Workshop : Initiation flutter partie 1

Exercice 1 : Créer le projet

Avant toute chose, nous devons créer un projet. Pour cela, utilisez la commande suivante :

flutter create workshop

Maintenant, rendez-vous dans le dossier lib. Ce dossier contient tout les fichiers sources de l'application.

Lorsque vous créez un projet, flutter génère automatiquement un fichier main qui contient un exemple de code.

A présent, lancer l'application (soit en utilisant l'option intégrée dans visual studio code soit en utilisant la commande *flutter run)*.

Exercice 2 : Créer une application qui utilise une map google

Pour cet exercice, nous aurons besoin du packet *google_maps_flutter.*

Maintenant, vous allez avoir besoin d'une clé d'API. Elles sont disponibles sur le google cloud platform. Vous pouvez suivre ce tutoriel pour en récupérer une.

Pour activer cette clé (pour android) rendez-vous dans le fichier suivant *android/app/src/main/AndroidManifest.xml.* Dans la partie **application,** ajoutez ceci :

```
<meta-data android:name="com.google.android.geo.API_KEY"
               android:value="VOTRE CLE API"/>
```

Pensez à lancer la commande *flutter clean* pour être sûr que les changement apporté soient validé lors de la prochaine compilation.

Etape 1 : Ajout du widget GoogleMap

Le widget devra être un StatefulWidget. Vous pouvez accéder à la doc du widget GoogleMap ici.
La map que vous devez créer devra couvrir tout l'écran.
Vous devez impérativement utiliser un *controller* (cf doc GoogleMap).
La map devra être accompagnée d'une AppBar. Cette AppBar devra être bleu, et avoir comme titre : « Mon Application Google Map ».
La map en elle-même devra :

- Être centrée sur Epitech (il vous faudra chercher les coordonnées d'Epitech).
- Avoir une position de départ qui permet de voir le bâtiment paritalie.
- Mettre une limite de zoom / dezoom et de mouvement limité afin de ne pouvoir voir que Paris et ses alentours.

Lorsque vous avez fini, remplacez votre travail par le contenu de la correction présente dans le fichier Exercice1-Etape1 avant de passé à l'étape suivante.

Etape 2 : Modifier notre map

Dans cette étape nous utiliserons des icones. Pour ce faire, dans le fichier pubspec.yaml vous devez ajouter la ligne suivant sous flutter :

```
flutter:
  uses-material-design: true
```

Maintenant, il est temps d'ajouter des options de modification à notre map.

Vous devez maintenant ajouter un bouton par-dessus la map. Pour ce faire, vous allez utiliser un Stack.

Comme bouton, nous utiliserons un FloatingActionButton. Ce bouton devra :

- Être positionné en bas à droite de l'écran (Padding et un Align seront la meilleur solution)
- Être rouge clair
- Avoir une icone
- Lorsqu'il est pressé, changer le mode d'affichage de la map

Le mode d'affichage de la map est géré par la propriété mapType dans le constructeur de notre map. Pour modifier une propriété d'un Widget en *direct*, vous devez utiliser la fonction SetState().

Etape 3 : Ajouter des points sur la map

Le Widget GoogleMap permet de facilement créer des pinpoints sur une map.

Pour commencer, ajoutez un nouveau bouton au-dessus du précèdent. Une nouvelle fois, on utilisera un FloatingActionButton. Pour que ce soit joli, je vous conseille de suivre les mêmes indications que dans l'étape précédente concernant les options du bouton.

GoogleMap possède une propriété markers. Cette propriété demande un set de Marker (set<Marker>). Comme vous l'aurez compris. Lorsque le bouton est pressé, il faudra ajouter un Marker au set de markers. (Pensez à SetState() de nouveau !).

Le nouveau marker que nous ajouterons se retrouvera automatiquement au milieu de la map. Il nous faut donc avoir la position de la caméra à tout moment.

Pour cela, nous devons utiliser la propriété onCameraMove. Cette propriété prend une fonction qui sera appelée en boucle quand on bouge la map. Il faut donc que cette fonction enregistre la position de la camera dans une variable a chacun de ses appels.

Les Markers que nous ajoutons devront :

- Être positionné au milieu de la map
- Avoir l'icone standard des pinpoints google map
- Avoir une InfoWindow (un texte qui s'affiche lorsqu'on appuie sur un marker)
- Un nom unique

Voilà, vous avez créé votre première App en flutter.

Vous trouverez la correction entière de l'exercice dans le fichier *CorrectionFull.*

S'il vous reste du temps, hésitez pas à venir nous voir pour que l'on vous guide vers d'autres exercices.