigation bar, allows for 'pushing' to new views
  - UITableViewController - embeds a tableview and includes methods for adding content

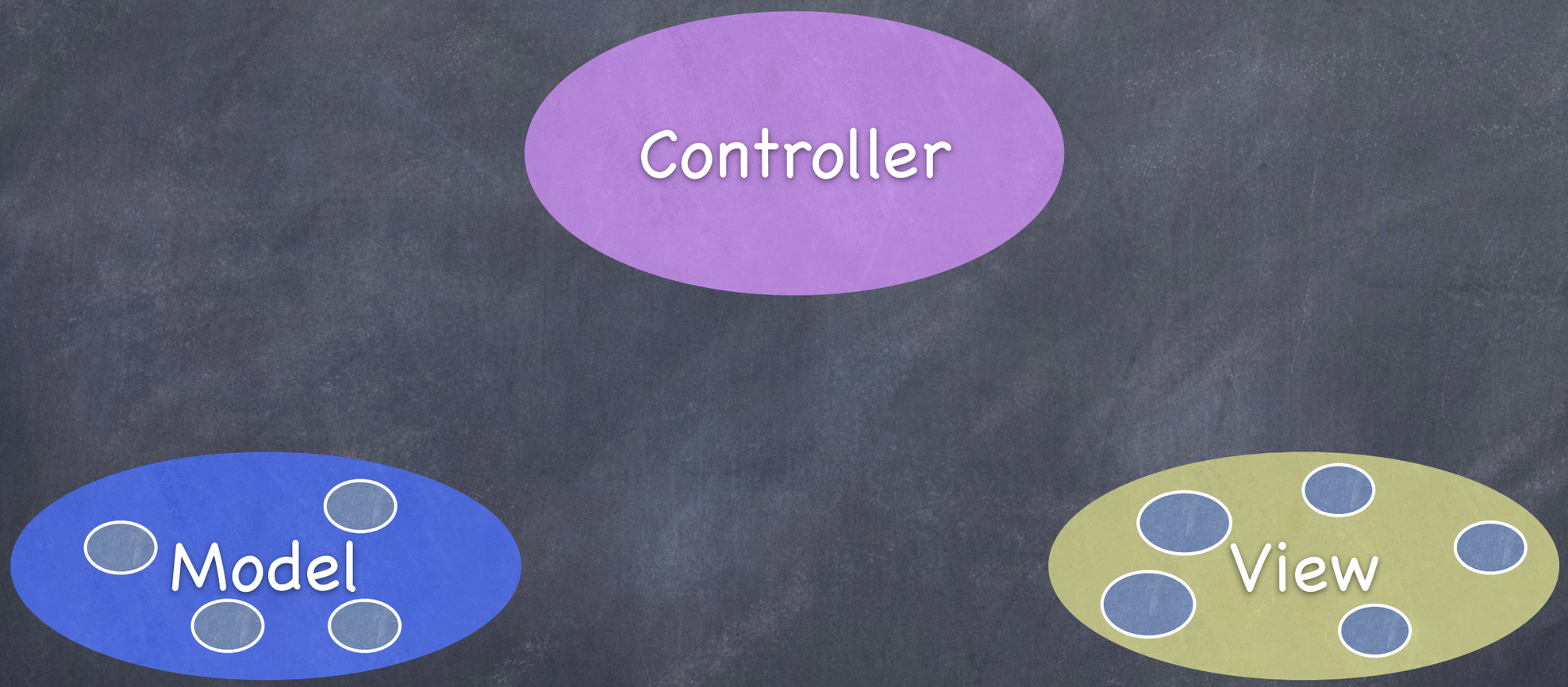
MVC



Model = What your application is (but not how it is displayed)

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

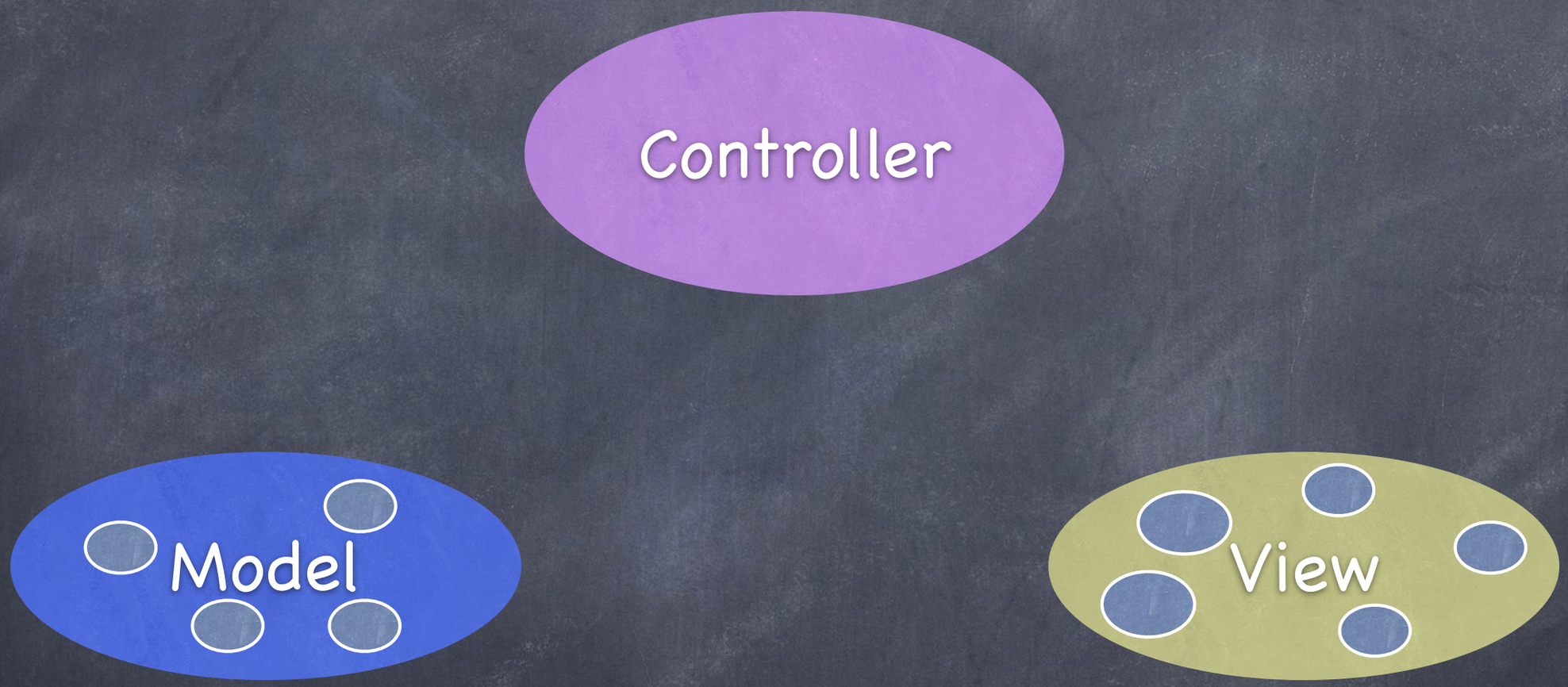
MVC



Controller = How your Model is presented to the user (UI logic)

