CountOnMe

Présentation

Lauriane Haydari
2019

RAPPEL DU CONTEXTE :

Améliorations :

- Le design et le responsive en portrait uniquement
- L’architecture du projet
- Les tests unitaires
- Il manque la division et la multiplication.

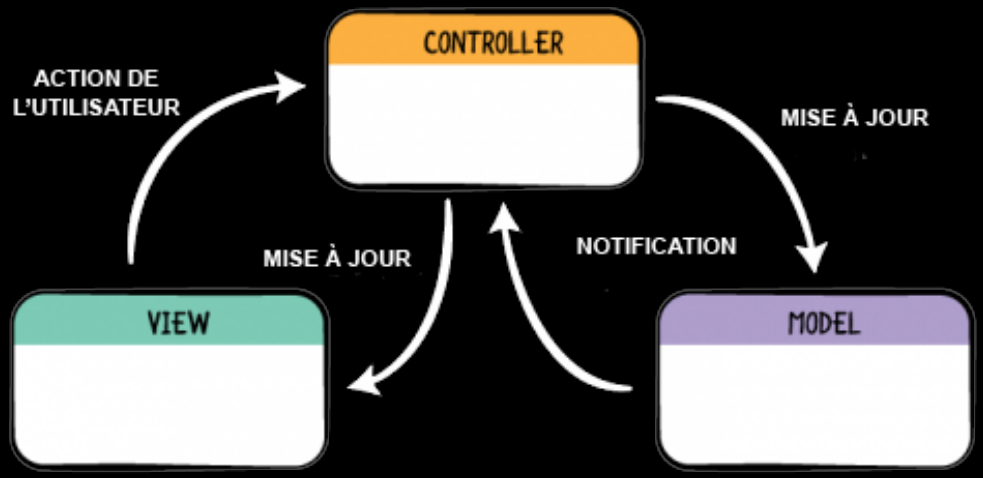
Bonus :

- Ajout d’un bouton Clear et Delete
- Ajout d’un bouton Comma
- Priorités de calcul
- Réduction des nombres à virgules à 3 chiffres après la virgule

QU’EST-CE QUE LE MODÈLE MVC ?

Le principe MVC est un patron de conception célèbre et très utilisé qui signifie :
Model View Controller.

- Le Modèle : gère la logique du programme
- La Vue : se concentre sur l'affichage
- Le Contrôleur : récupère les informations du modèle et les affiche dans la vue.



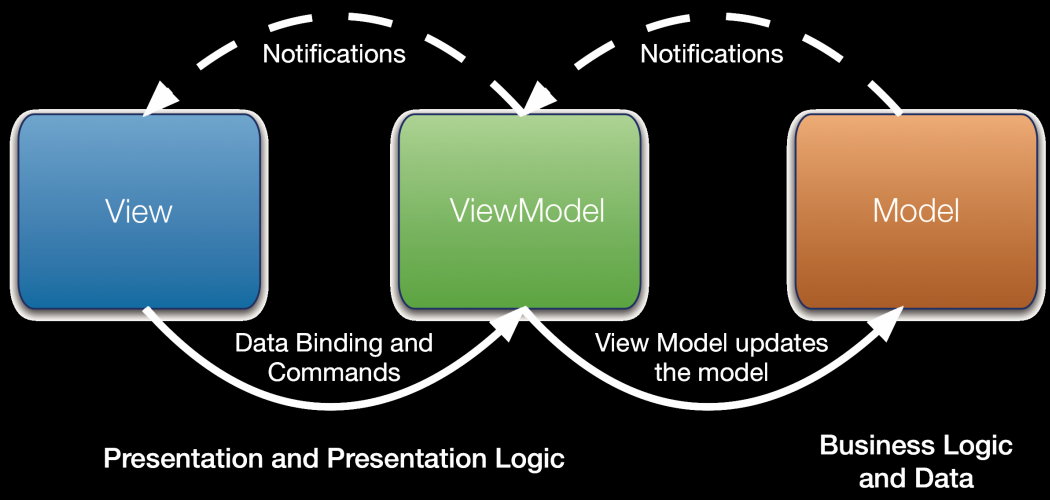
QU'EST-CE QUE LE MODÈLE MVVM ?

Le Model-View-ViewModel, est un modèle d'architecture.

Pour ce projet il a été préférable de choisir le MVVM Réactif natif.

Réactif => La vue est abonnée à des closures, une closure c'est une variable dont le type est une fonction.

- Les outputs : receptacles des événements des inputs correspondants.
- Le ViewController : controlle la vue, il est claire, simple, il ne connait pas les données.

