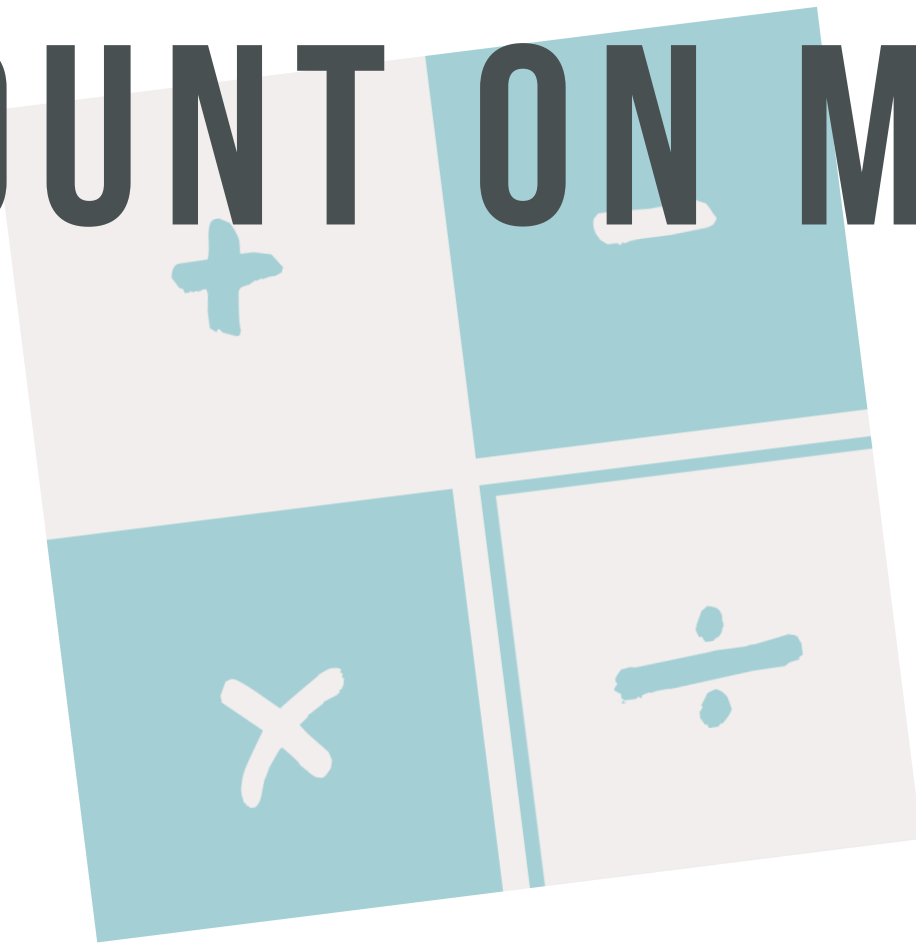


COUNT ON ME



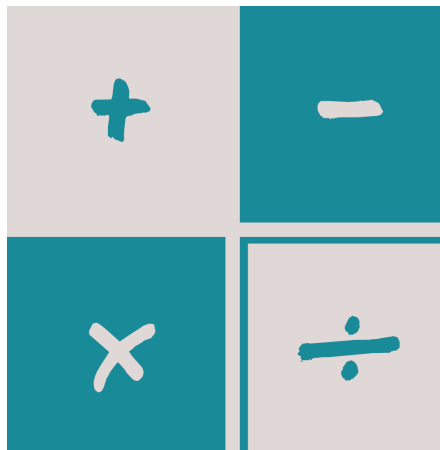
Projet 5 OPENCLASSROOMS - iOS

DÉCEMBRE 2019

PASTE Erwan

Mentor : Lilian Alvarez

Améliorez une application existante



COMPÉTENCES ÉVALUÉES

Utiliser le principe MVC

Apporter des améliorations de fonctionnalités à une application

Corriger des dysfonctionnements sur une application

Effectuer une suite de tests unitaires et d'intégration

LIVRABLE :

Le code source sous forme de projet Xcode, mis en ligne sur Github.


Le code doit :

- Être commentés correctement
- Écrit en anglais
- Respecter le MVC
- Accompagné de tests unitaires complet
- Permettre l'affichage de l'application dans toute les tailles d'iPhone en mode portrait.
- Être exempt de tout erreur ou warning.
- Être fonctionnelle sur iOS 11 et supérieur et écrit en Swift 4 minimum.
- Un support (PowerPoint ou PDF) décrivant l'architecture de l'application mettant en évidence la mise en place de l'architecture MVC

RÉSUMÉ DU TRAVAIL EFFECTUÉ :



A partir d'un simple StoryBoard et quelques lignes de code dans le ViewController :

- Création d'un XD Mockup
 - Ajout des touches multiplier, diviser, point, AC
 - Ajout des contraintes dans le storyboard (Portrait iPhone)
 - Création et ajout d'icône dans les ASSETS tous formats
 - Création d'un launchScreen
 - Ajout des tests unitaires
 - Ajout des tests d'interface
 - Mise en place d'un modèle MVC
- 