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

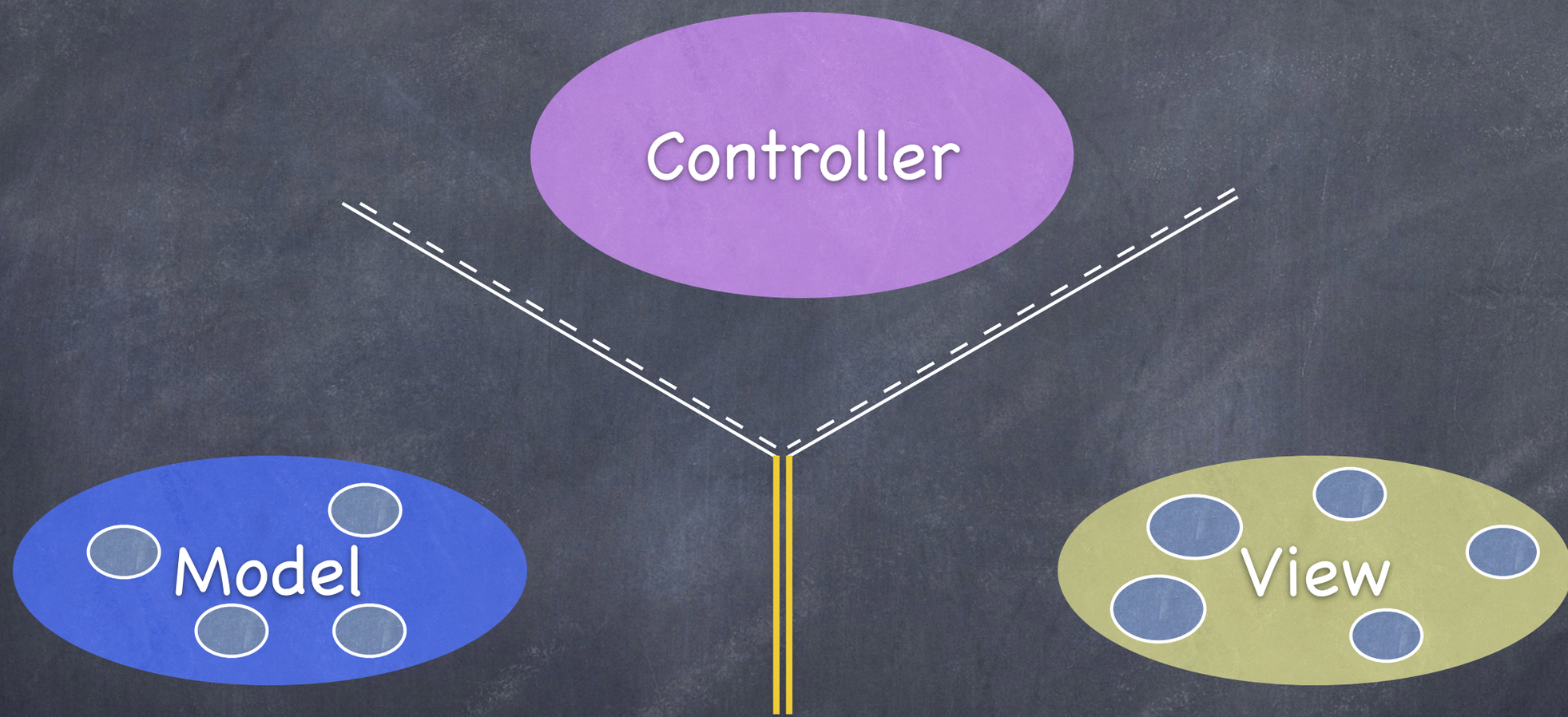
MVC



View = Your Controller's minions

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

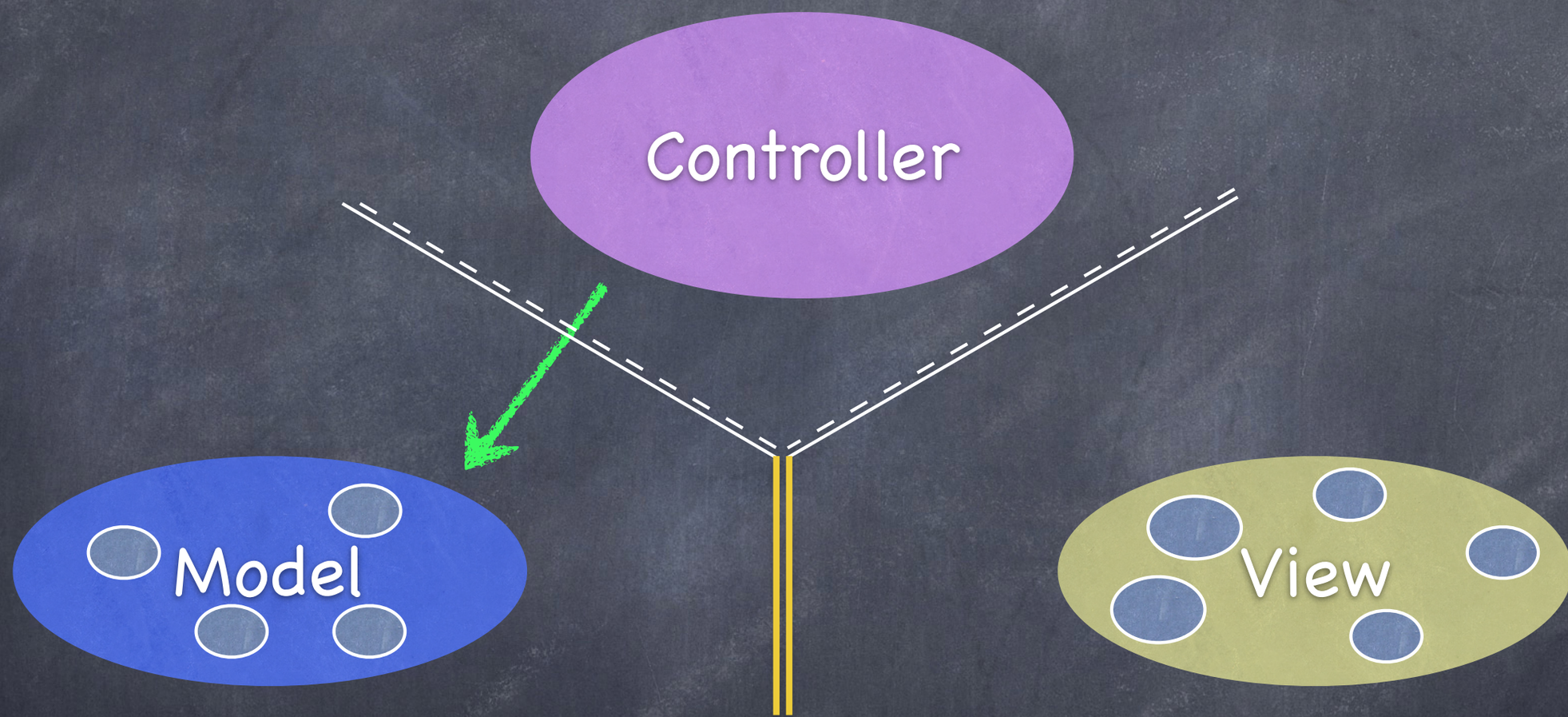
MVC



It's all about managing communication between camps

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

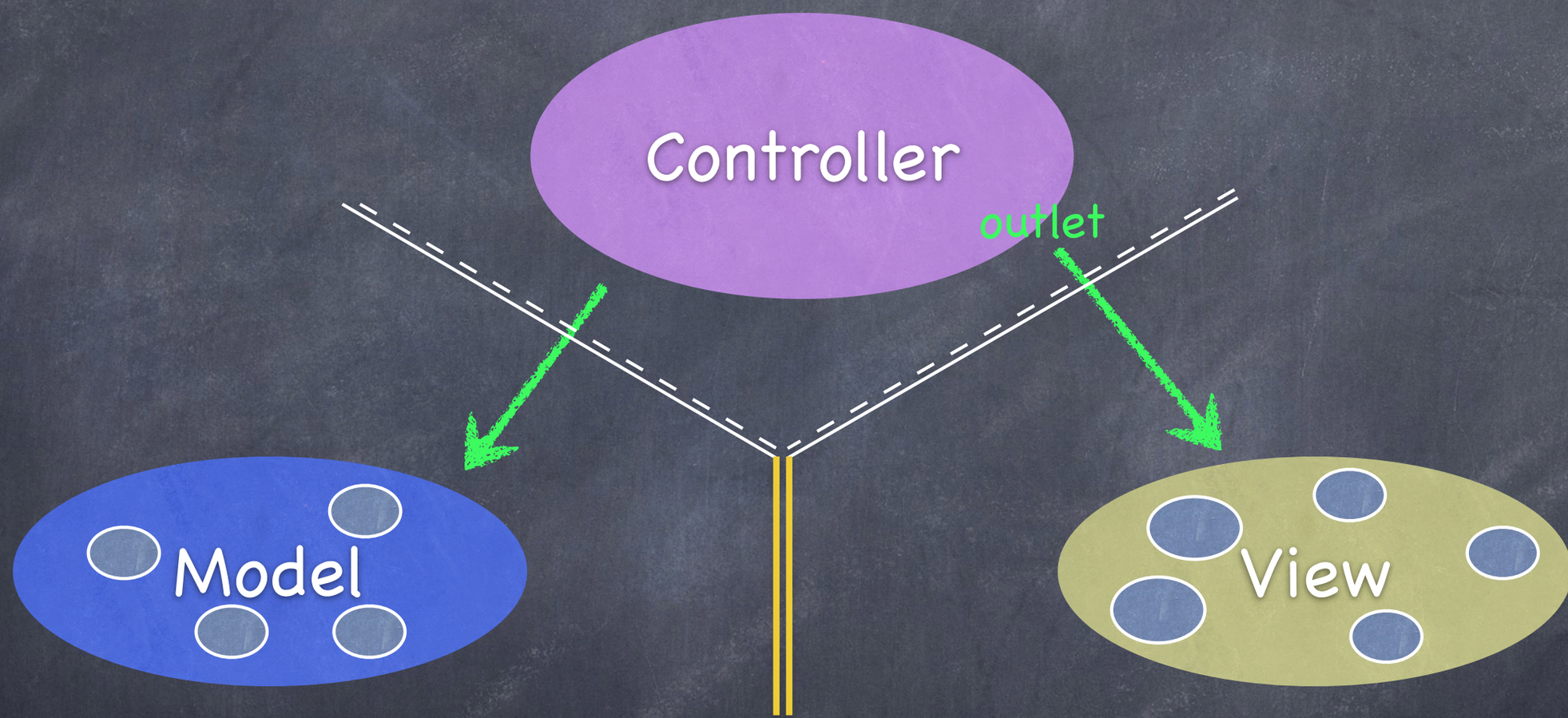
MVC



Controllers can always talk directly to their Model.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

