

iOS App 開發基礎班

# MVC (Model-View-Controller)

---

蔡智強 Denny Tsai

denny@hpd.io

<https://iosdev.hpd.io>



# MVC是什麼

---

- 一種軟體開發的設計模式
- 把軟體分成三大主要的陣營
  - Model
  - View
  - Controller

# Model

---

- 軟體中的資料部分
- 通訊錄中的聯絡人、LINE裡面的對話記錄、遊戲裡面的排行榜分數...
- 計數器Demo的計數器變數
- 複雜的時候會使用物件的方式來處理

# View

---

- 直接面向使用者的部分
- UILabel, UIButton, UIView, 地圖, LINE的對話泡泡, iOS的滑動解鎖
- 將Model的資料呈現在畫面上

# Controller

---

- 負責控制Model跟View之間的溝通
- 管理所有的運算和邏輯
- 讓App運作的「腦袋」

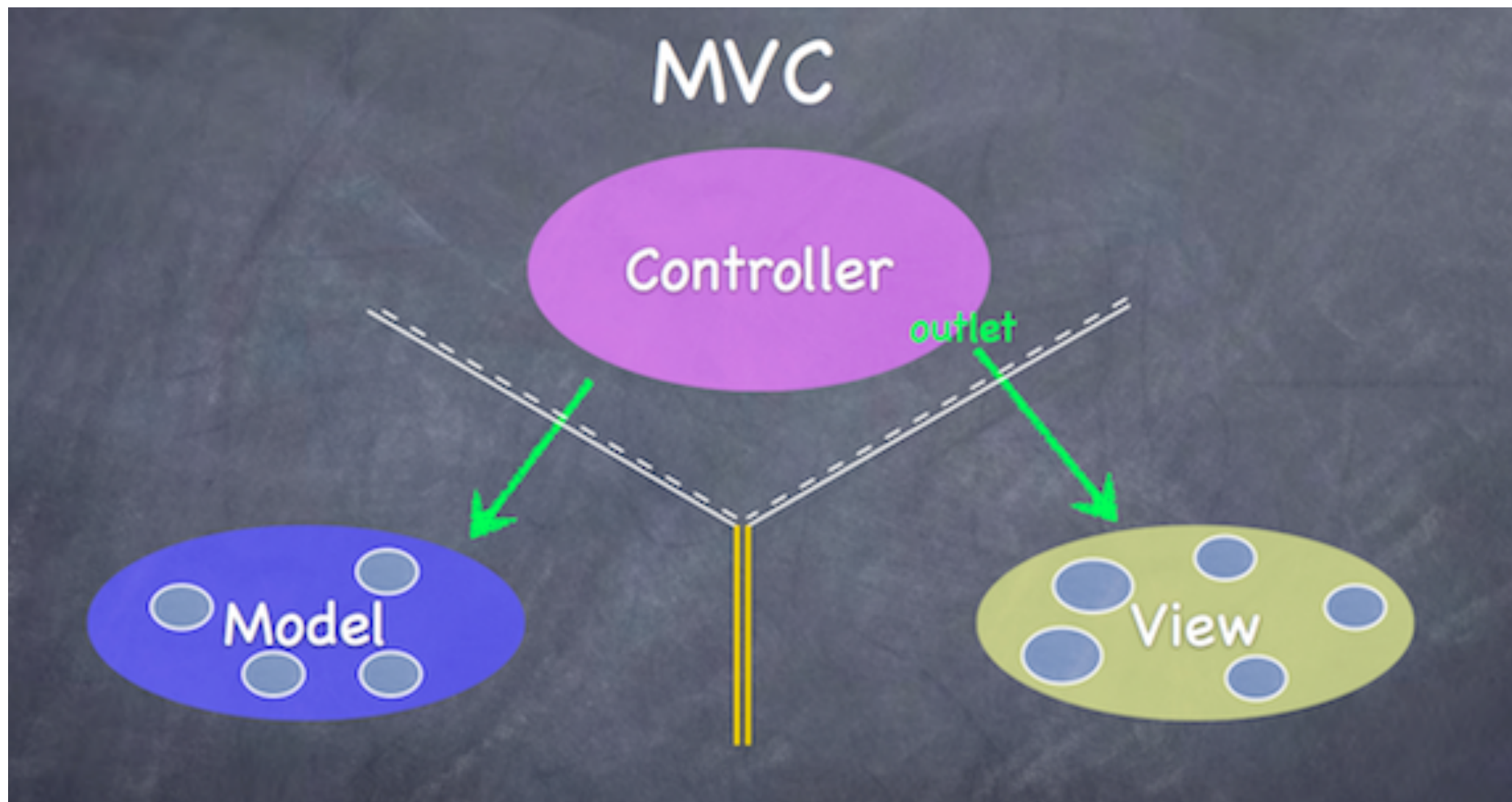
# View Controller

---

- 通常控制一整個App頁面
- App頁面的「大腦」
- 通常會有非常非常多行code...
- 真的有必要的時候可以把部分的東西切分出來到獨立的controller

# MVC團結！

---



(圖片引用自 Stanford CS193p)

# 溝通方式

---

- Outlet
- Target-action
- Protocol & Delegate
- Closure callback
- Notification & KVO