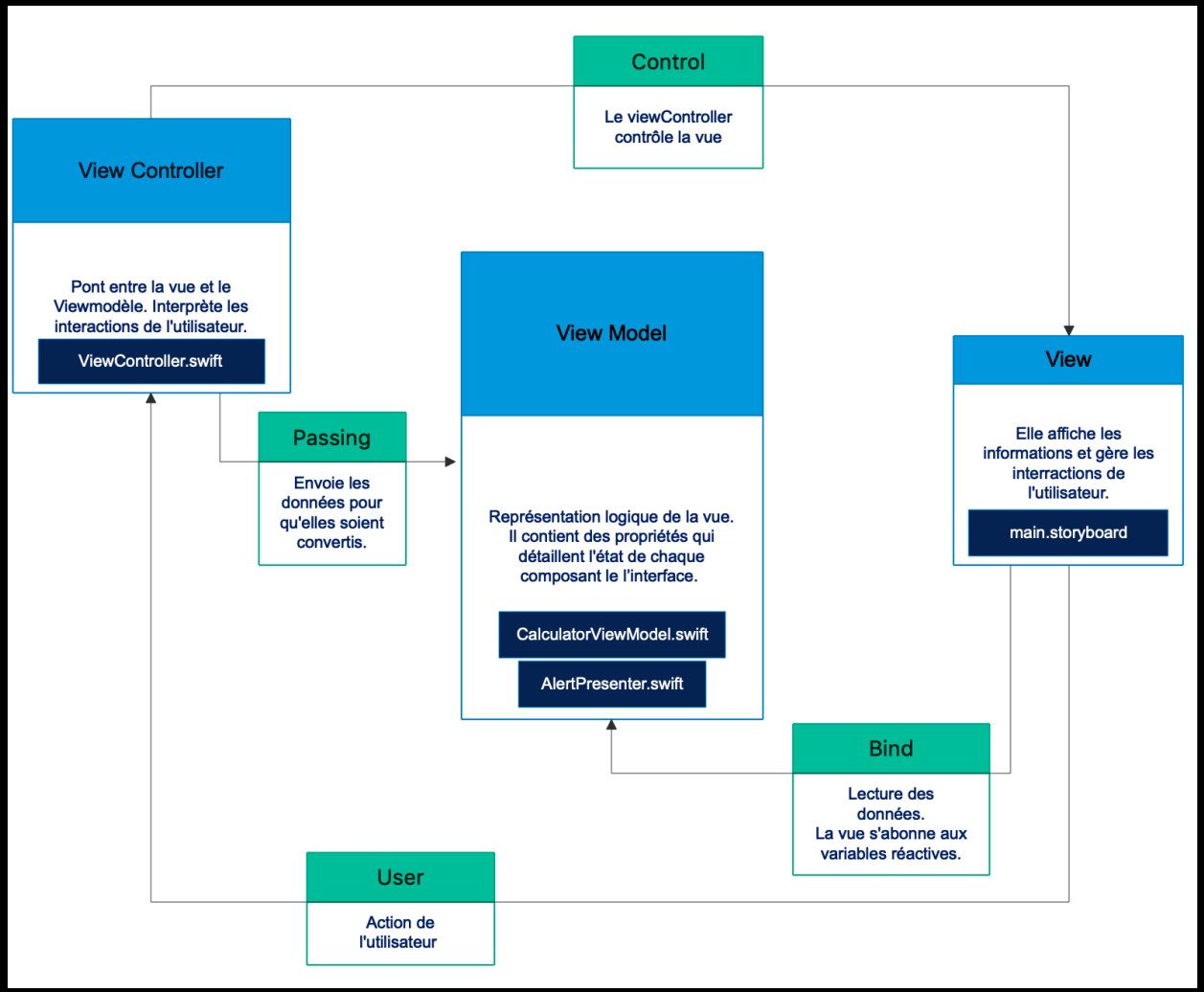


LA DÉMARCHE UTILISÉE

1. D’après le code Legacy fourni, difficilement réutilisable et qui fonctionne de façon incomplète.
2. Je ne me risque pas à toucher le code tant que je ne l’ai pas testé.
3. J’écris la suite des tests de mon ViewModel, testant les comportements attendus.
4. Je déporte la logique du ViewController dans mon ViewModel en toute sécurité.

SIMULATION DE L’APPLICATION

LE DIAGRAMME MVVM RÉACTIF NATIF DE L'APPLICATION



PRÉSENTATION DU CODE


```
// Outputs

var displayedText: ((String) -> Void)?
var nextScreen: ((Screen) -> Void)?
```

```
// Inputs

func viewDidLoad()
func didPressOperand(operand: Int)
func didPressOperator(at index: Int)
func didPressClear()
func didPressDelete()
```

PRÉSENTATION DU CODE

QUESTIONS ?