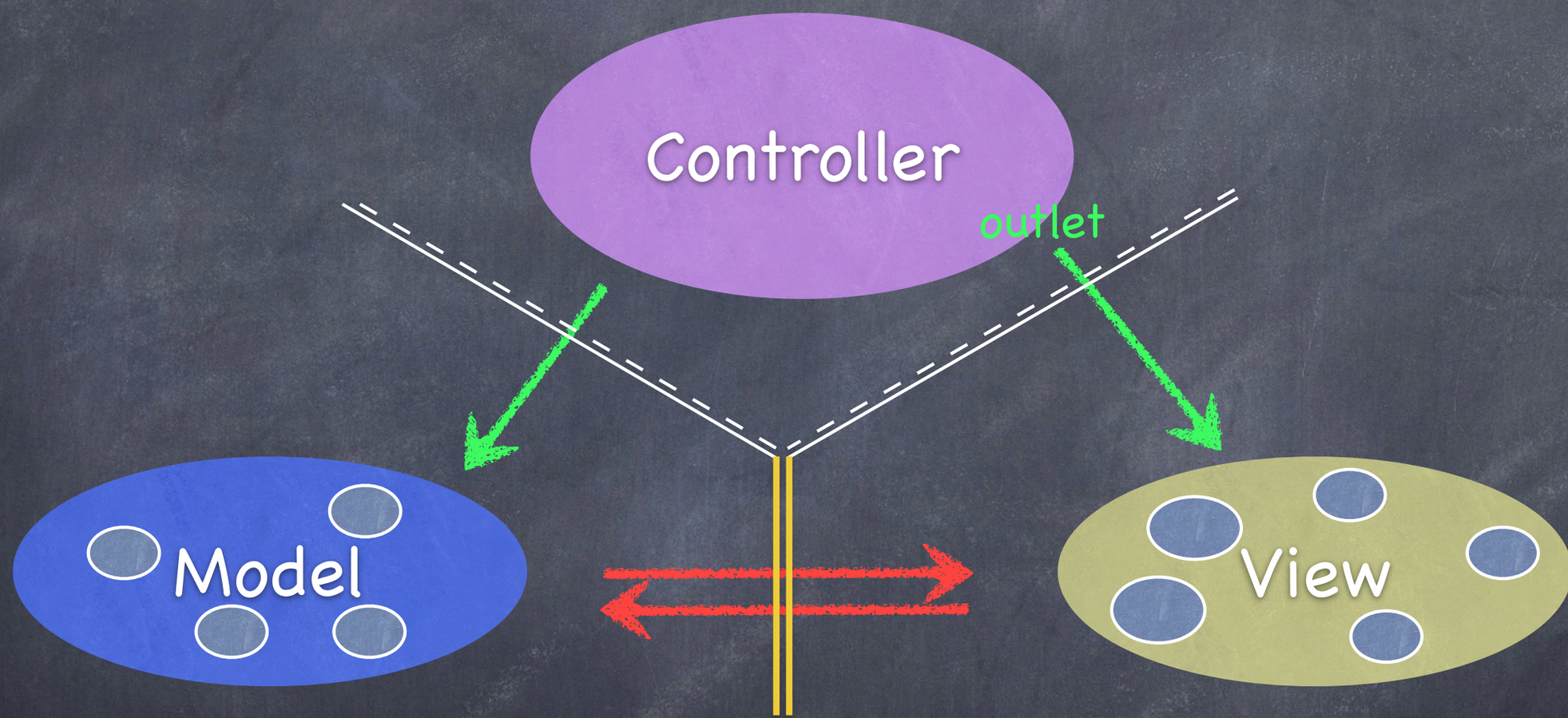
MVC



Controllers can also talk directly to their View.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

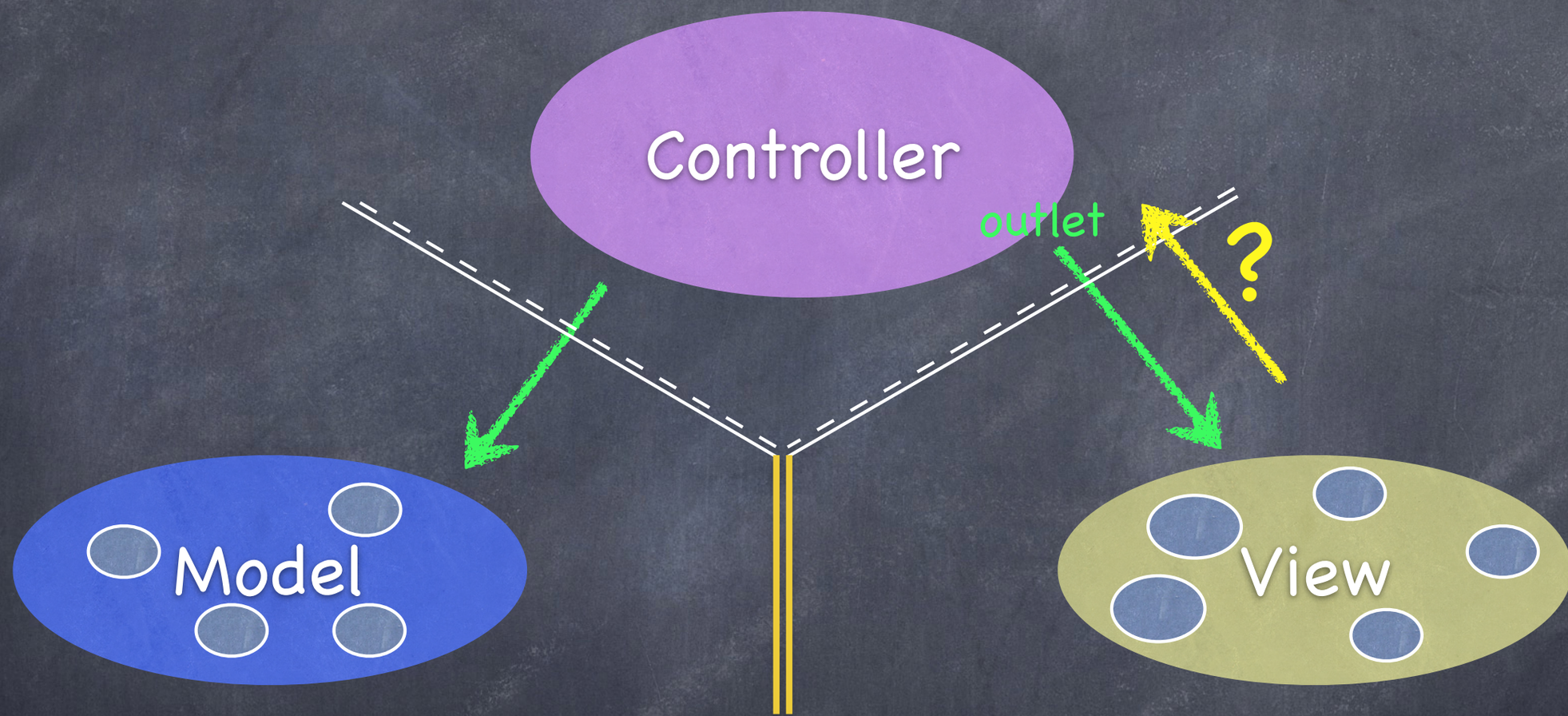
MVC



The Model and View should never speak to each other.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

MVC

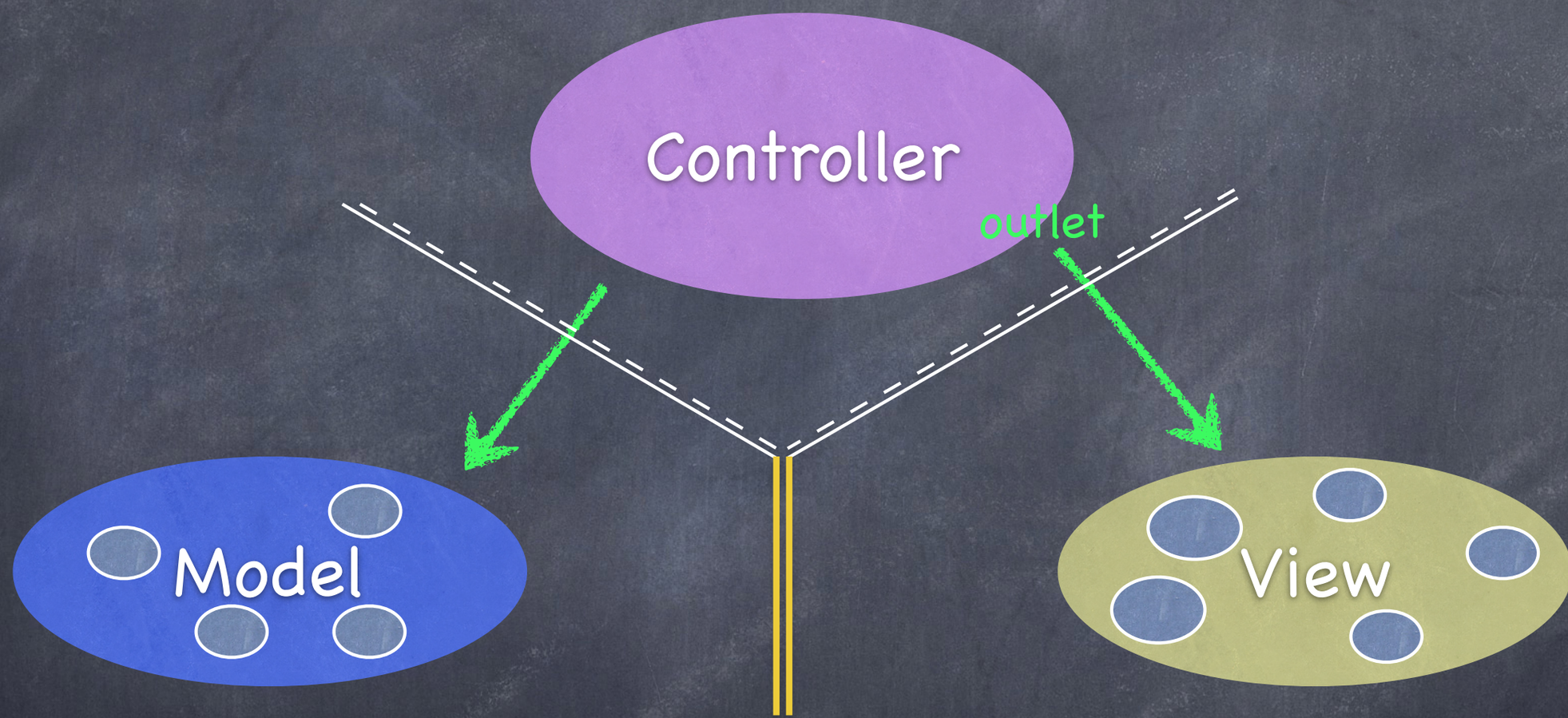


Can the View speak to its Controller?

