Tema 3: Disseny

Anna Puig

Enginyeria Informàtica

Facultat de Matemàtiques i Informàtica,

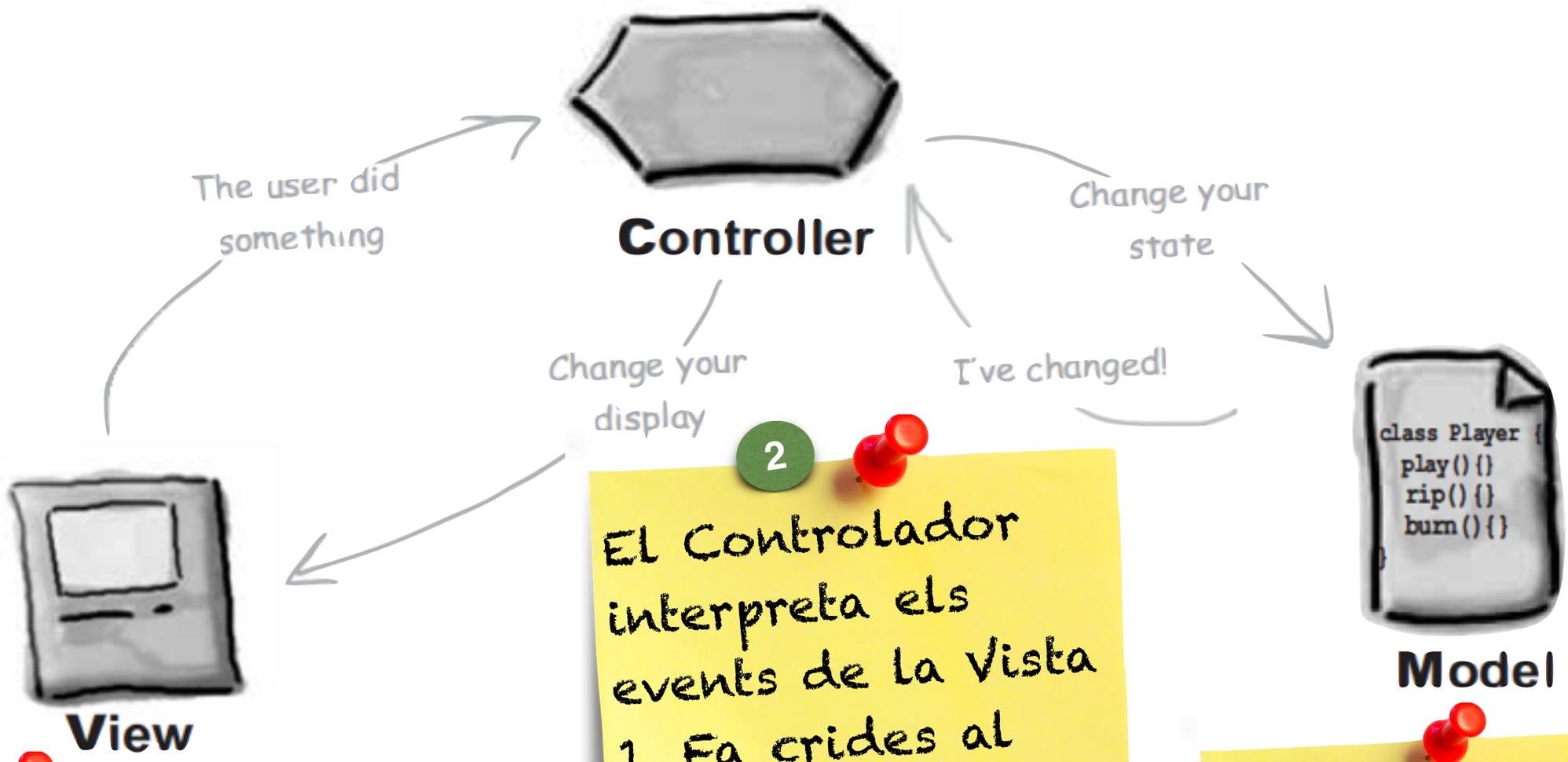
Universitat de Barcelona

Curs 2021/2022

UNIVERSITAT DE BARCELONA

Temari

Model-Vista-Controlador
Aproximació senzilla (pràctica 3)

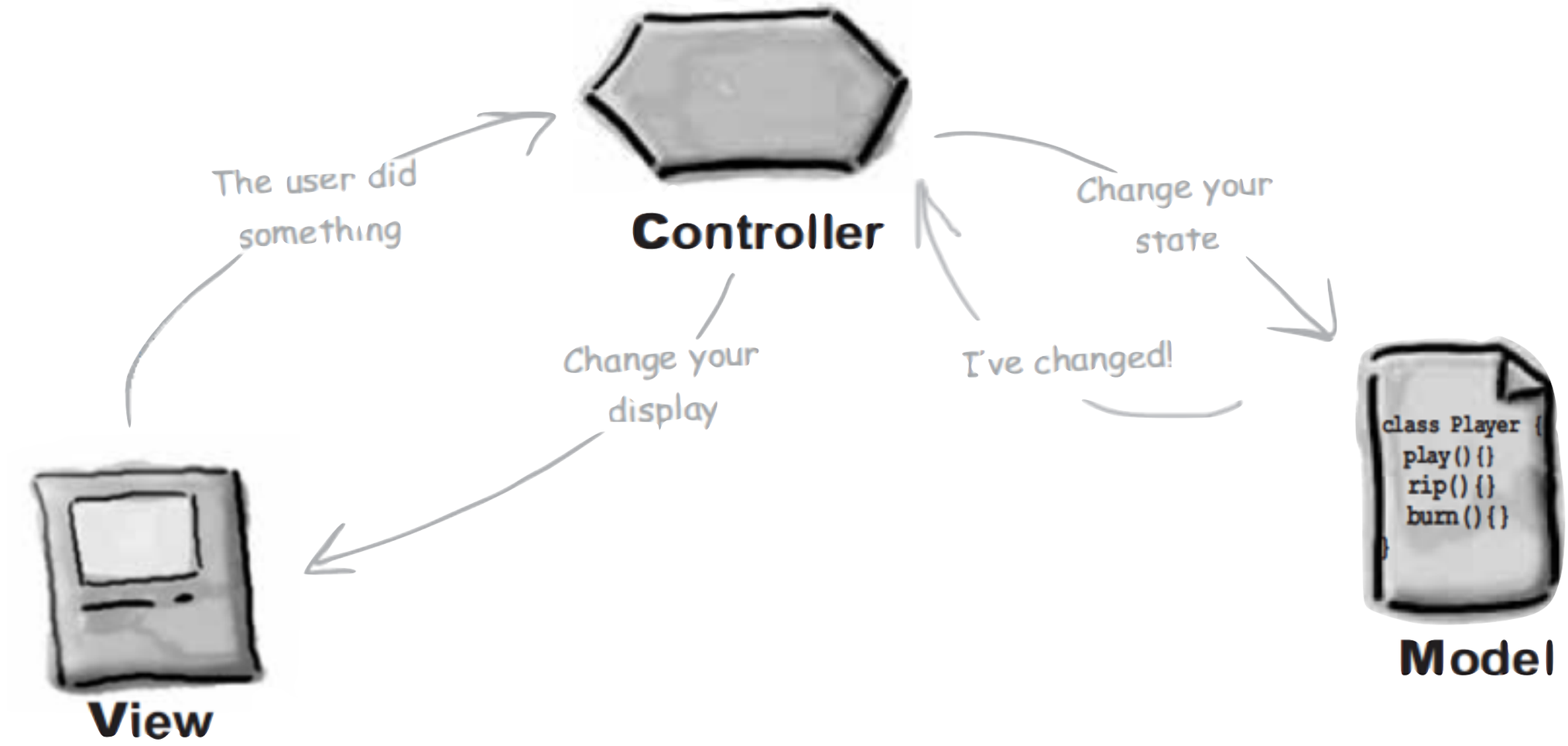


1. La Vista captura l'input de l'usuari i mostra l'estat del model

2. El Controlador interpreta els events de la Vista
   1. Fa crides al Model
   2. Fa crides a la Vista

3. El Model canvia el seu estat i ho notifica al Controlador

Model-Vista-Controlador
Aproximació senzilla (pràctica 3)



Problema:

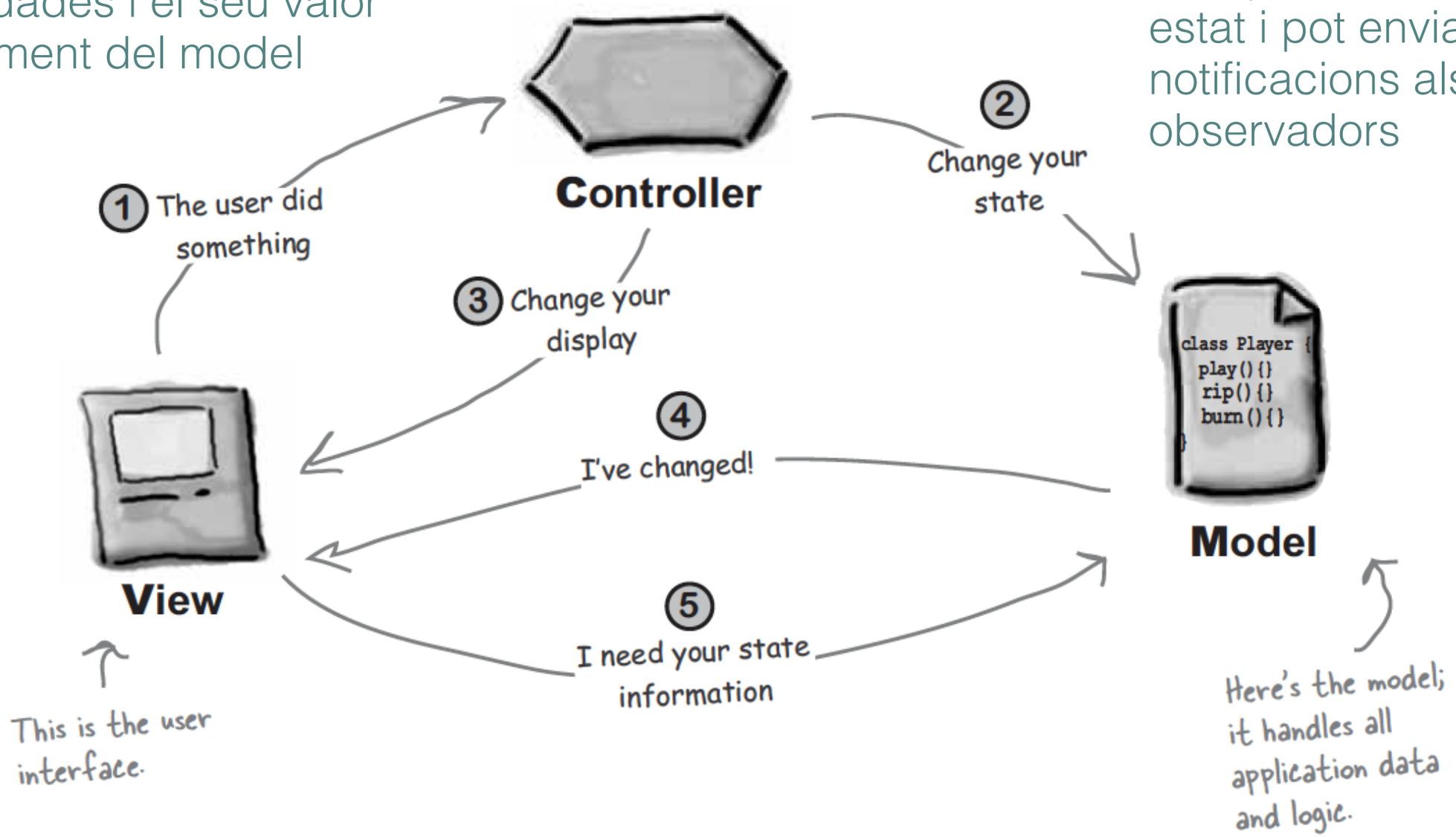
- Quan hi han actualitzacions independents en el model que no estan fetes des de la vista, com pot el model notificar aquests canvis a la Vista?
- El Controlador centralitza tota la comunicació produïnt retards a vegades innecessaris.