# Outlet

---

- Controller藉由outlet來對view物件進行動作
- Outlet通常會設為controller物件的屬性

# Target-action

---

- 當view物件發生特定事件(event)的時候會啟動
- Target-action會指定一個controller作為處理事件的target
- Controller上面的action是一個controller的方法，當controller接收到事件的時候會使用指定的action方法來處理
- IBAction就是target-action的一種

# Delegate

---

- 一個在Apple frameworks中很普遍的design pattern
- 一個物件如果有需要會指定一個delegate物件
- 物件透過一個protocol中定義好的方法來呼叫delegate物件
- Delegate物件需要實作protocol中定義的方法
- 常見命名方式: should\*, did\*, will\*

# Closure

---

- 有點類似其他語言的inline function, anonymous function, lambda function
- 通常當為callback使用
- 常常有提供delegate方式的API也會提供類似的closure callback方式

# Notification

---

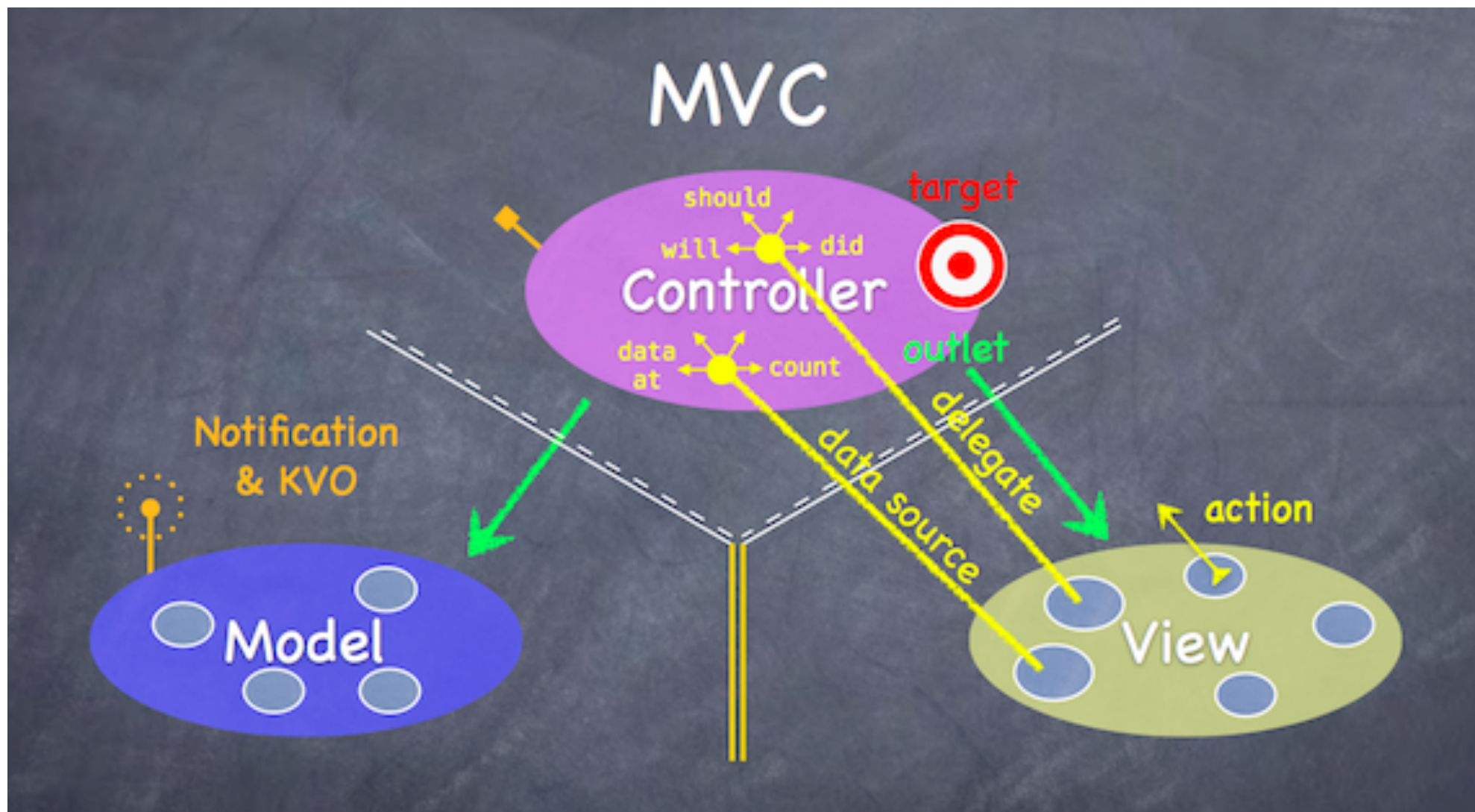
- 物件可以發出 NSNotification到 NotificationCenter
- 其他物件可以上NotificationCenter監聽他有興趣的 notification
- 當notification送出時，監聽的物件便可以動作
- 有點類似廣播跟收音機的概念

# KVO (Key value observing)

---

- 針對物件屬性改變的時候會發出的notification
- 修改物件屬性的時候，監聽的物件便可以做出動作

# MVC互相溝通



(圖片引用自 Stanford CS193p)