CS193p
Winter 2015

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

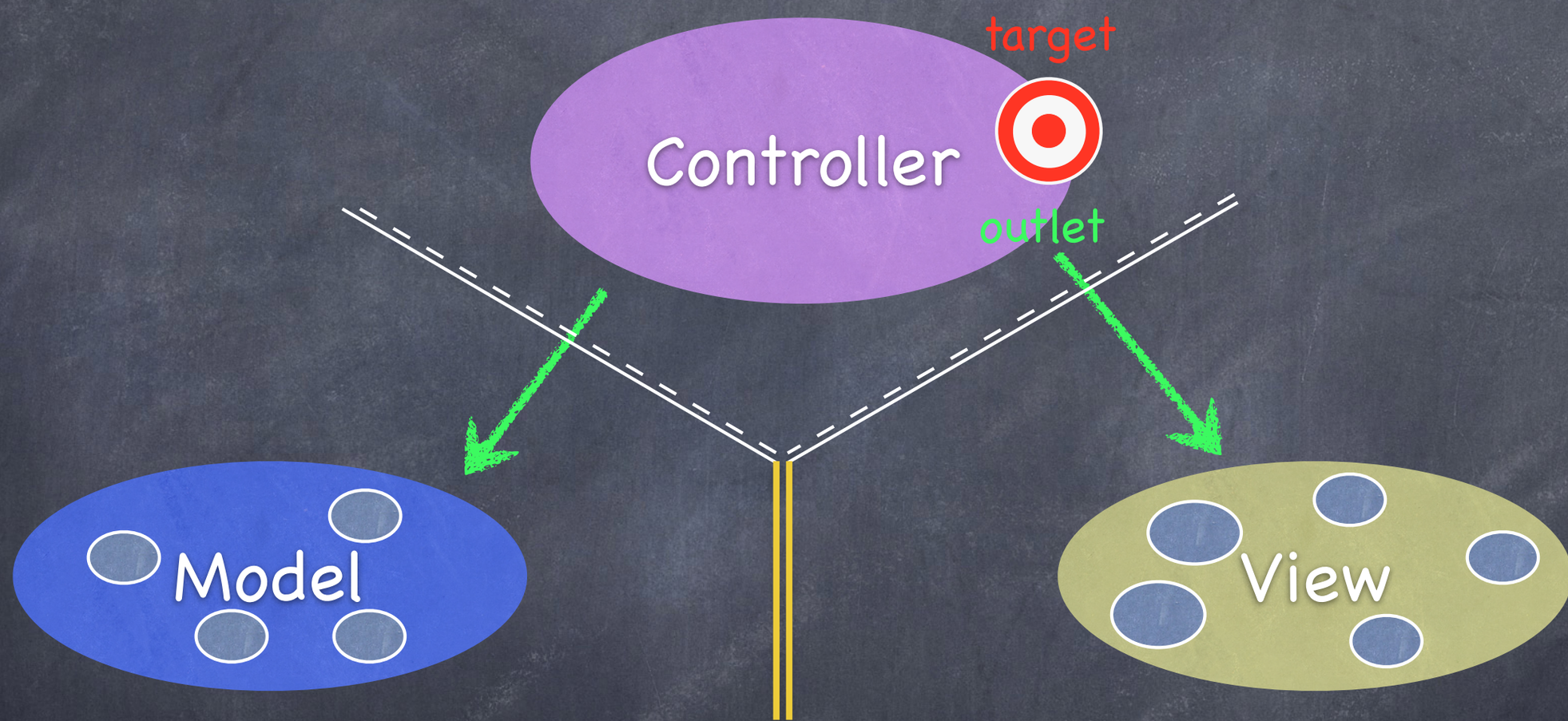
MVC



Sort of. Communication is "blind" and structured.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

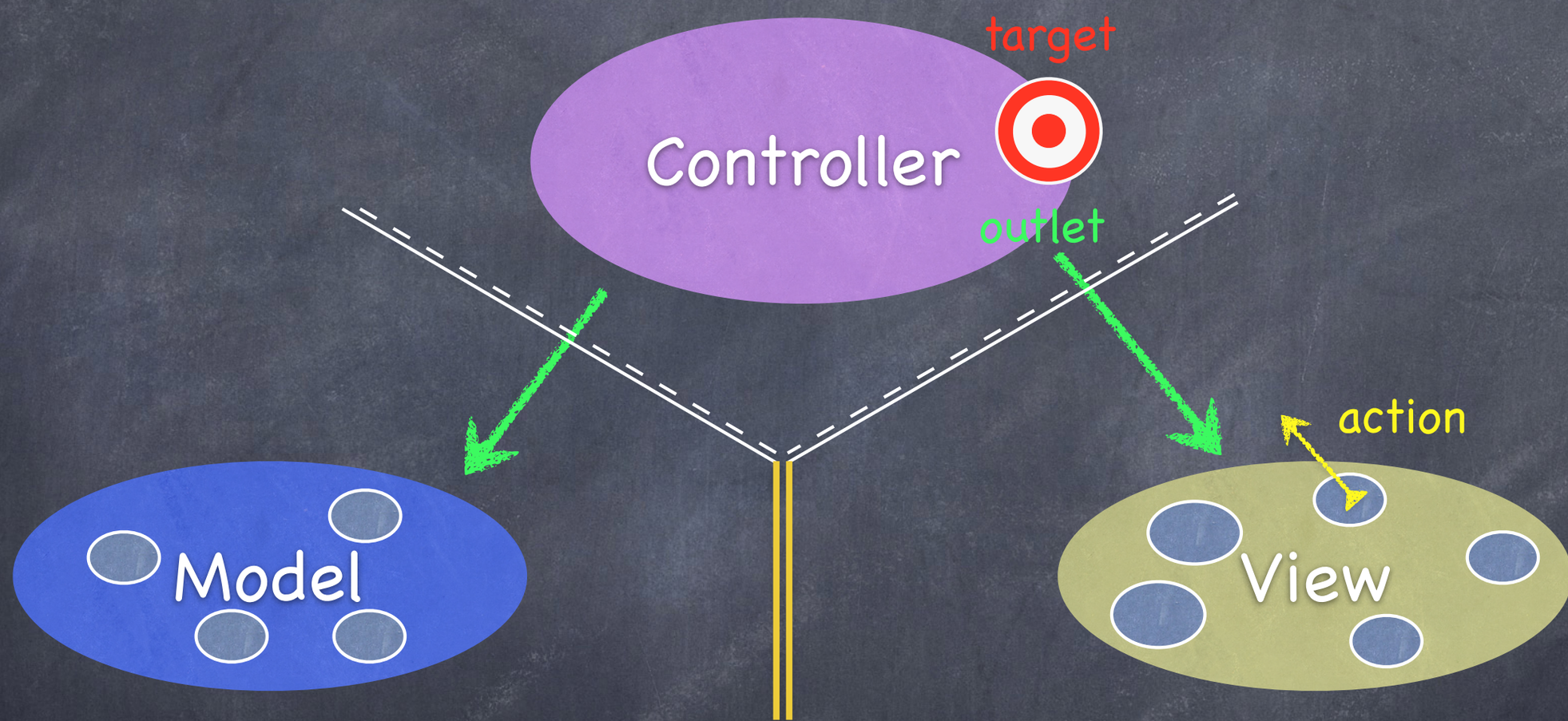
MVC



The Controller can drop a target on itself.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

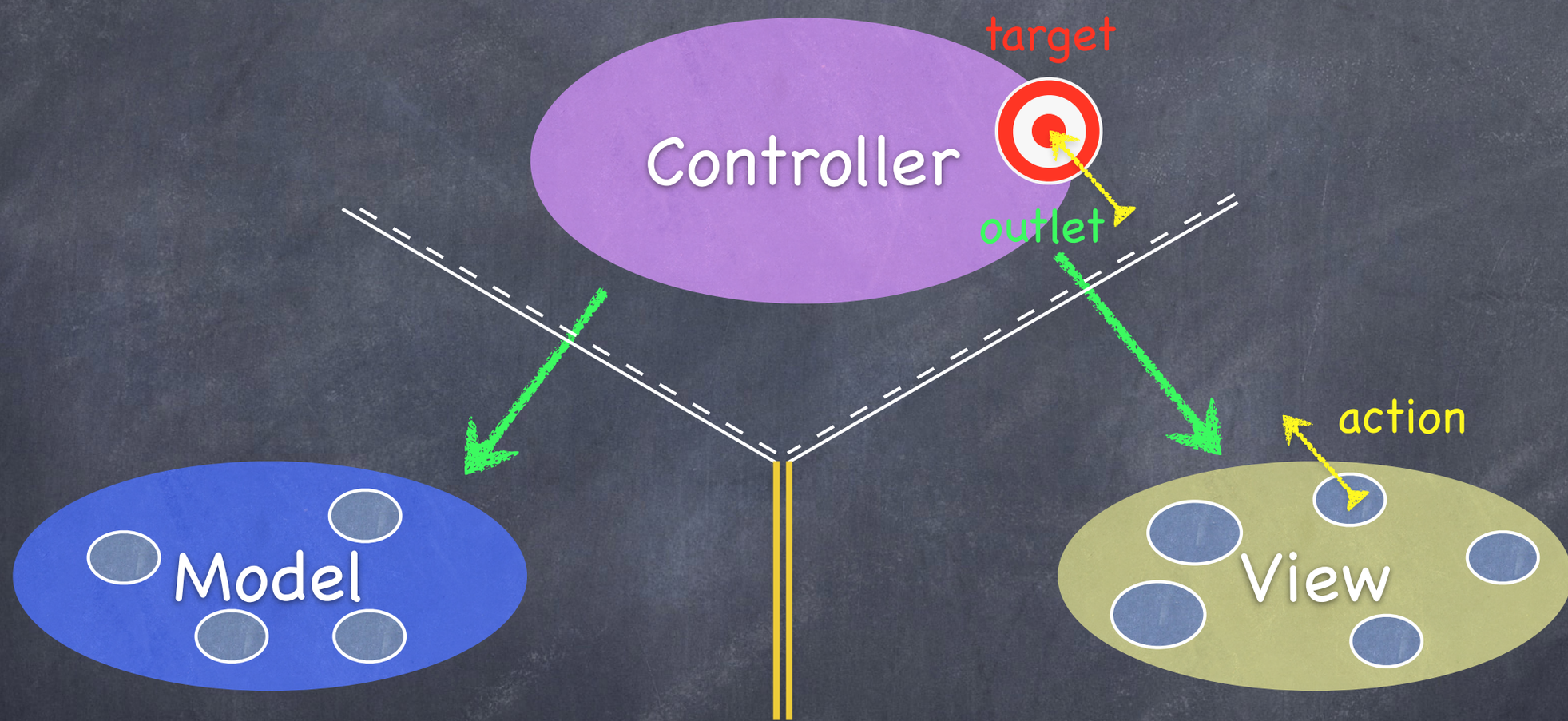
MVC



Then hand out an action to the View.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