Model-Vista-Controlador
Aproximació general (pràctica 4)

VISTA:
Dóna la presentació del model. La vista normalment mostra l'estat de les dades i el seu valor directament del model

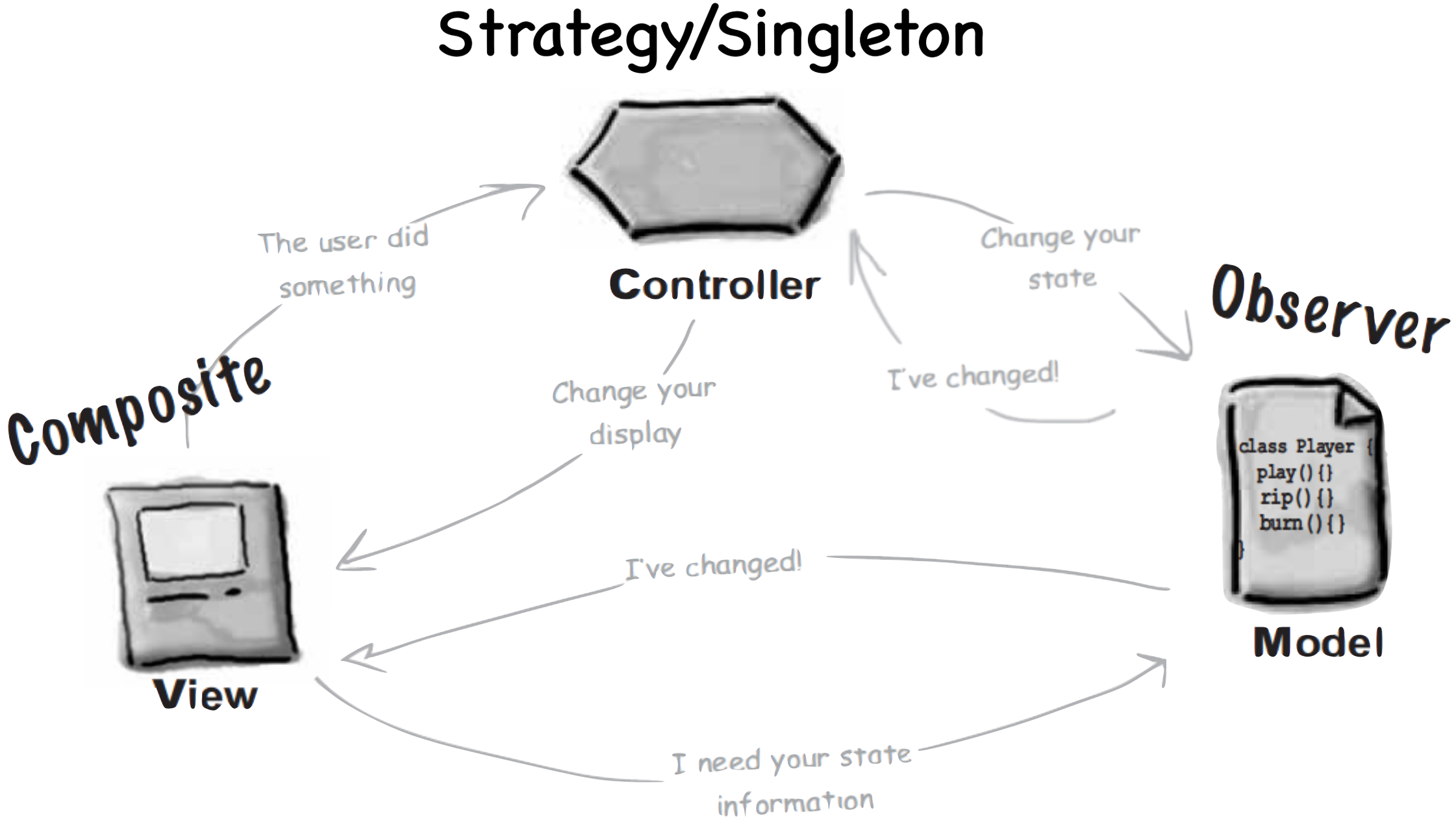
CONTROLADOR:
Agafa l'entrada de l'usuari i li dóna el què significa al model i actualitza la vista.

MODEL:
El model guarda totes les dates, l'estat i la lògica de l'aplicació. Dóna una interfície per manipular i donar el seu estat i pot enviar notificacions als observadors