04 ARCHITECTURE :

Le LaunchScreen



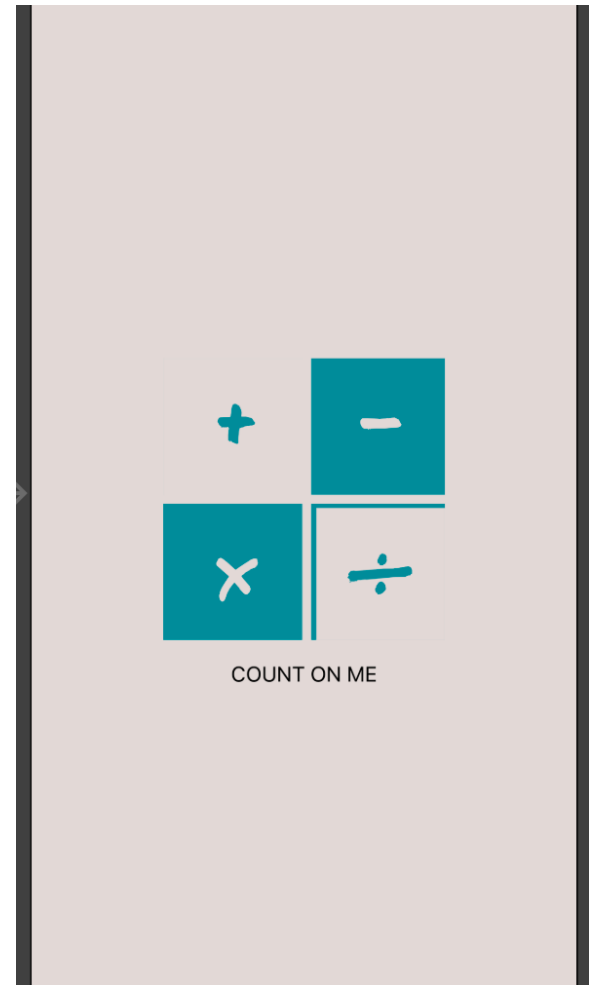
Launch Screen

BackGround color #DFD8D6

Composé dans le storyboard

Contraintes ajoutées pour une adaptation
de la taille :

Centrage horizontal et vertical



04 ARCHITECTURE :

Le Main Story Board



Mockup réalisé dans le main StoryBoard

Color #3F8896 et #DFD8D6

Police : Système

Contraintes ajoutées avec les variations pour iPhone portrait toutes tailles

"TextView"

qui contiendra le résultat et la saisie utilisateur

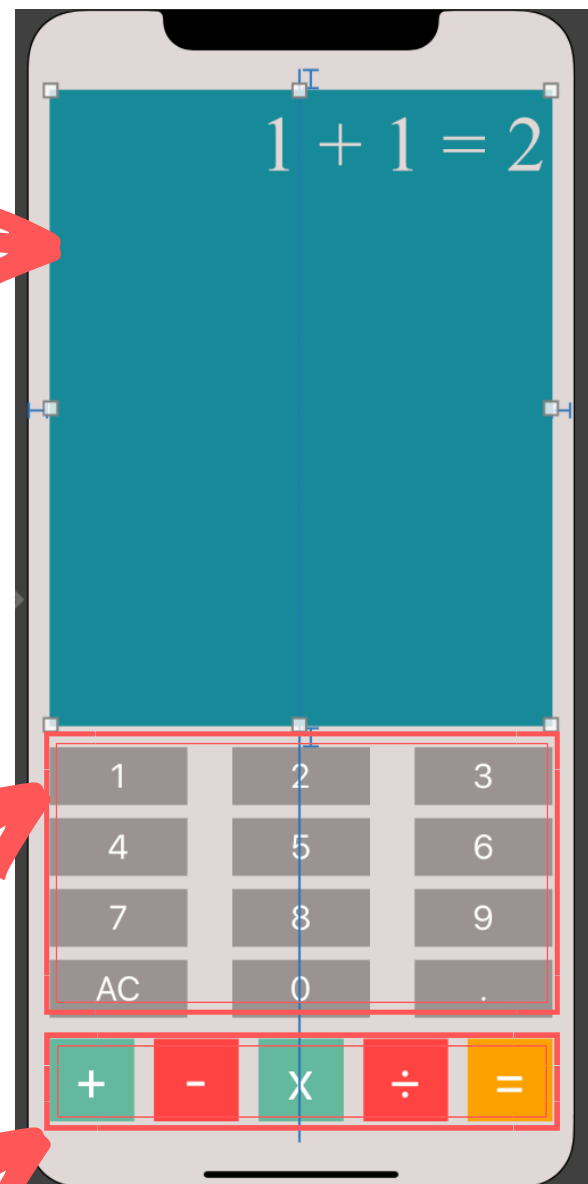
"StackView" Horizontale

Parent de 3 **StackViews Verticale**

Ces dernières contiennent des UIButton

StackView Horizontale

Contient les UIButton pour les opérations



04 ARCHITECTURE :

MVC

CHOIX :