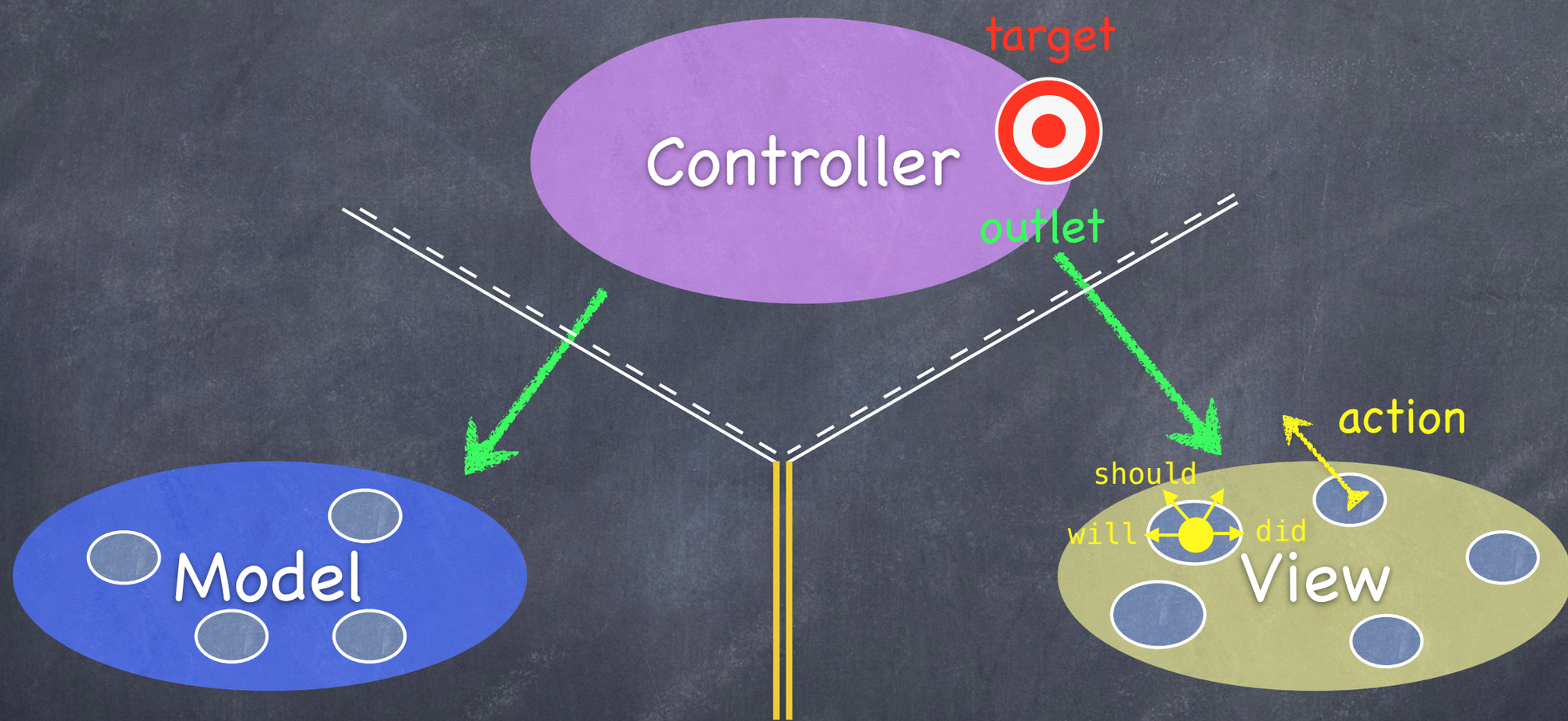
MVC



The View sends the action when things happen in the UI.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

MVC

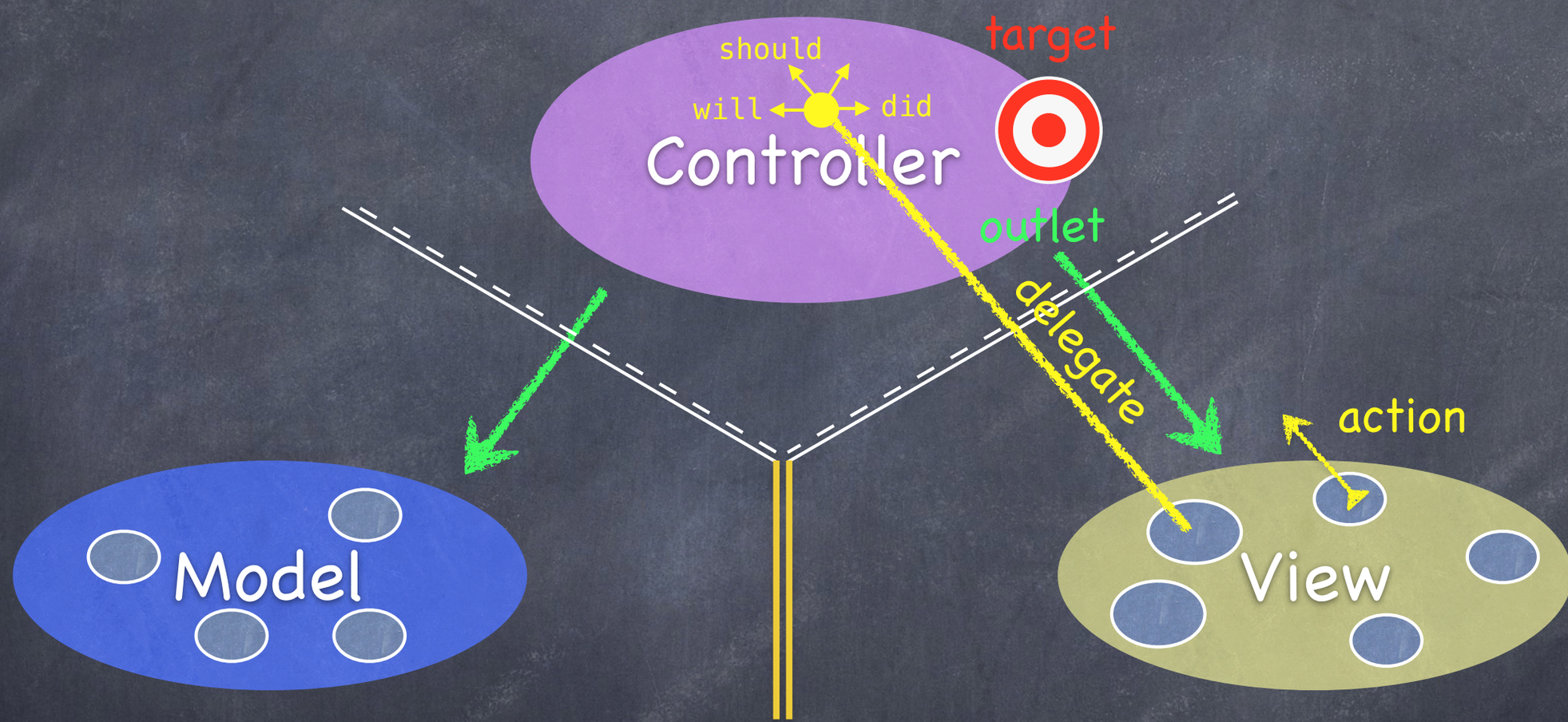


Sometimes the View needs to synchronize with the Controller.

CS193p
Winter 2015

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

MVC

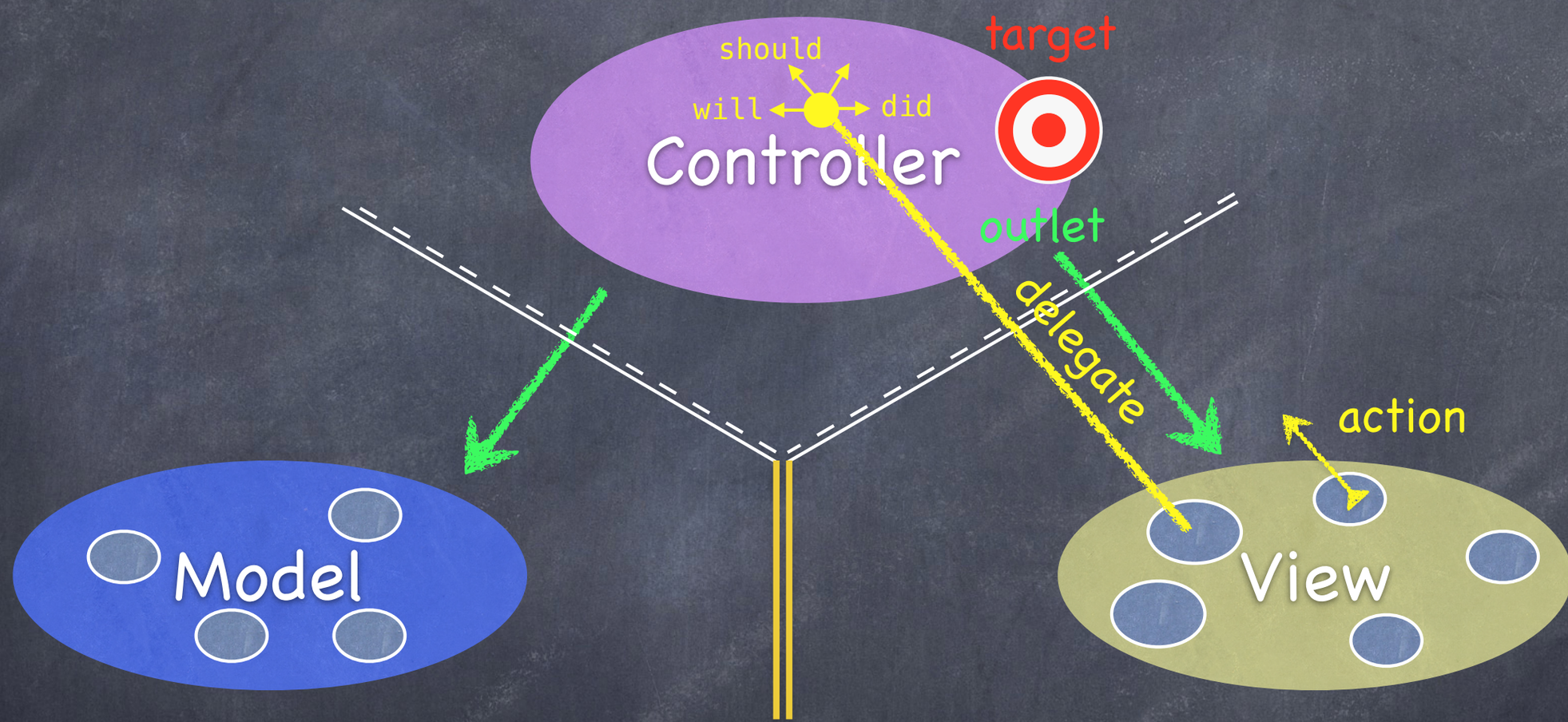


The Controller sets itself as the View's delegate.