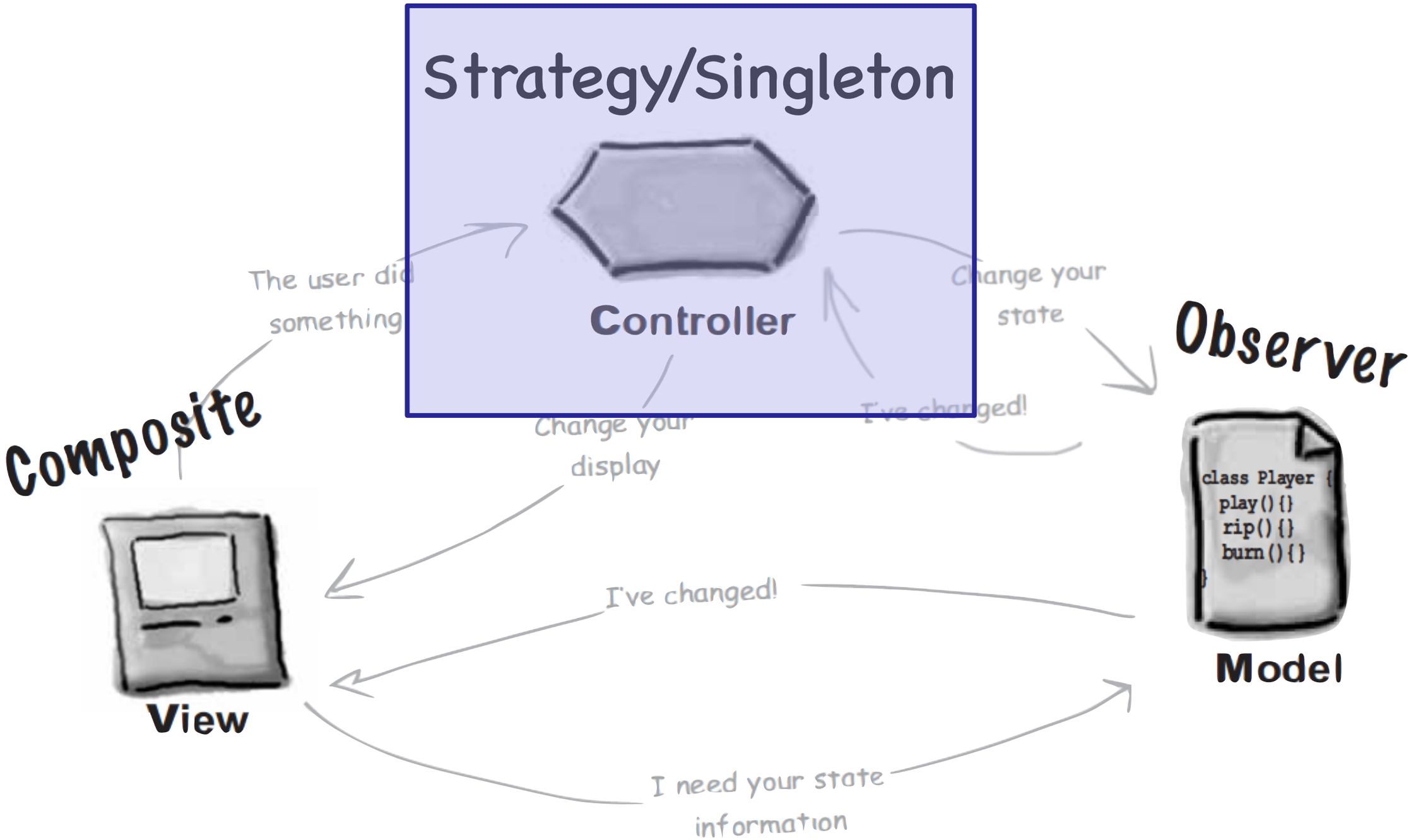
Model-Vista-Controlador

Aproximació general - Principals patrons



Model-Vista-Controlador

Aproximació general - Principals patrons

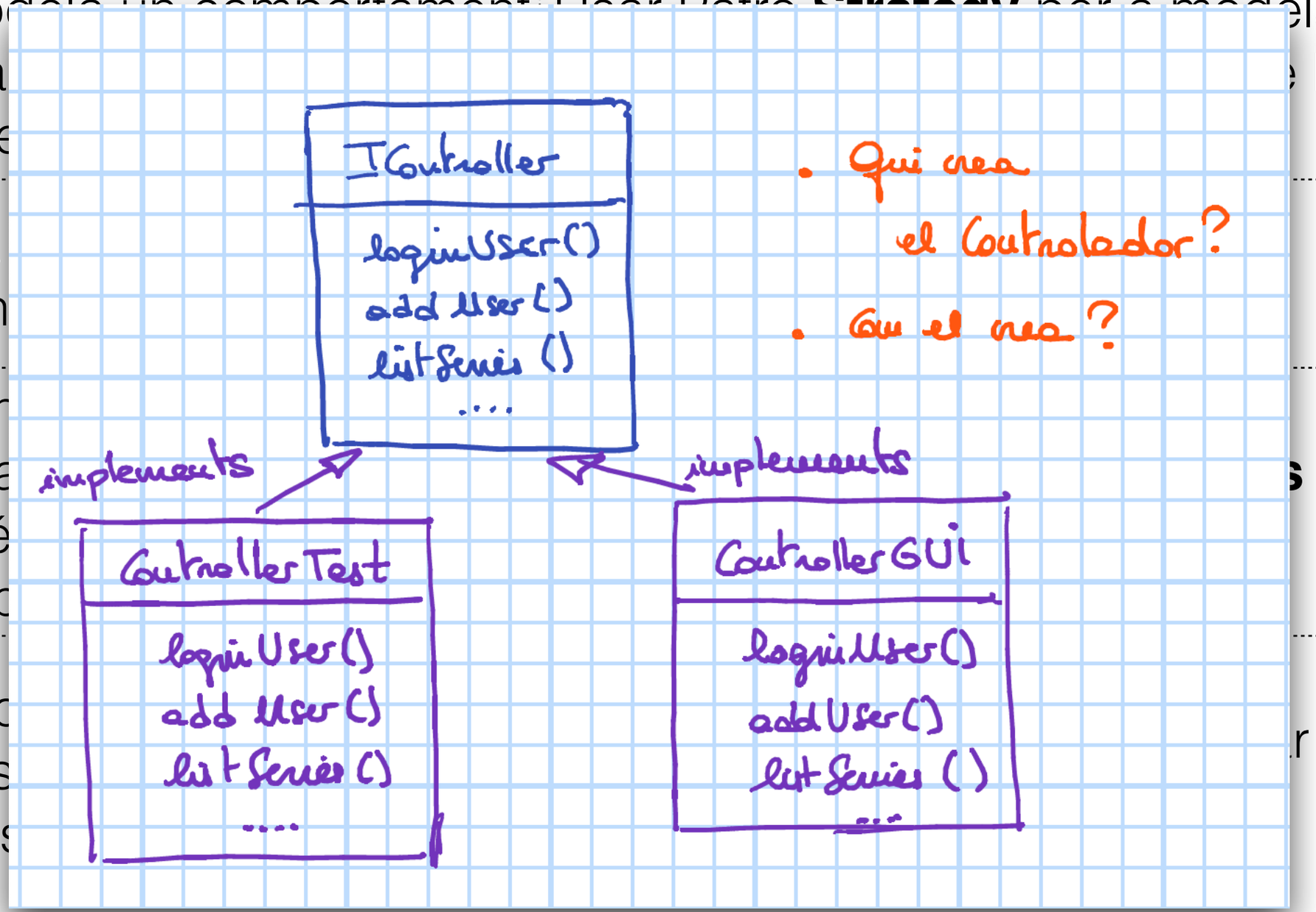


Controlador

Problemes	Possibles solucions
Un controlador modela un comportament concret de la Vista, que pot canviar en el temps	Usar Patró **Strategy** per a modelar els diferents comportaments de la Vista
Només es vol una instància de la classe Controlador	Patró **Singleton**
Hi ha un únic controller rebent tots els events del sistema i en són molts (baixa cohesió). Esdevé un oracle que tot ho sap o objecte "déu".	Utilitzar diferents **controladors** i diferents **Façanes** del model
Un controlador manté atributs i informació d'altres objectes o duplica la informació que es troba en altres llocs	Usar Patró **Expert** per delegar responsabilitats

Controlador

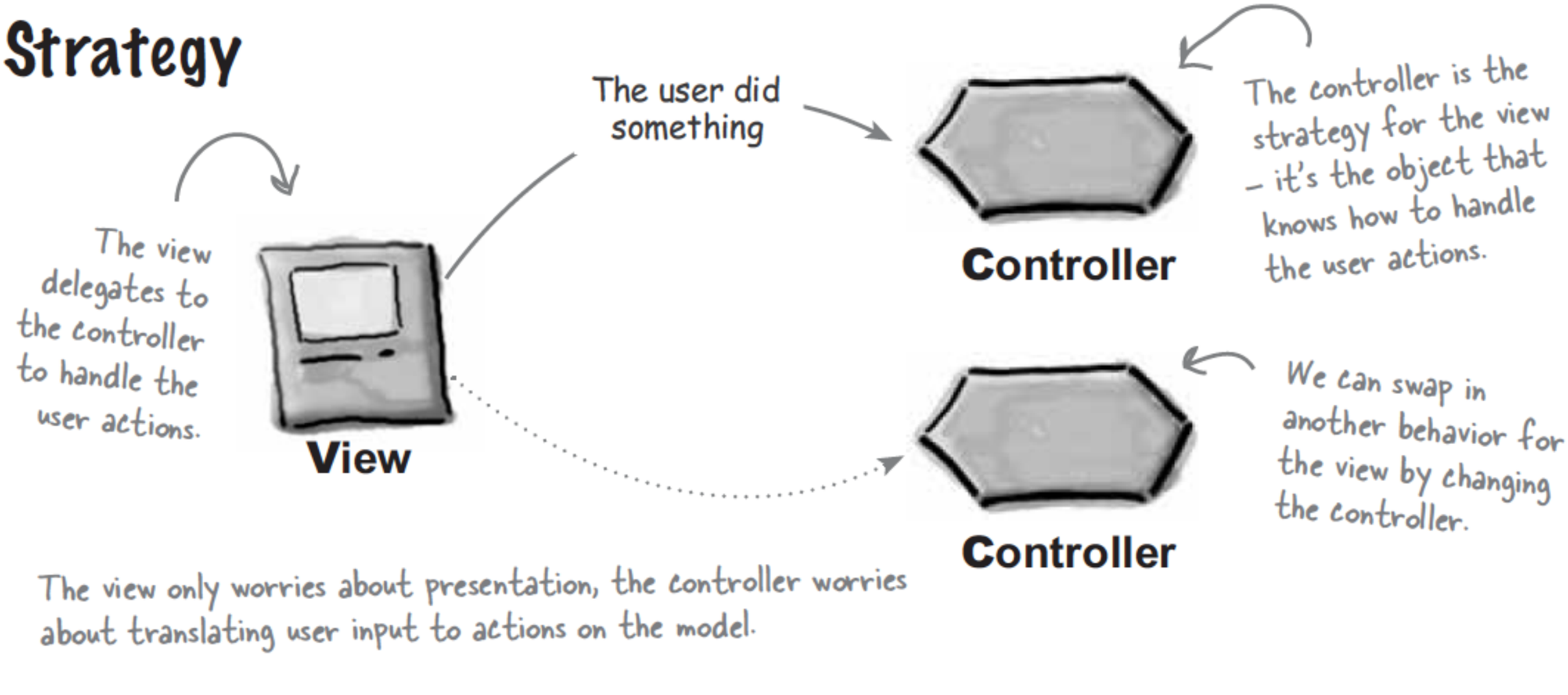
Problemes	Possibles solucions
Un controlador mo concret de la Vista te	Usar Patró **Strategy** per a modelar la
Només es vol una Con	
Hi ha un únic cor events del sistema cohesió). Esdevé sap o ob	**s** i
Un controlado informació d'altres informació que es	r



Model-Vista-Controlador

Patrons en el Controlador:

- **Strategy**
- Singleton



Model-Vista-Controlador

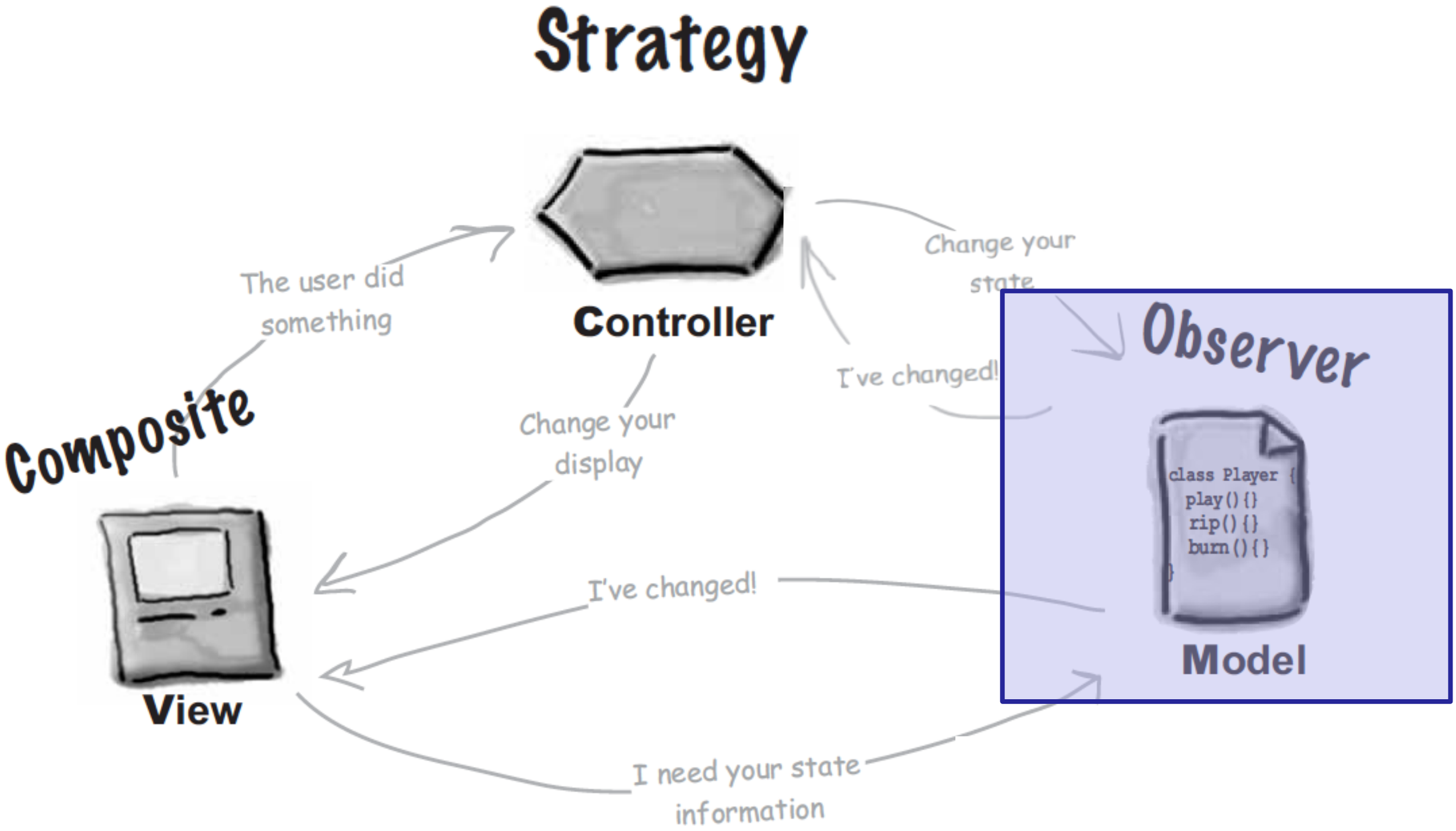
Patrons en el Controlador:

- Strategy
- **Singleton**

- El Controlador manega els estats de la Vista
- Cal assegurar que només hi ha una única sola instància del controlador i cal proporcionar un punt d'accés global a ella
- Interessa fer la instància de la classe només quan faci falta (*lazy instantiation*)

