

TP 2 : Création d'interfaces graphiques en Flutter

Aliou GUEYE

January 2, 2023

1 Introduction

Le but de ces exercices est de se familiariser avec les Future, Async, FutureBuilder, ainsi que la gestion des state en utilisant les Provider, et les Bloc (Business Logic Component). Ainsi, on sera amené à organiser le projet par package de responsabilité comme ça a été expliqué dans le cours : package presentation (avec des sous package : animations, pages, screens, widgets), package data (avec des sous package : dataproviders, models, repositories), et package business_logic (avec des sous package : blocs, cubits).

Pour cela on se base sur l'application Quizz « Questions/Réponses ».

2 Exercice 1 : Utilisation du Povider

cette exercice consiste à utiliser les « Povider » pour la gestion des états plus tôt que l'utilisation du setState. Pour ce faire, voici quelques indications :

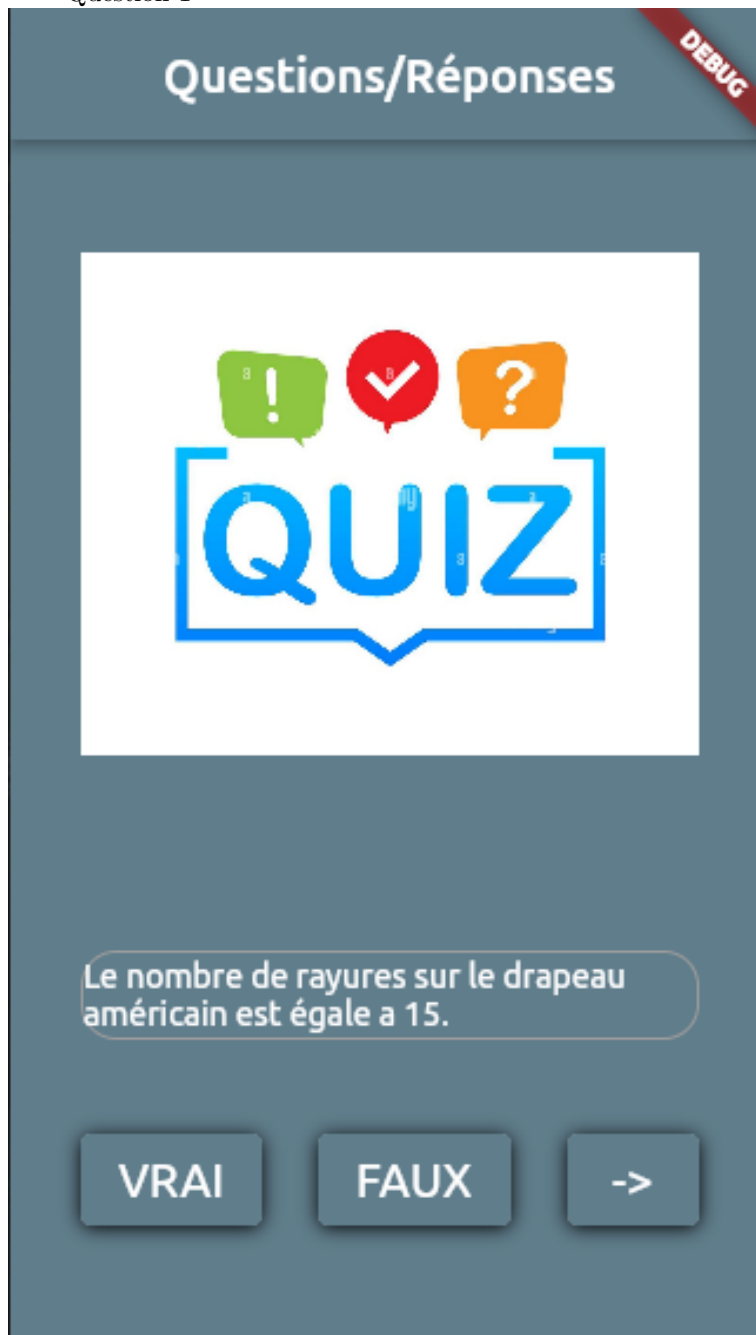
- Ajouter provider : 6.0.1 dans le « pubspec.yml »
- Quelques classes utiles : ChangeNotifier, ChangeNotifierProvider, Consumer, Provider.of ;Class-Model; (context, listen: false).method();
- Notifiez un changement en utilisant la méthode notifyListeners();
- Déclarez un ChangeNotifier

2.1 Exercice 2 : Utilisation du Bloc/Cubit

Le deuxième exercice consiste à utiliser les Bloc/Cubit pour la gestion des états plus tôt que l'utilisation du setState. BLoC (Business Logic Components) est un modèle de conception spécifiquement créé pour Flutter et qui permet de gérer le « state management » de nos apps. Concrètement, BLoC nous permet de traiter le contenu dynamique de nos applications avec une architecture dédiée.

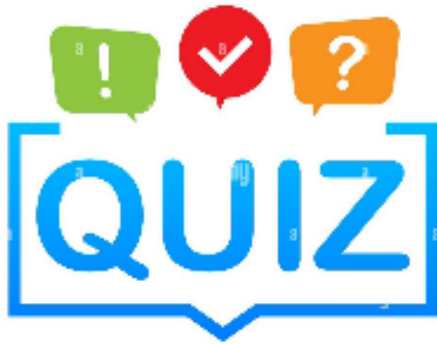
Voici à quoi ressemble l'application :

- Question 1



Questions/Réponses

DEBUG



l'animal national de l'Australie est le kangourou rouge.

VRAI

FAUX

->

- Resultat :



3 lien git

https://github.com/Aliouc bd/TP2_devMobile