CS193p
Winter 2015

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

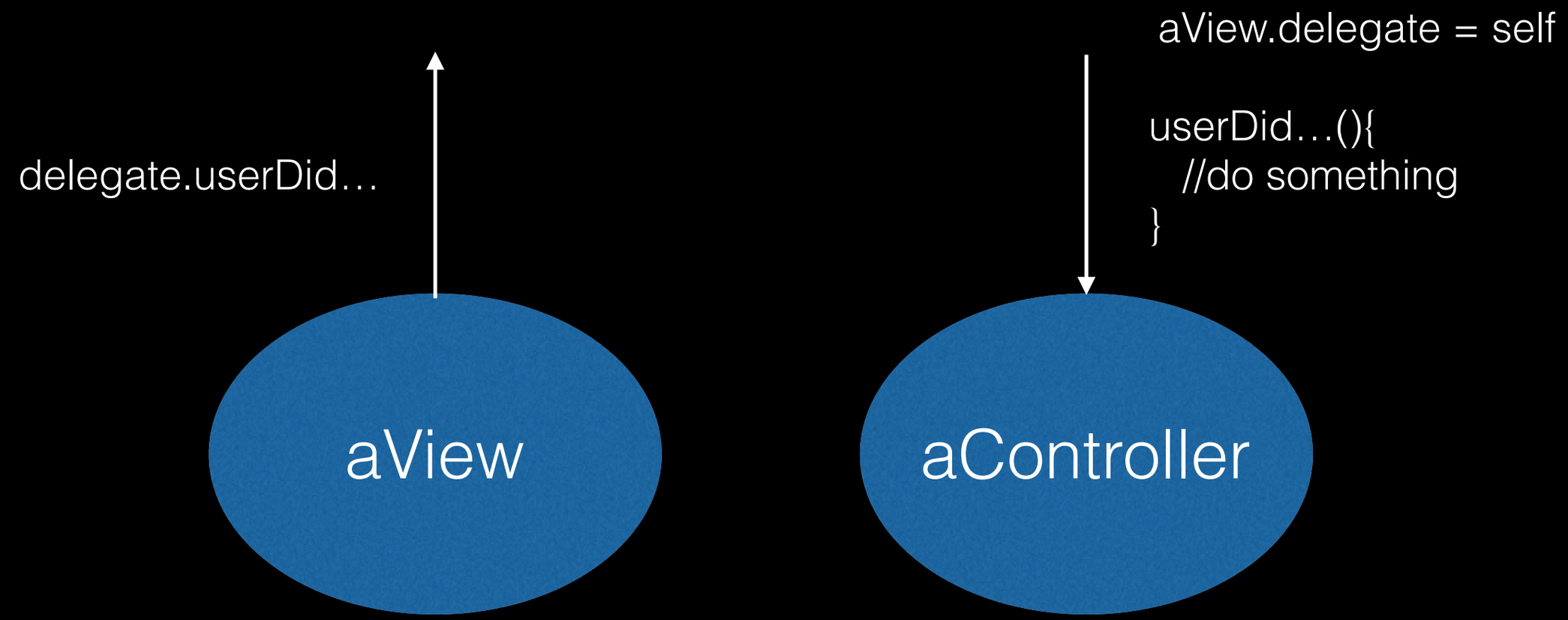
MVC



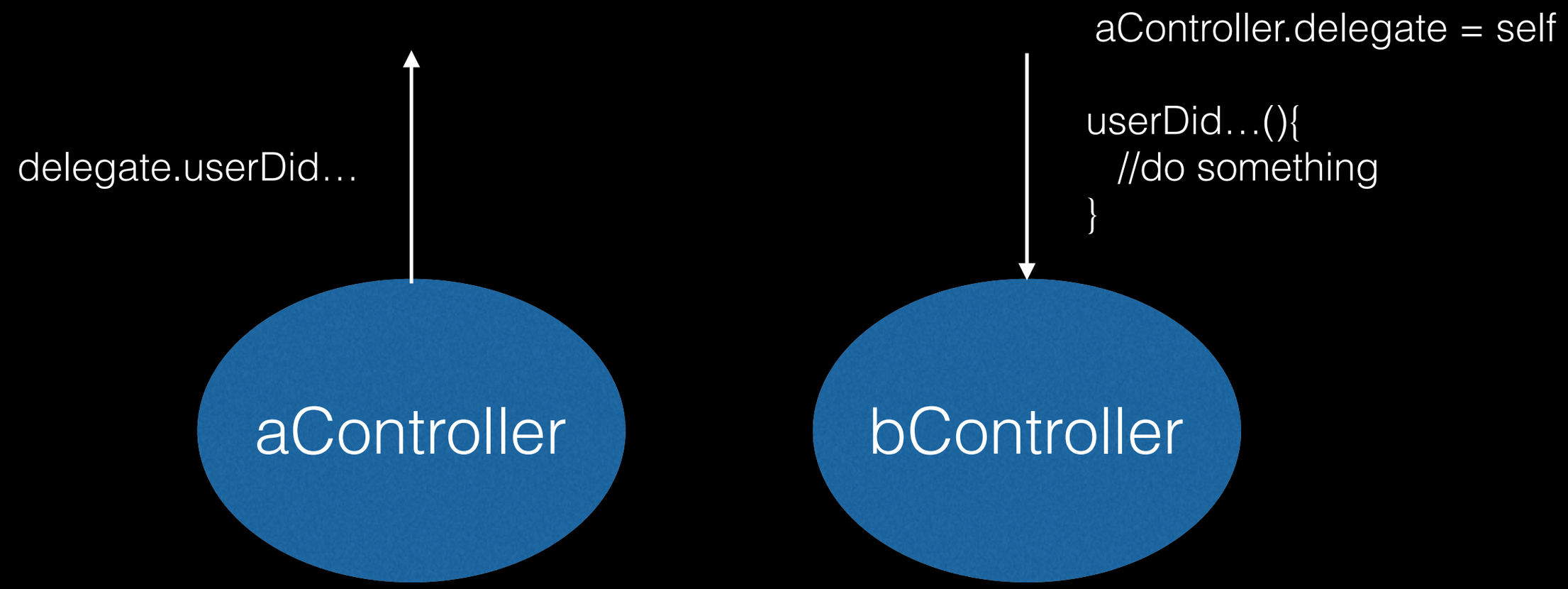
The delegate is set via a protocol (i.e. it's "blind" to class).

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

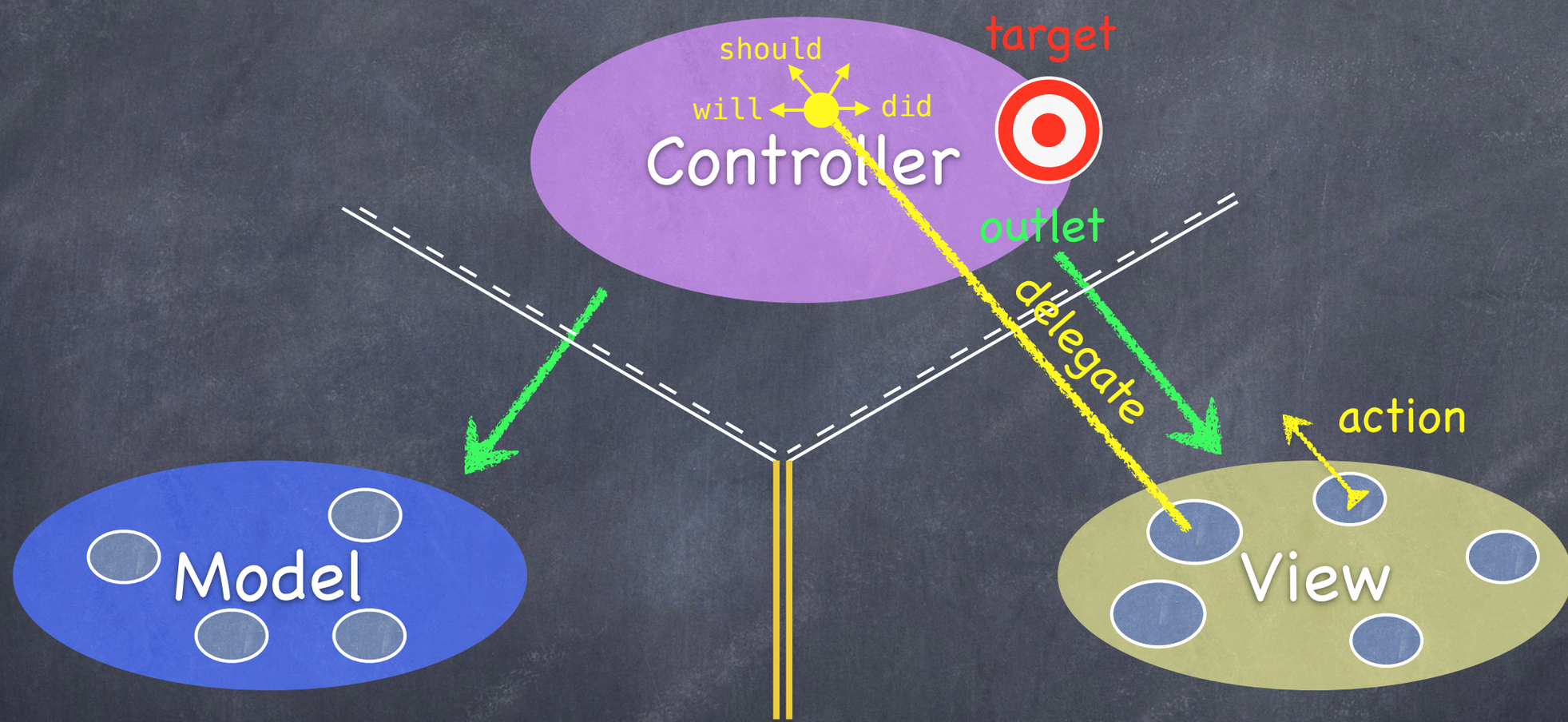
Delegates


aView.delegate = self
delegate.userDid…
userDid…(){
//do something
}
aView
aController