Model-Vista-Controlador

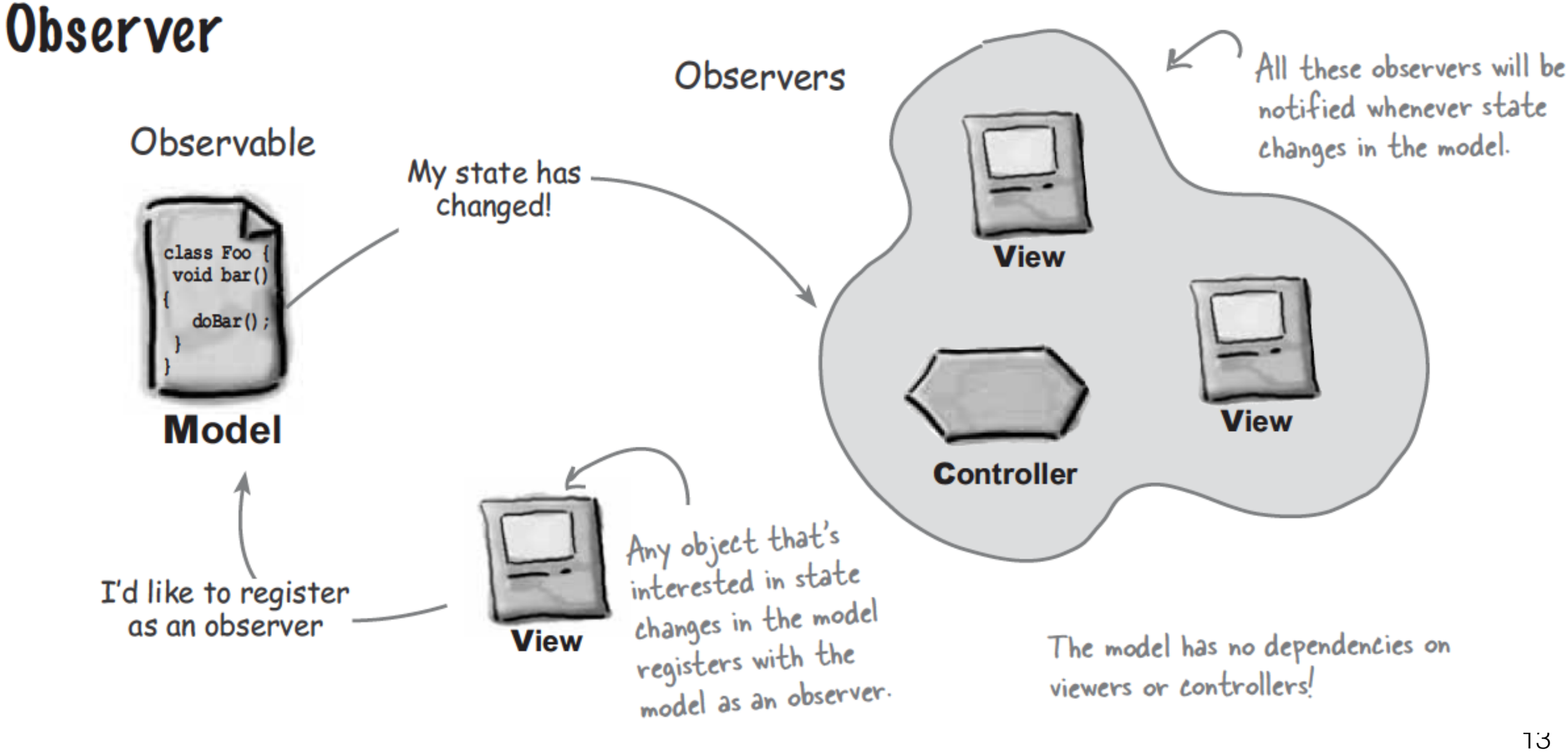
Aproximació general



Model-Vista-Controlador

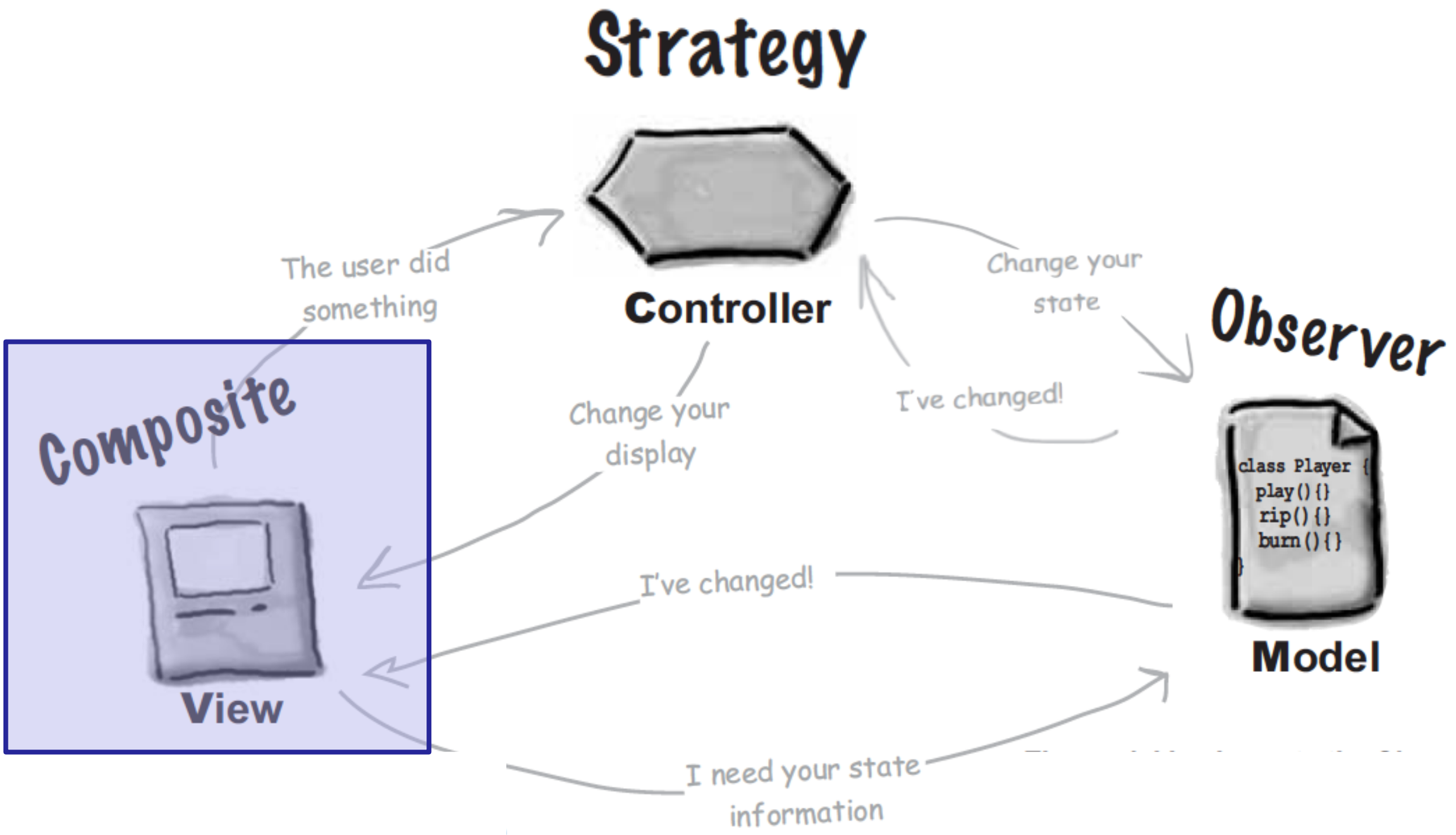
Patró en el Model:

- **Observer**



Model-Vista-Controlador

Aproximació general

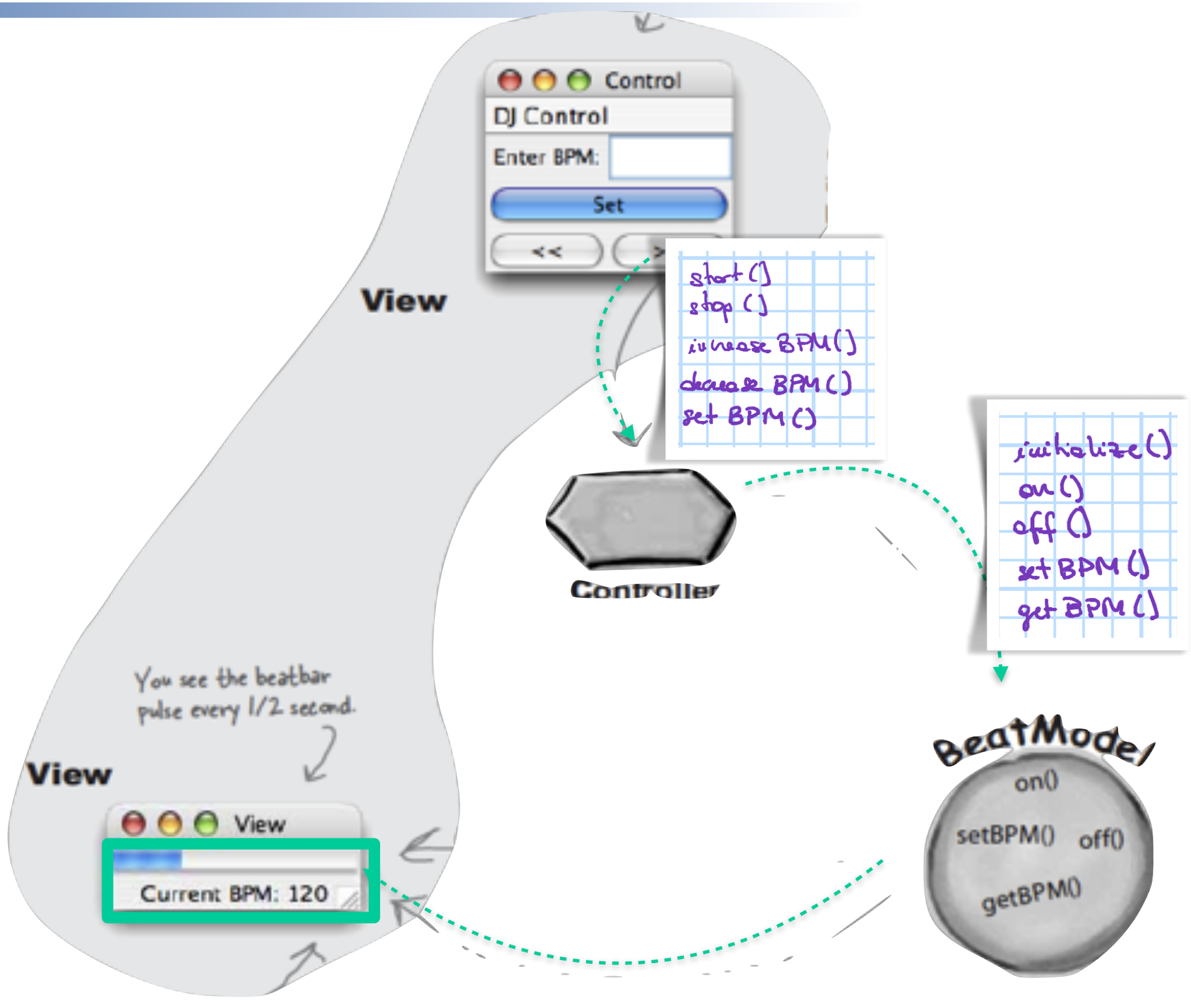


Exemple: Usant el MVC per controlar el ritme

- Objectiu: Obre el projecte del campus i aprén a aplicar el patró Observer

Fer una aplicació per a que un DJ pugui controlar el ritme (Beats Per Minute - BPM) de la música