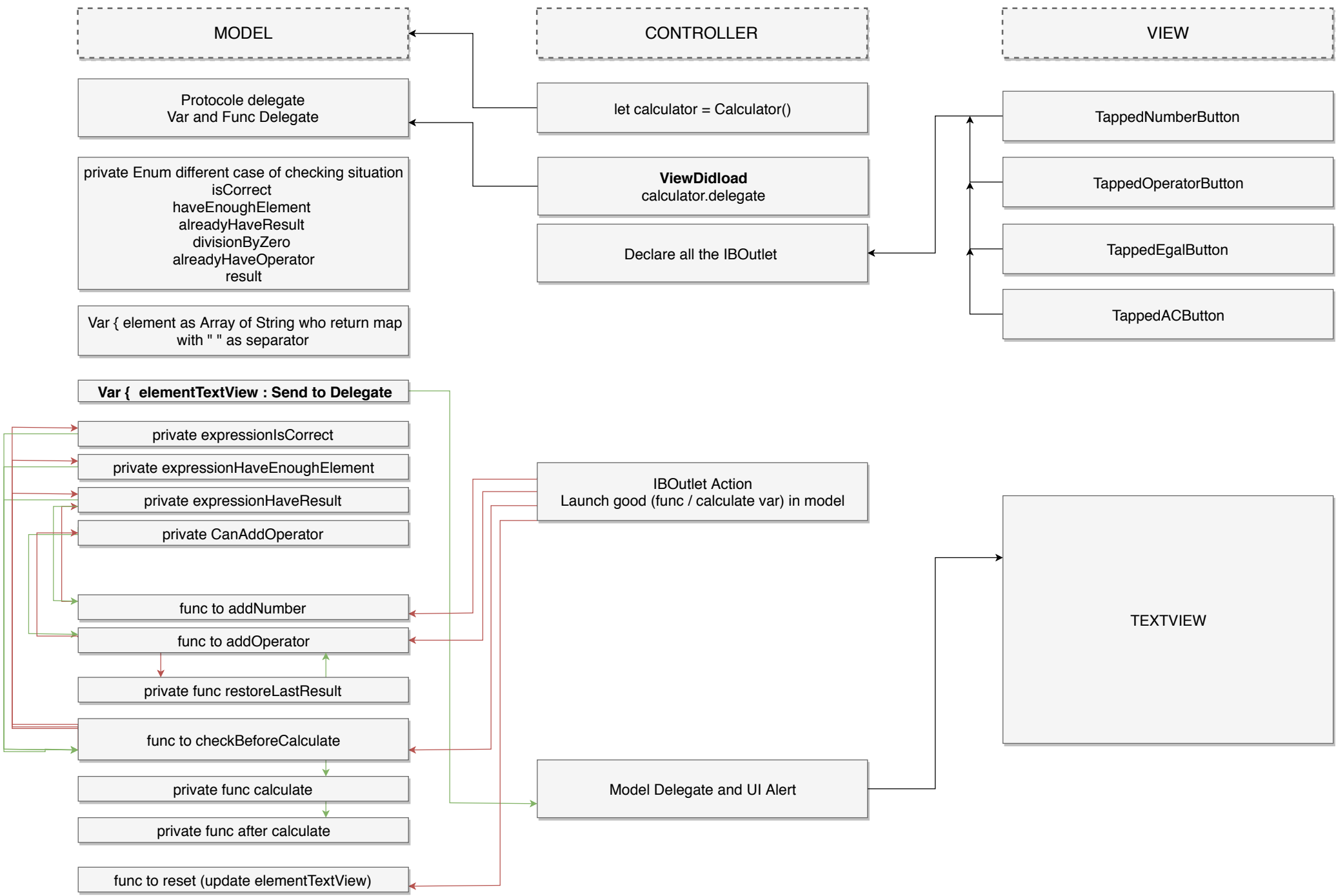


Le model MVC répond plutôt bien à ce type de projet
En effet, la logique peut restée entièrement dans le model, tandis que le controller gère uniquement la liaison entre ce que l'utilisateur a tapé sur l'écran, et quel fonction le model doit exécuter.

NOTE :

Il y aurait pu avoir une couche supplémentaire au niveau du model, dissociant la logique de Checking, et celle des calculs finaux.

Voyons dans le PDF suivant, comment le MVC a été utilisé avant de le présenter dans le code



04 LES TESTS :

Test Unitaire



Les tests unitaires ont été tapés de manière à couvrir entièrement la logique du modèle.

Nous allons également voir ceci dans l'explication orale

Test Interface



Quand aux tests d'interface, ils viennent tester l'ensemble des boutons de l'écran un après l'autre
Puis un test de performance est réalisé