Delegates


aController.delegate = self
delegate.userDid…
userDid…(){
//do something
}
aController
bController

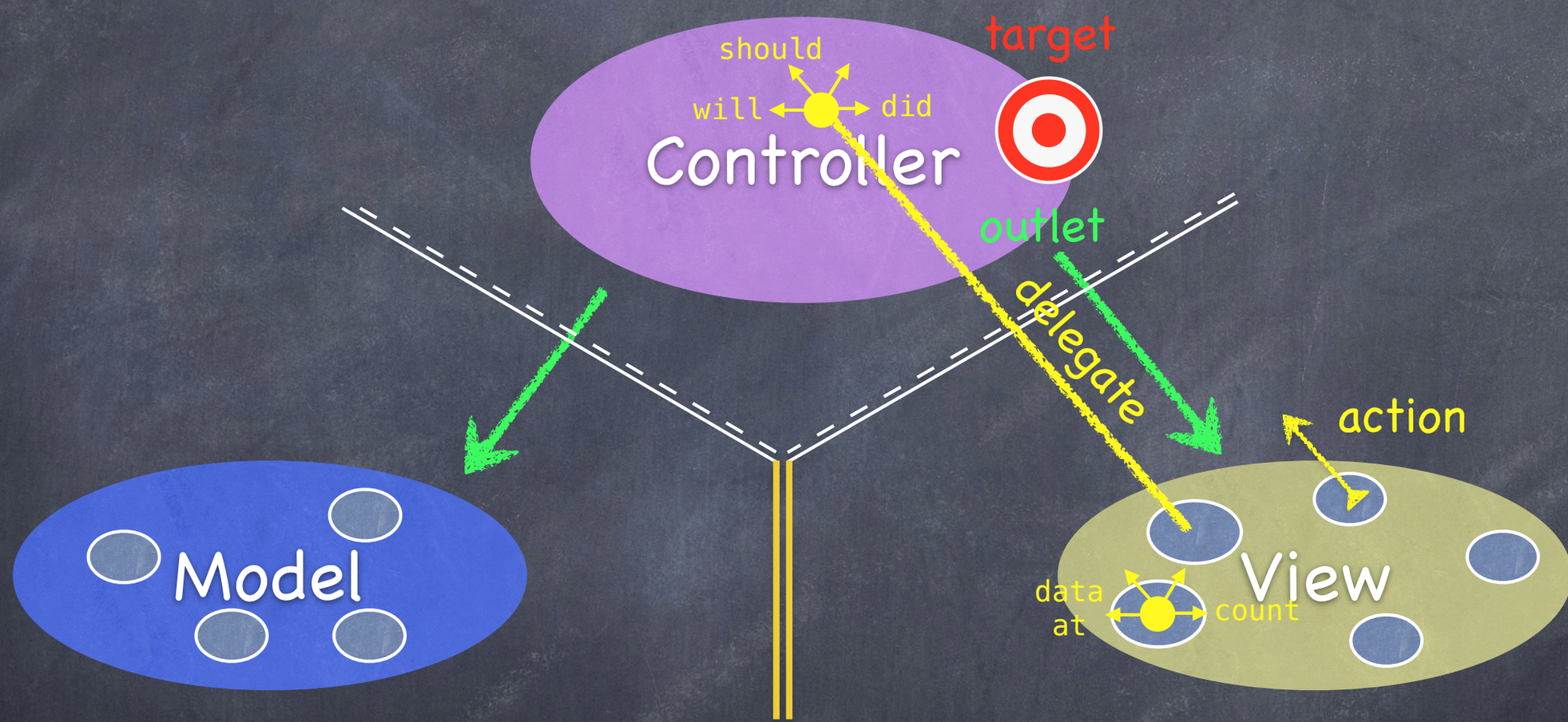
MVC



Views do not own the data they display.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

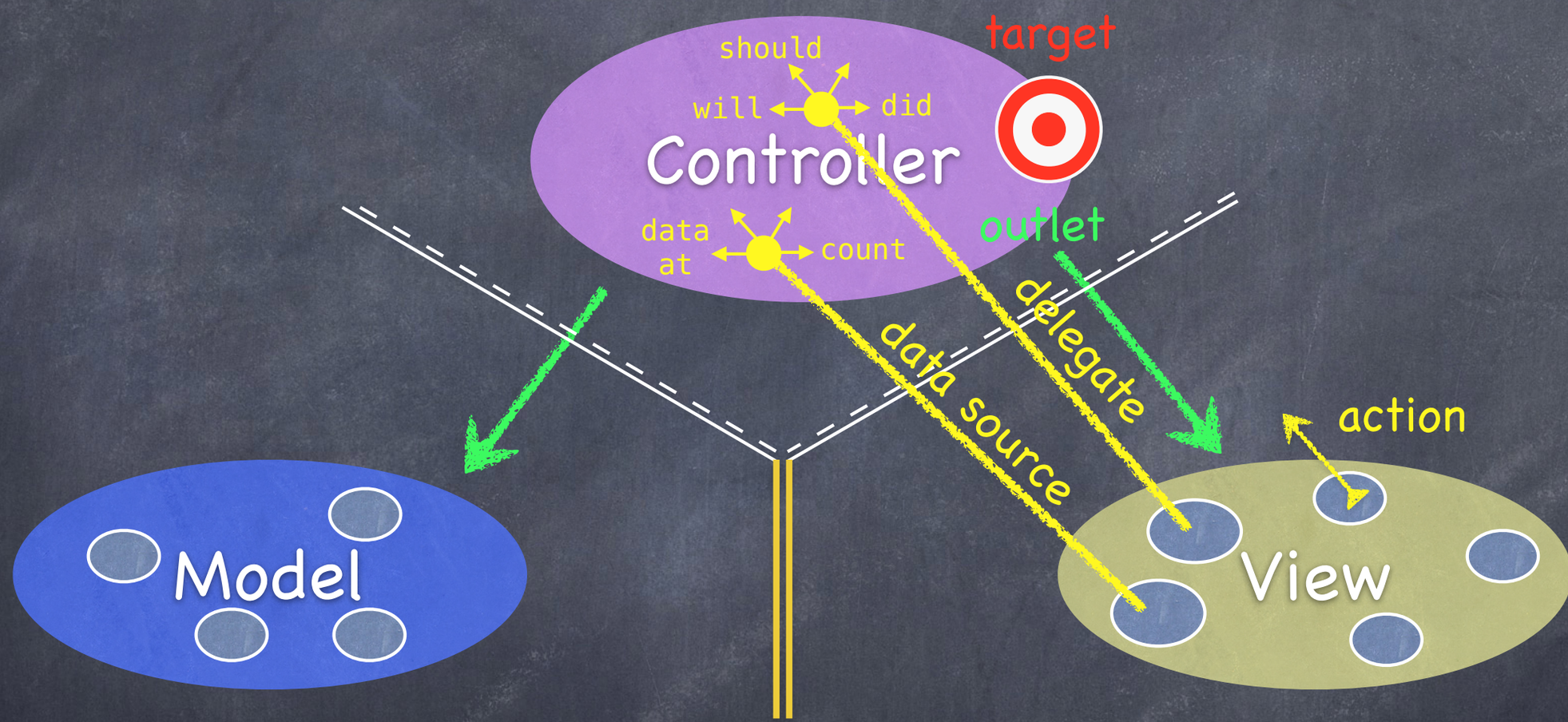
MVC



So, if needed, they have a protocol to acquire it.

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

MVC

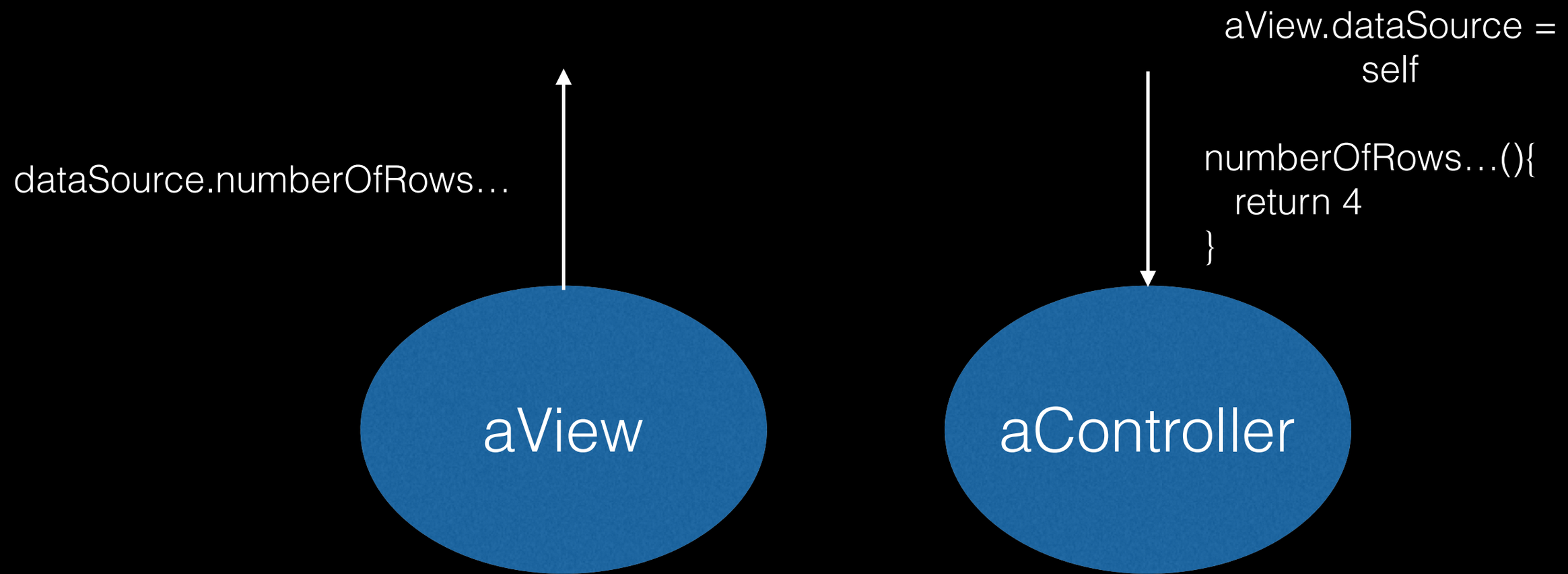


Controllers are almost always that data source (not Model!).