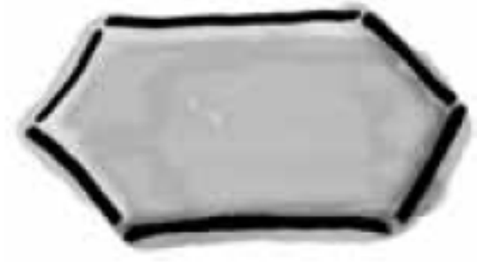
Model Vista Controlador



Com es controla la Vista

De la Vista al Controller

Totes aquestes accions s'envien al controlador



Controller

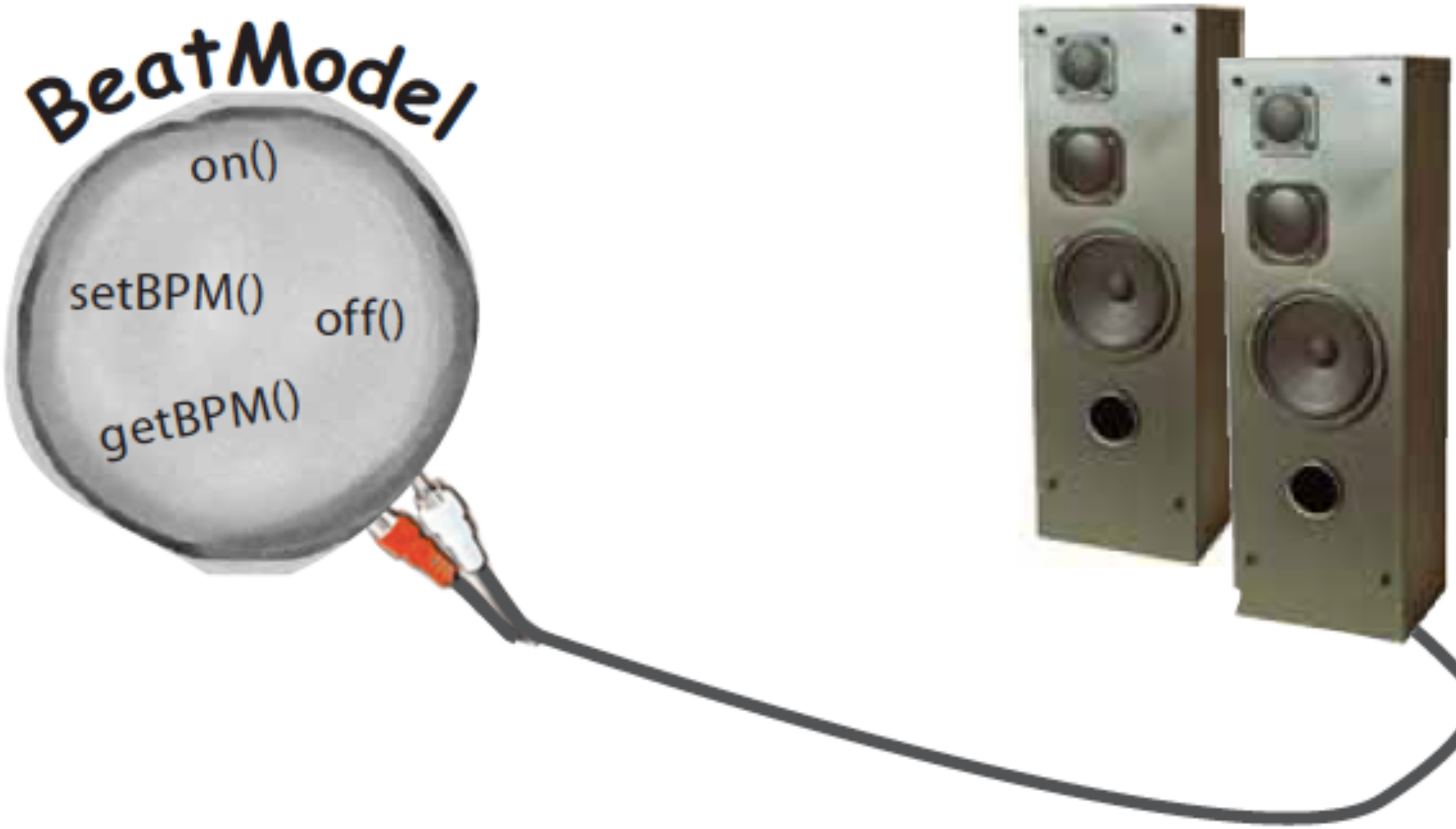
Vista

La vista té dues parts:

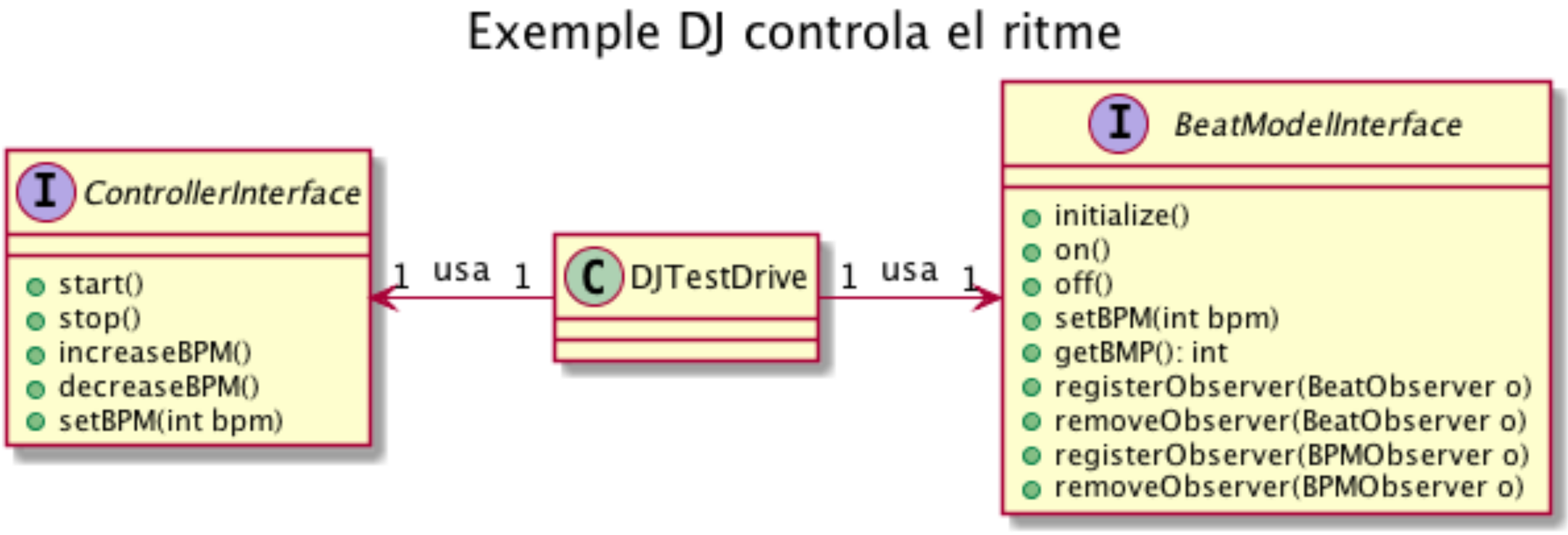
Part per veure **l'estat** del model

Part per controlar Beats per Minut

Utilitats que dóna el model

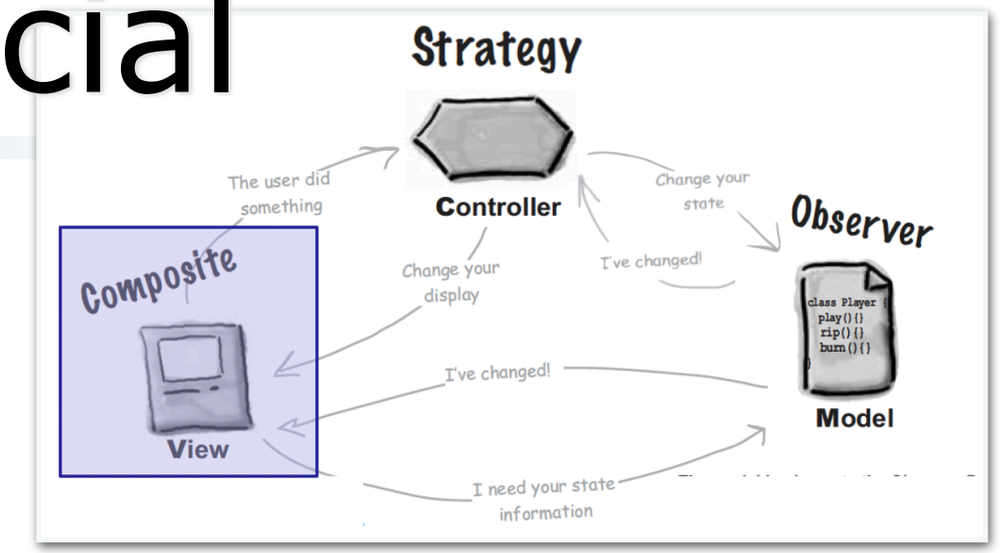


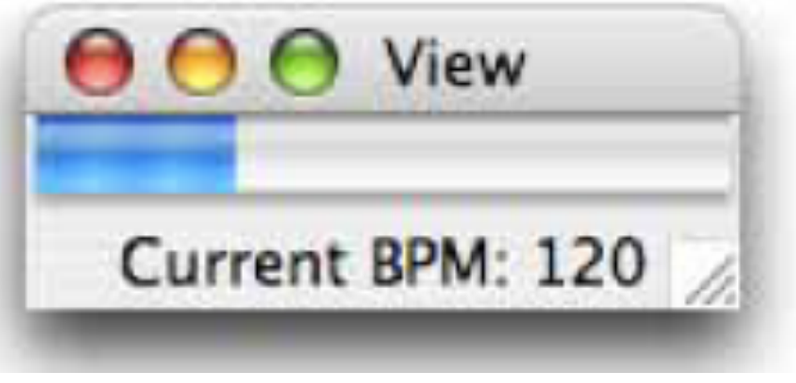
Classe DJTestDrive

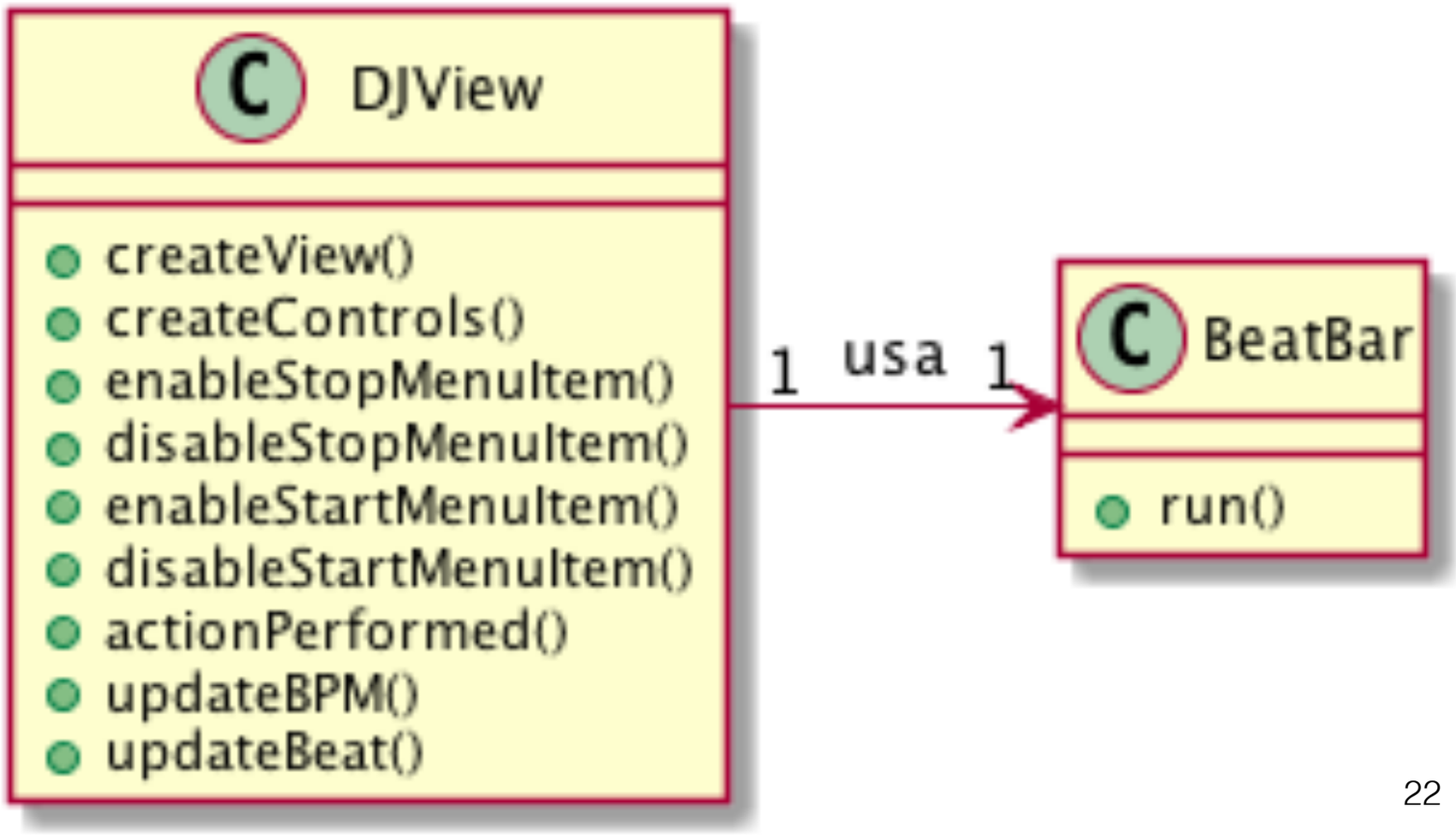


Qui crea qui?
Qui crea la Vista?
Acoblament? Patró expert?

