

Lecture 2

Project Memorize

Arquitectura MVVM

El Modelo: (MemoryGame.swift)

Encapsula la data y la lógica y es SIEMPRE una class

- Son las tarjetas que son considerados como la data
- La lógica es que pasa cuando selecciono una tarjeta, como hago match, cuantos puntos obtengo cuando hago match, que pasa si no hago match, estos son algunos ejemplos

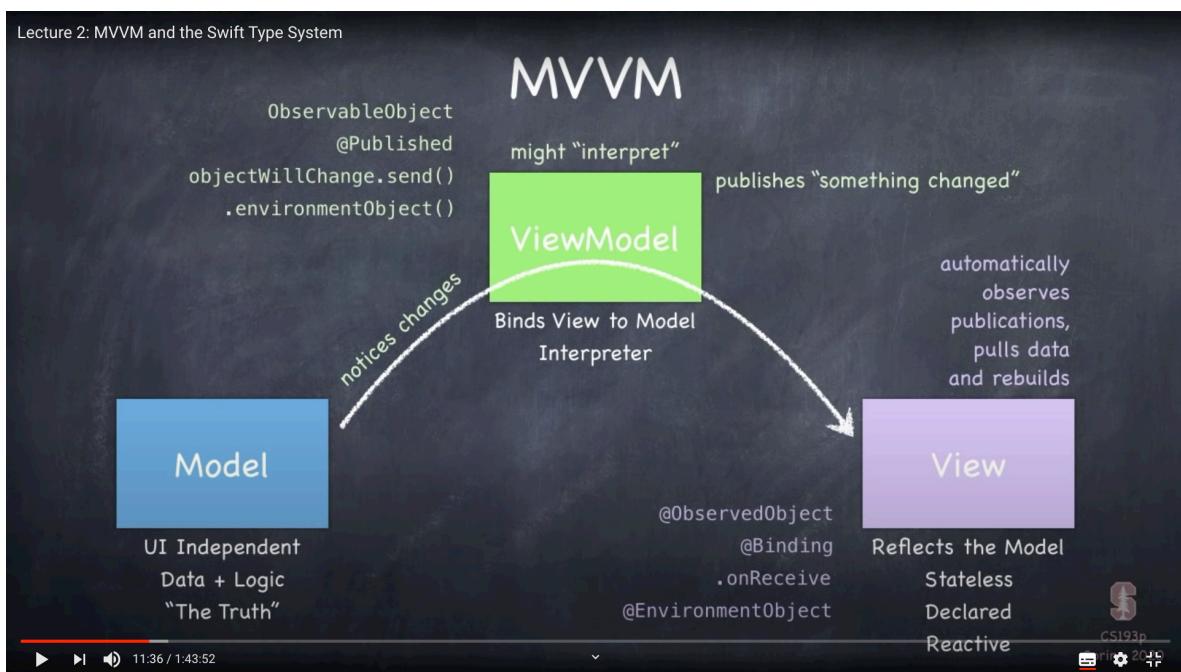
La Vista:

Refleja el modelo, no tiene estado ya que el estado del juego va a estar en todo momento en el modelo, la vista es declarativa, cambia solo cuando el modelo cambia. La vista tiene que ser reactiva, lo que quiere decir es que cada vez que el modelo cambie la vista debe cambiar automáticamente.

El View Model: (EmojiMemoryGame.swift)

Es el "traductor" entre el modelo y la vista, hace que cuando un cambio en el modelo ocurre, se vea reflejado en la vista automáticamente. Le hace el favor a la vista de interpretar al modelo. Por ejemplo es el encargado de hacer conversiones de data type.

El view model NUNCA tiene un puntero hacia su vista, no habla de forma directa con su vista, cuando algo cambia en el modelo solo publica que algo ha cambiado y la vista se suscribe a esas publicaciones y cuando una de esas publicaciones ocurre va a preguntar al View Model cual es el estado del mundo en ese preciso momento para pintarse de tal forma que represente el estado actual del mundo



En el sentido contrario, por ejemplo cuando hay una interacción de usuario con un botón que se encuentra en la vista, se agrega una nueva responsabilidad al ViewModel que son los INTENTS. Por ejemplo en el proyecto actual de memory card un intent es cuando el usuario selecciona una tarjeta. Al momento que se de un intent en la vista se le comunica al ViewModel y este se encarga de modificar el modelo.