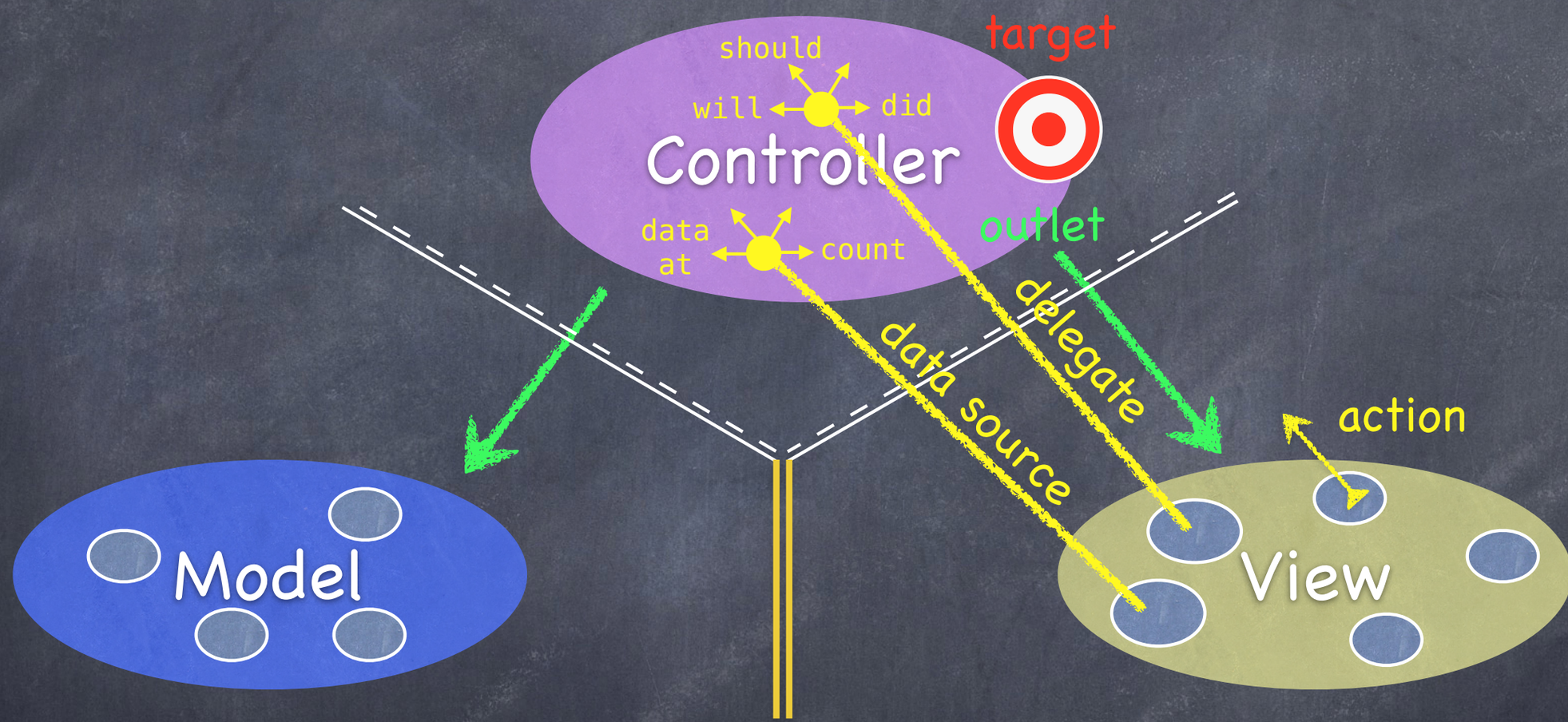
CS193p
Winter 2015

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

Data Source


aView.dataSource =
self
dataSource.numberOfRows…
numberOfRows…(){
return 4
}
aView
aController

MVC



Controllers are almost always that data source (not Model!).

CS193p
Winter 2015

This work by Stanford University is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 United States License. Based on a work at cs193p.stanford.edu

In Summary

- Communication from V to VC, or VC to VC happens 4 different ways:
  - Outlet: allows the VC to change the view
    - Ex: make a label show some text
  - Action: allows the view to notify the VC of some user action
    - Ex: user taps a button
  - Delegate : allows a view or VC to blindly publish an event. This is received by any VC that has claimed responsibility for that event
    - Ex: user taps on a row in a tableview
  - Data source : allows a view to blindly request information. This is fulfilled by any VC that claims responsibility for this information
    - Ex: tableview requests how many rows it needs to display

Now we can begin...

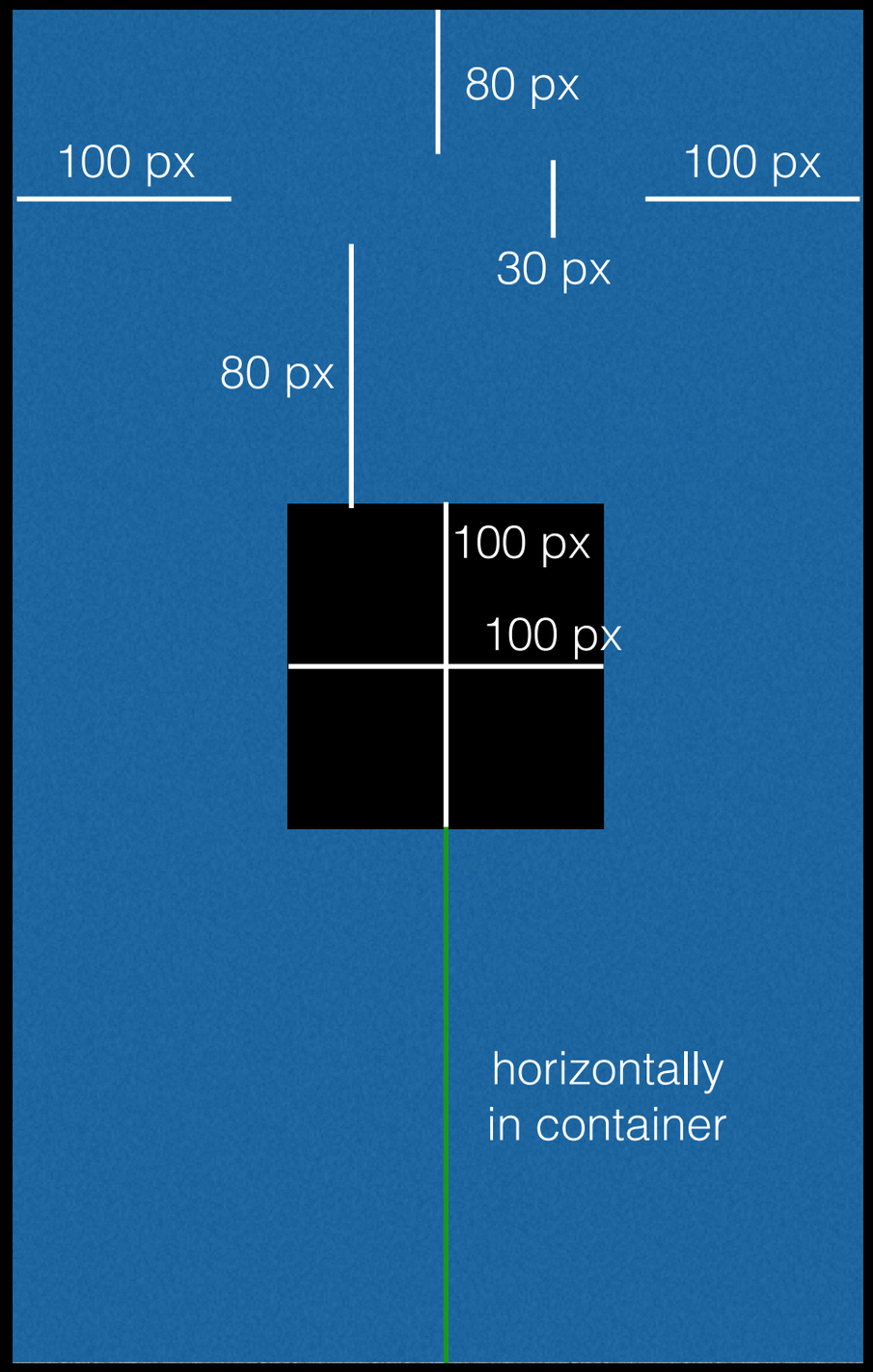
- Let’s place a UILabel on our view in Main.storyboard

Auto Layout - Background

- When there was one iPhone size, views could be defined simply with X, Y, Height, Width
- iPhone 4 complicated things with a screen resolution increase of 2x
  - Fine, just scale everything up and include @2x graphics
- Now:
  - iPhone 4S and before
  - iPhone 5, 5C, 5S (taller, can’t scale)
  - iPhone 6, 6S (non-proportionally taller and wider)
  - iPhone 6+, 6S+ (Same as 6, 6S but scale to @3x)
  - iPods, iPads

Auto Layout - Constraints

- Views need dynamic rules for defining where they are to be placed, and what size they will be
  - Called constraints
- Constraints can be absolute, or based on the location of other views
- X and Y position, along with width and height must be resolved for each view


80 px
100 px
100 px
30 px
80 px
100 px
100 px
horizontally
in container