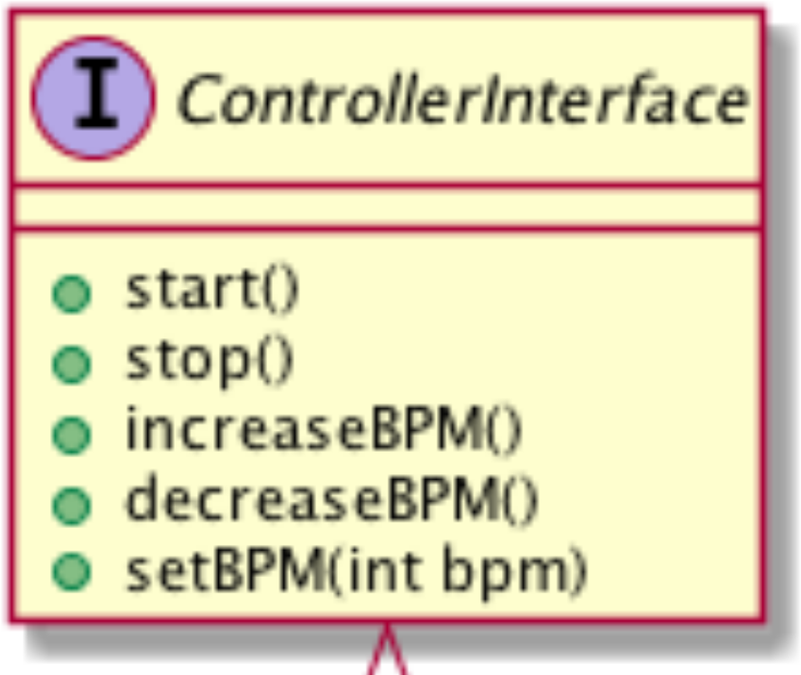
DJView inicial

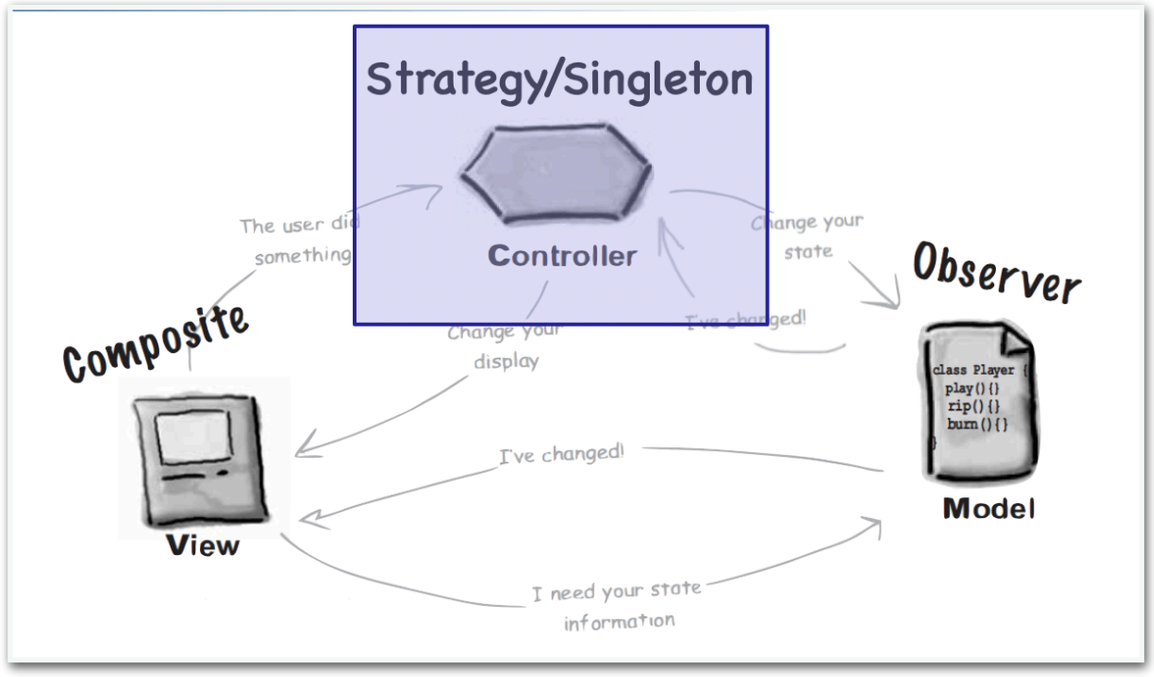


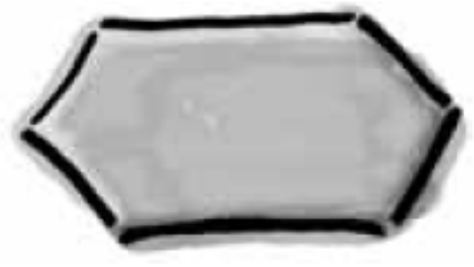




Controller bàsic

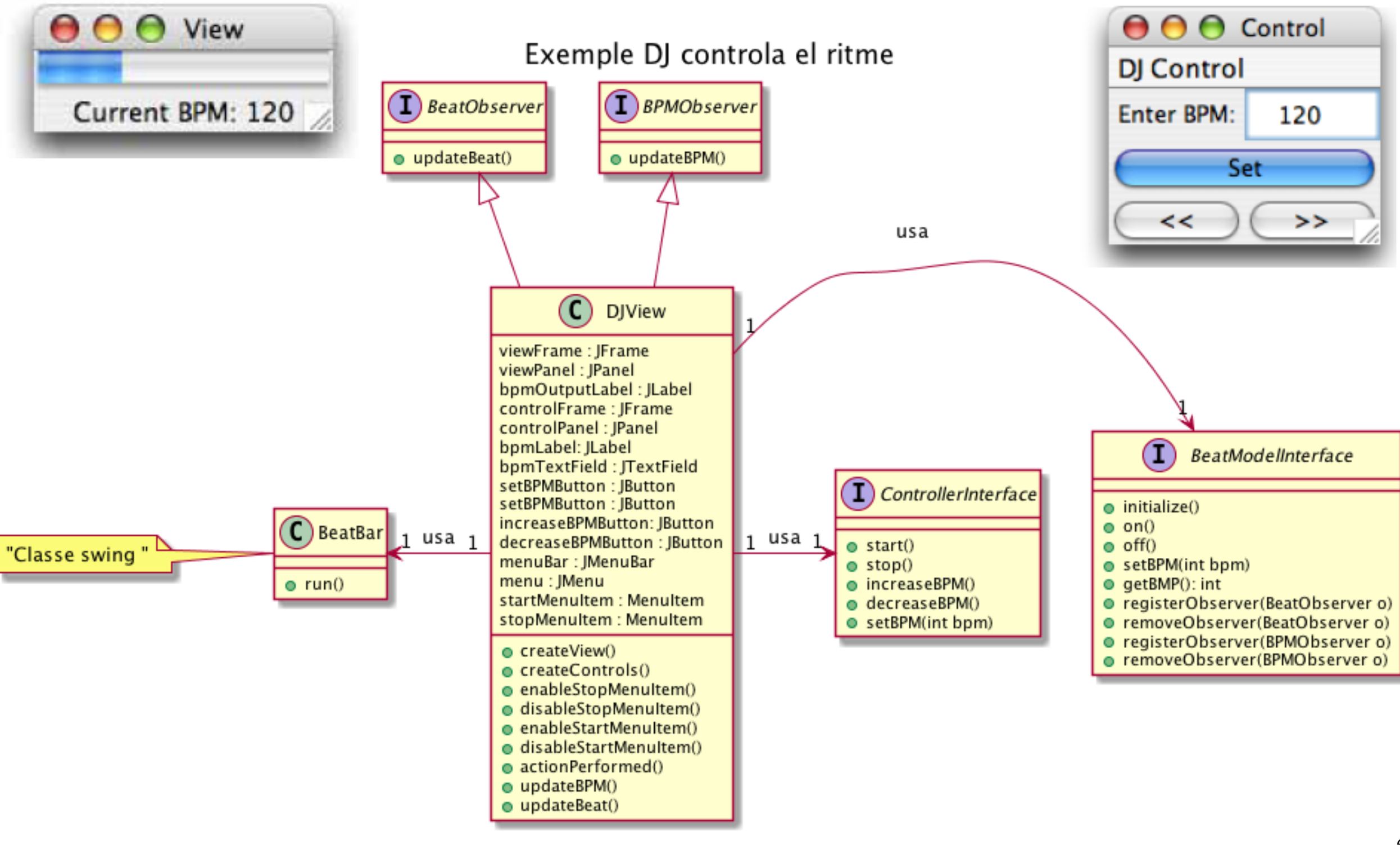




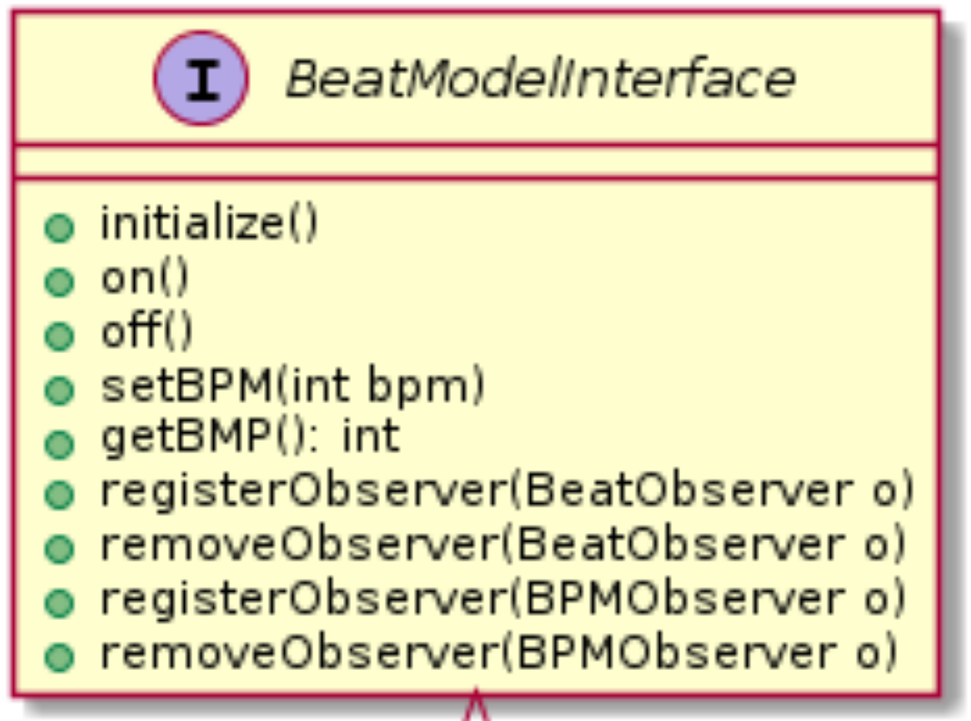


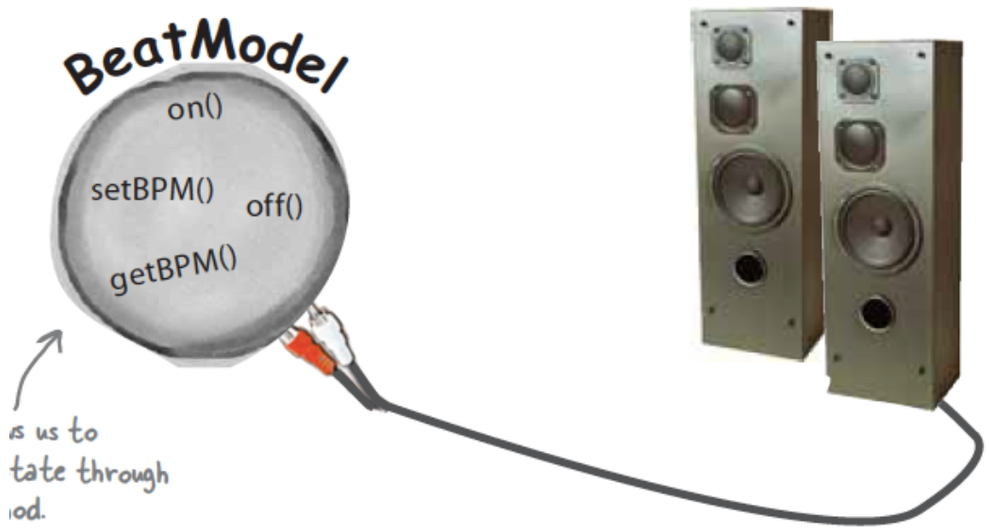
Controller

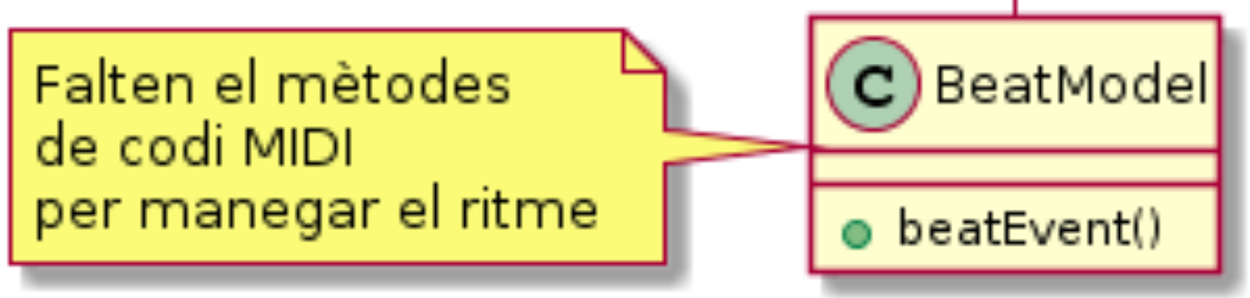
Vista amb més detall



Model bàsic

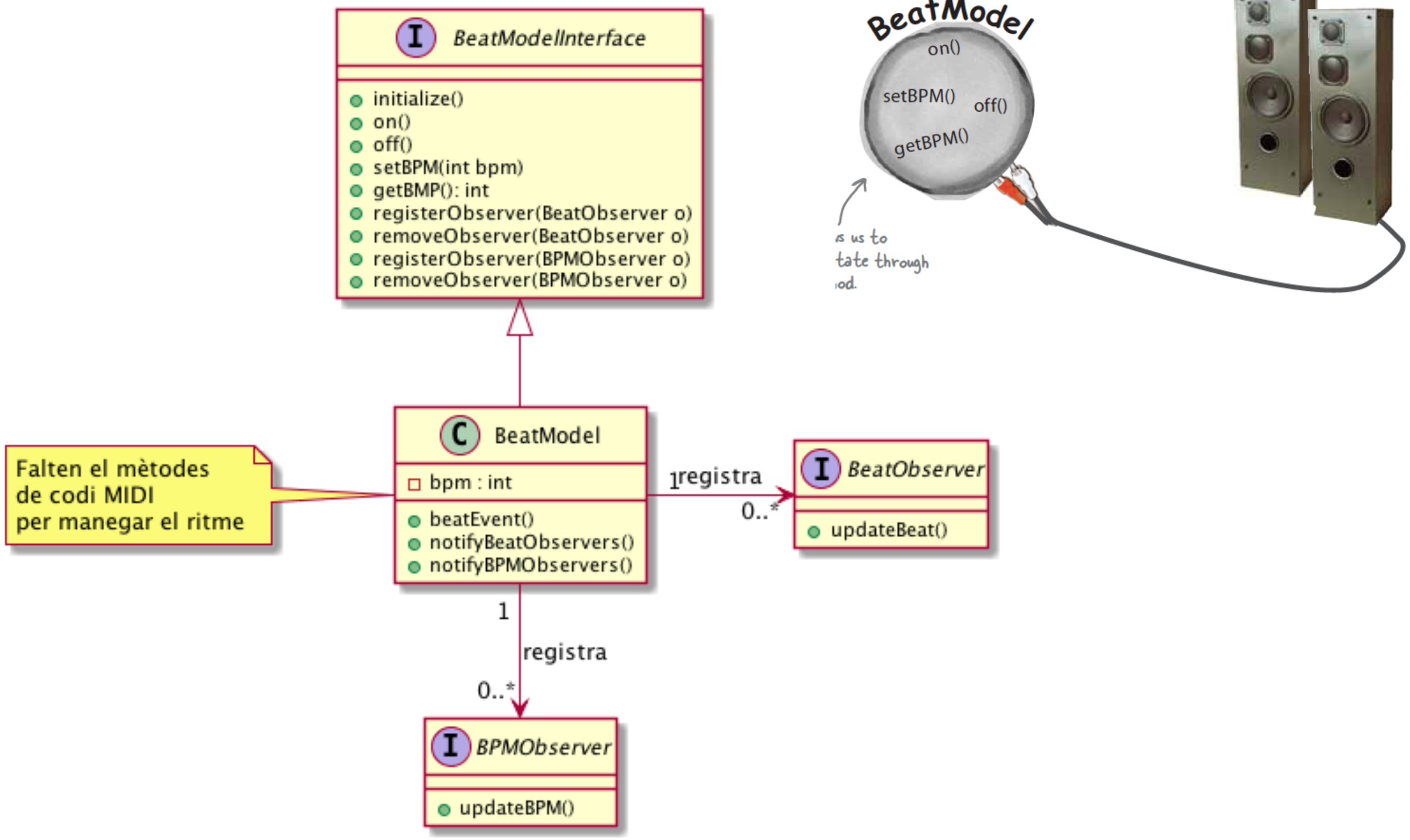

BeatModelInterface
initialize()
on()
off()
setBPM(int bpm)
getBMP(): int
registerObserver(BeatObserver o)
removeObserver(BeatObserver o)
registerObserver(BPMObserver o)
removeObserver(BPMObserver o)


BeatModel
on()
setBPM()
off()
getBPM()

