Model View Controller

Advanced Topics in iOS & Swift
11/02/2016

Name things that you are planning to do to level up as a programmer in this quarter.

Think, pair, share.

New quarter - new chances 🤓

- pay attention and **participate in class**
- **finish challenges** started in class (if possible, by yourself, even if that means you’re getting stuck on a problem for a longer time)
- **ask for help** and come to office hours
- establish daily/**regular coding habits**
- don’t wait for the perfect side-project, **start small** (can be a spec project, extension/modification of challenges)
- be **diligent**, reflect on the code you write
- pay attention to **code formatting** and **naming**

What is MVC?

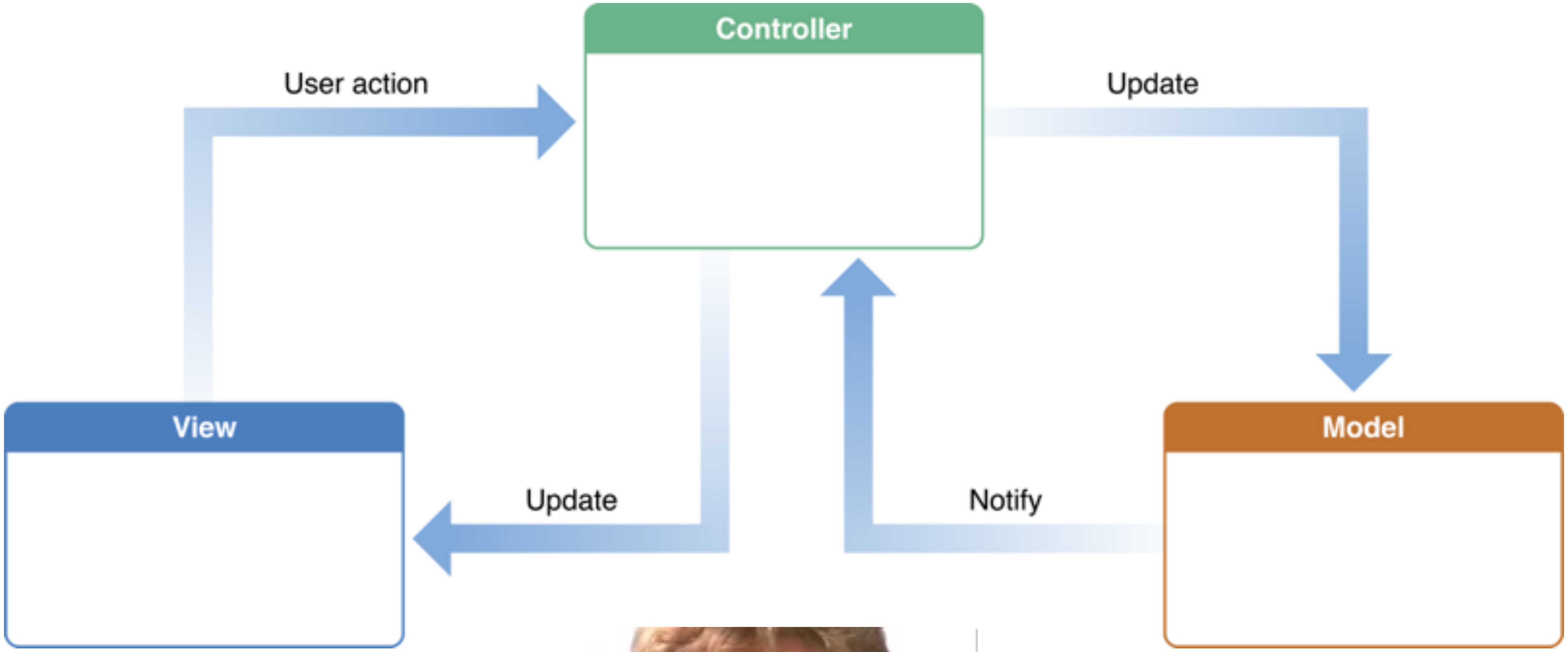
Why do we use it?

What do you know about the responsibilities of its components (Model, View and Controller)?

Think, pair, share.

Information Flow in the MVC pattern

🤔



Separation of Concerns
Model

stores **domain objects** and encodes domain rules (also called business objects/rules)

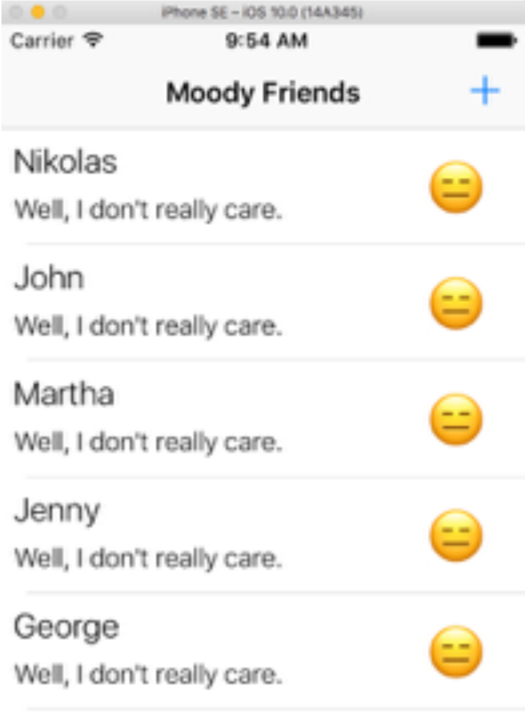
represents the **data** in our application

Separation of Concerns
View

displays information on the screen

accepts **touch input** (which it delegates to the controller)

does not “store” information



Moody Friends

Nikolas
Well, I don't really care. 😑

John
Well, I don't really care. 😑

Martha
Well, I don't really care. 😑

Jenny
Well, I don't really care. 😑

George
Well, I don't really care. 😑

Separation of Concerns
Controller

“glue” that **helps model and view communicate**

receives info about touch input from view

receives info about changes to data from model

updates model

updates view

Getting real…

- Creating a small *MoodTracker* app that displays a list of friends in a table view

- each friend has a **`mood`** that can be toggled with a tap on the button that displays the mood symbol (😁😑😡)

- **Goal:** implement a *proper information flow* according to the MVC pattern

- Note: you'll learn about usage of enums and different communication patterns along the way

New quarter, new chances!

- if you don’t fully grasp a concept that is considered being part of the fundamentals, make sure and review it outside of class
    - creating a class with a number of properties
    - instantiating that class and passing values for the properties in the initializer
    - subclassing
    - populating a table view
    - reading text input using text fields