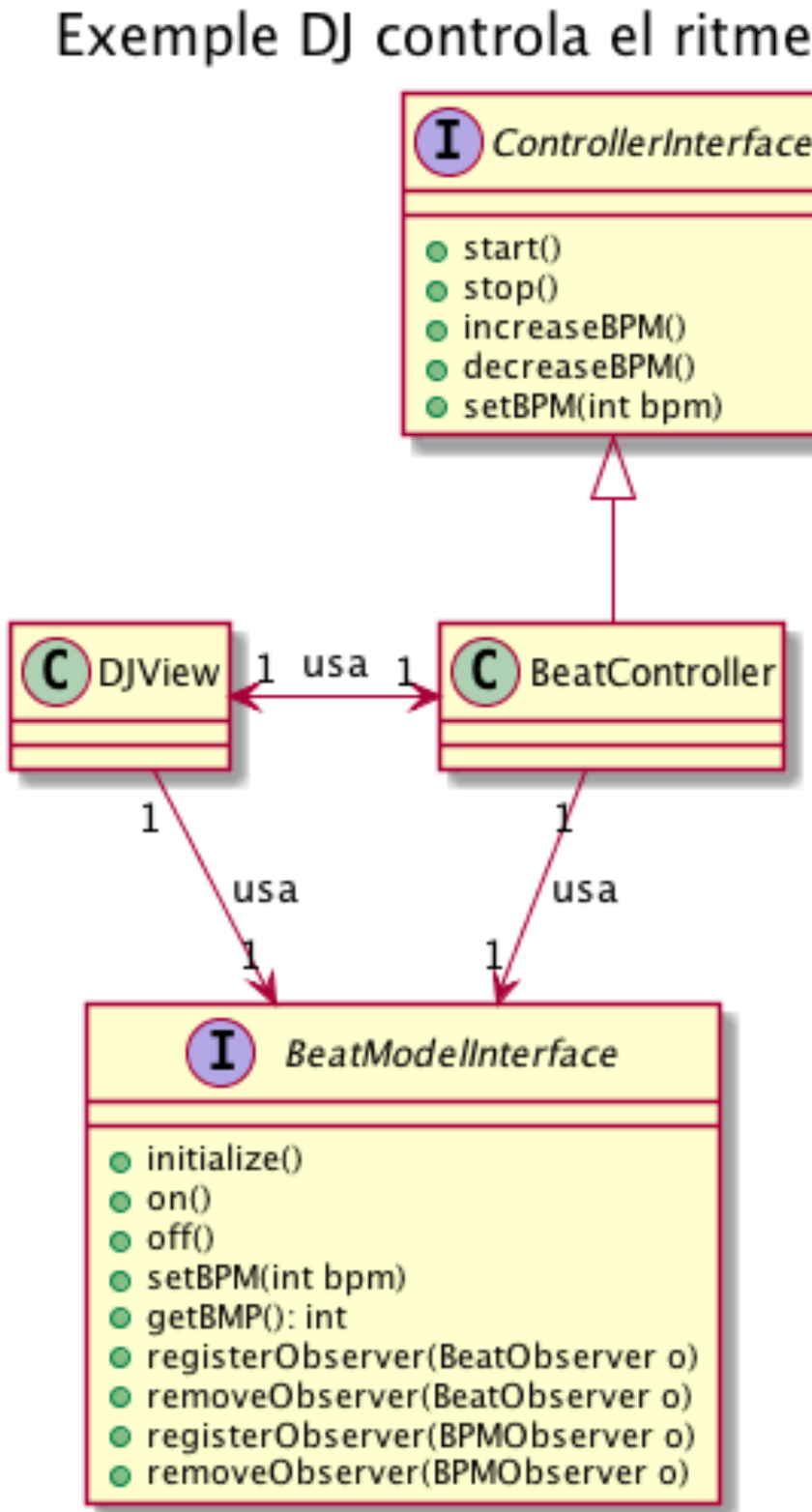

Falten el mètodes
de codi MIDI
per manegar el ritme
BeatModel
beatEvent()

Model amb més detall

Exemple DJ controla el ritme

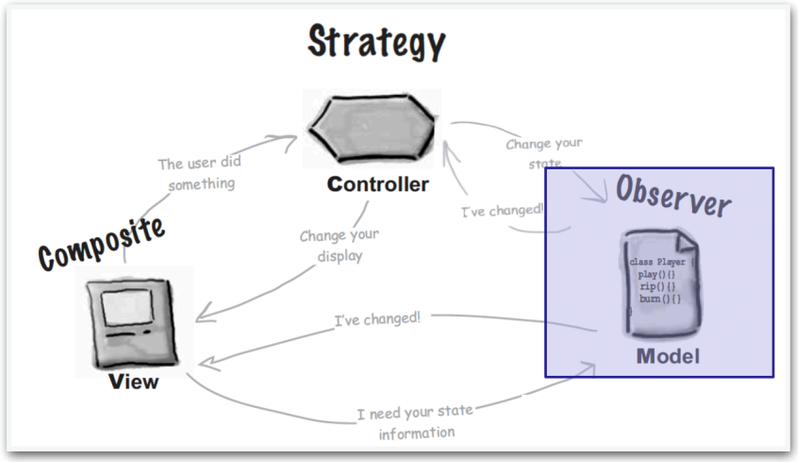


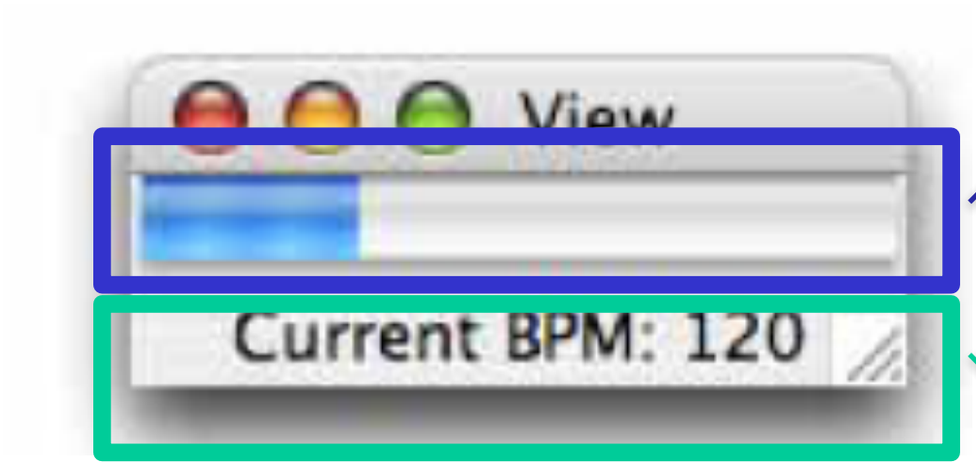
Controller amb més detall



Què es vol observar del model?

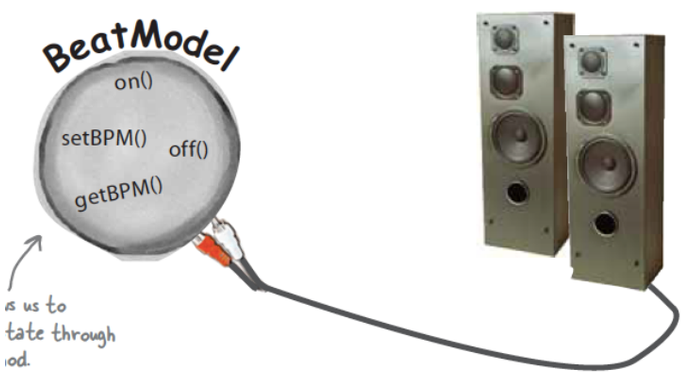
La vista de l'estat del model té dues parts:





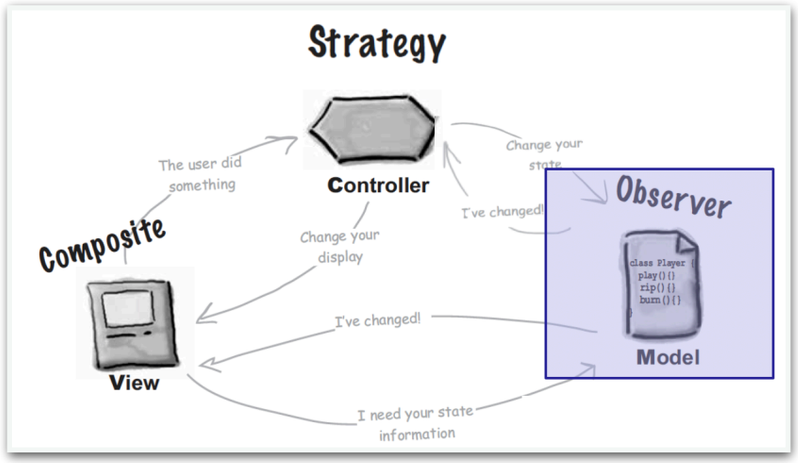
La part que es refresca a cada "beat"

La part que es refresca quan canvia el valor de BPM

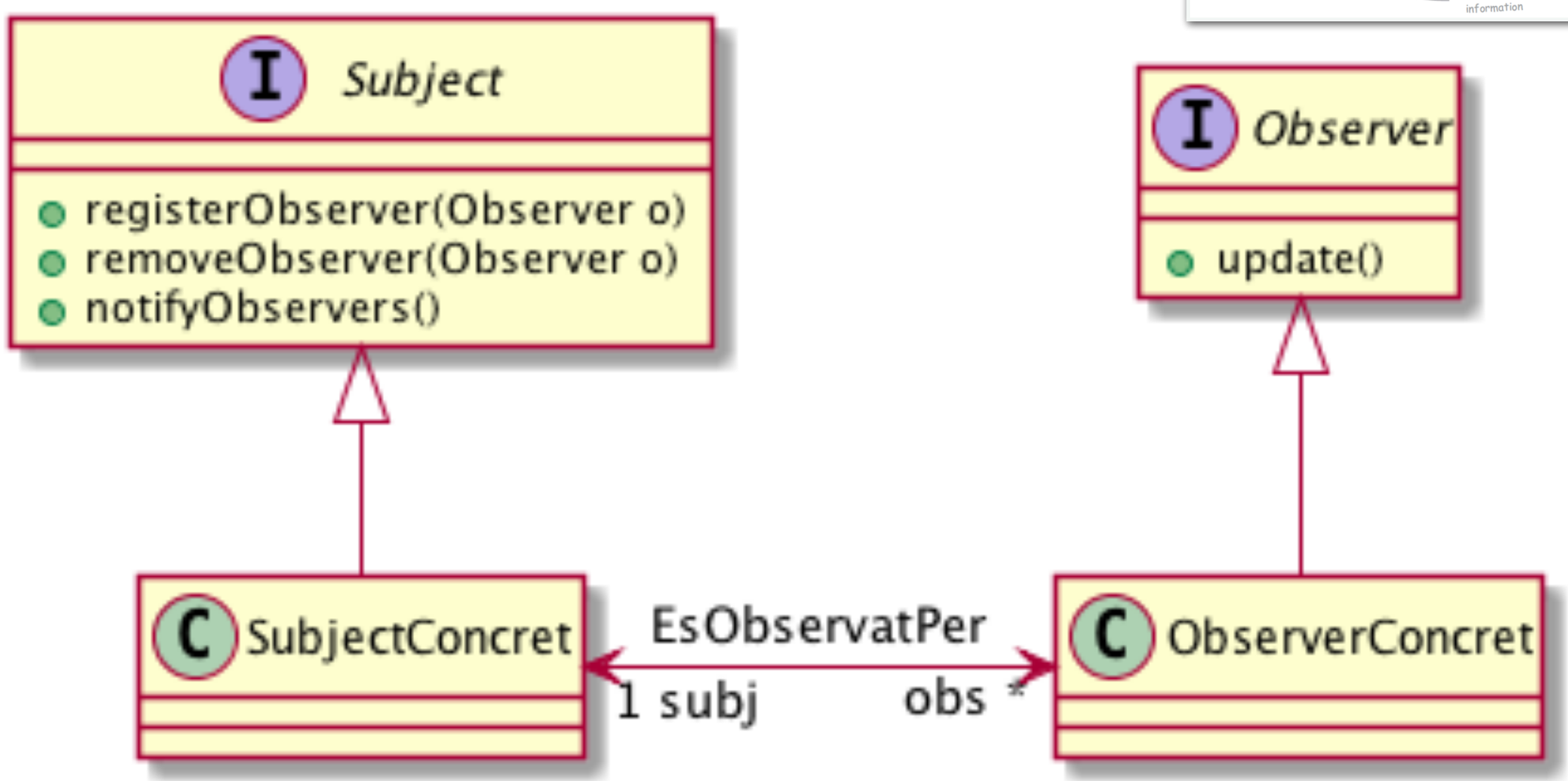


Això porta a observar dues informacions diferents: BeatObserver i BPMObserver

Patró Observer



Observer



Exercici:

1. Obre el projecte BeatApplication del Campus.
2. Aplica el patró Observer per a refrescar la Vista View:
   - Qui és l'observador i qui és el subject?