

TP Mona Lisa

Flutter



Objectifs

- Prise en main de Flutter et Dart,
- Découverte de la mise en page à l'aide de widgets,
- Gestion des interactions de l'utilisateur,
- Gestion d'état applicatif (Stateless / Stateful),
- Emploi d'assets,
- Affichage conditionnel.
- Aidez-vous de la documentation officielle de Flutter pour réaliser ce TP
(cf. <https://docs.flutter.dev/cookbook>).

Flutter

Alexandre Leroux (alex@sherpa.one) - 2024

Etape 1 : Initialisation

(cf. structure de l'application en annexe).

1. Créez un nouveau projet Flutter avec la commande
`flutter create -e monalisa`
2. Importez le package ***flutter/material.dart*** à la première ligne du fichier
`./lib/main.dart`
3. Créez une classe nommée ***MuseumApp*** héritant de ***StatelessWidget***
4. Faites en sorte que la méthode ***build*** de ce widget retourne un widget de type ***MaterialApp***.
Affectez à l'attribut ***title*** du widget ***MaterialApp***, la valeur "***Museum***".
Pour l'attribut ***theme*** du widget ***MaterialApp***, renseignez le ***ThemeData*** configuré avec la couleur primaire de votre choix (ex: *Brown*).
5. Créez une nouvelle classe nommée ***ArtworkWidget*** héritant de ***StatelessWidget***.
Employez un widget ***Scaffold*** en tant que widget racine retourné par la méthode ***build***.

Intégrez les widgets nécessaires au fonctionnement d'un widget ***Scaffold*** pour obtenir un rendu graphique à l'écran (ex : *appBar, body, floatingActionButton...*).
6. Dans le widget ***MuseumApp***, employez une instance du widget ***ArtworkWidget*** en tant que valeur de l'attribut ***home*** du widget ***MaterialApp***
7. Dans le fichier ***main.dart***, créez une méthode nommée ***main*** préfixée par le retour de type ***void***
Au sein de la méthode ***main***, exéutez la méthode ***runApp*** avec, en paramètre une instance de ***MuseumApp***
8. Afin de vérifier le fonctionnement de votre application, exécutez-la dans votre navigateur web *Chrome* ou *Edge*, ou dans un émulateur mobile ou encore dans un device mobile connecté.
Employez la commande ***flutter run*** pour démarrer l'application (ou en cliquant depuis le lien " *run*" affiché par votre IDE dans le fichier *main.dart*).

Etape 2 : Assets

1. A la racine de votre projet, créez un dossier `./images` et placez le fichier **Mona_Lisa.jpg** (fourni).
2. Adaptez le fichier `./pubspec.yaml` afin d'intégrer le fichier **Mona_Lisa.jpg** au build de l'application (attention à l'indentation du fichier yaml).

Attention, les modifications effectuées dans le fichier `./pubspec.yaml` ne prennent effet qu'après avoir relancé l'application (le *hot reloading / restarting* ne suffisent pas). Stoppez et relancez l'application.
3. Téléchargez la police de votre choix (ex: *Merriweather*) depuis *Google Fonts* (ex : [Merriweather - Google Fonts](#)). Dézippez l'archive et placez les fichiers `.ttf` utiles dans le dossier `./fonts/<nom-de-la-police>` (ex: `fonts/merriweather`).

Etape 3 : UI

(cf. captures d'écran en annexe).

1. Dans le widget **ArtworkWidget**, affichez l'image **Mona_Lisa.jpg**. Centrez l'image horizontalement.
2. Sous l'image, affichez le texte "**Mona Lisa**" avec la police personnalisée , `fontSize` à 30 et couleur **Brown**.
3. Sous le texte "**Mona Lisa**", affichez un second texte contenant la mention "**Léonard De Vinci**", avec la police personnalisée, `fontSize` à 15, `fontWeight` en **bold**, et couleur **Brown**.
4. Superposez à l'image un widget **Icon** avec le symbole **favorite**, `fontSize` à 100, couleur **White**, avec une **opacité** de 75%, centré horizontalement et verticalement.
5. Sous le texte "**Léonard De Vinci**", affichez deux **IconButton**, un avec l'icône **article** et l'autre, avec l'icône **favorite**.

Etape 4 : StatefulWidget

1. Convertissez le widget **ArtworkWidget** en **StatefulWidget**, afin de disposer d'un état applicatif (state) permettant de mettre à jour l'interface (UI) en fonction des interactions de l'utilisateur et de l'évolution des variables d'état.
2. Créez une variable d'état **_isFavorite** de type **bool**, initiée à **false**
3. Lorsque l'utilisateur clique sur l'**IconButton** de type **favorite** située en-dessous de l'image, alternez la valeur (**true / false**) de la variable d'état **_isFavorite**
4. Faites en sorte que la couleur de **l'icône superposée à l'image, passe de White à Red et que son opacité passe de 75 à 100%**, lorsque **_isFavorite** prend la valeur **true**
5. Alternez également la couleur de l'**IconButton favorite** (placé sous l'image), de **Brown à Red**, lorsque **_isFavorite** passe de **false** à **true**
6. A l'aide d'un widget **Snackbar**, affichez une notification (ex: "*Oeuvre ajoutée à vos favoris*") lorsque l'utilisateur clique sur l'**IconButton favorite** et que la valeur de **_isFavorite** est à **true**
7. Au clic sur l'**IconButton** de type **article**, affichez / masquez le texte suivant à l'aide d'une variable d'état, nommée **_showDescription** de type **bool** :

La Joconde, ou Portrait de Mona Lisa, est un tableau de l'artiste Léonard de Vinci, réalisé entre 1503 et 1506 ou entre 1513 et 1516,2, et peut-être jusqu'à 1517 (l'artiste étant mort le 2 mai 1519), qui représente un portrait mi-corps, probablement celui de la Florentine Lisa Gherardini, épouse de Francesco del Giocondo. Acquise par François Ier, cette peinture à l'huile sur panneau de bois de peuplier de 77 × 53 cm est exposée au musée du Louvre à Paris. La Joconde est l'un des rares tableaux attribués de façon certaine à Léonard de Vinci.

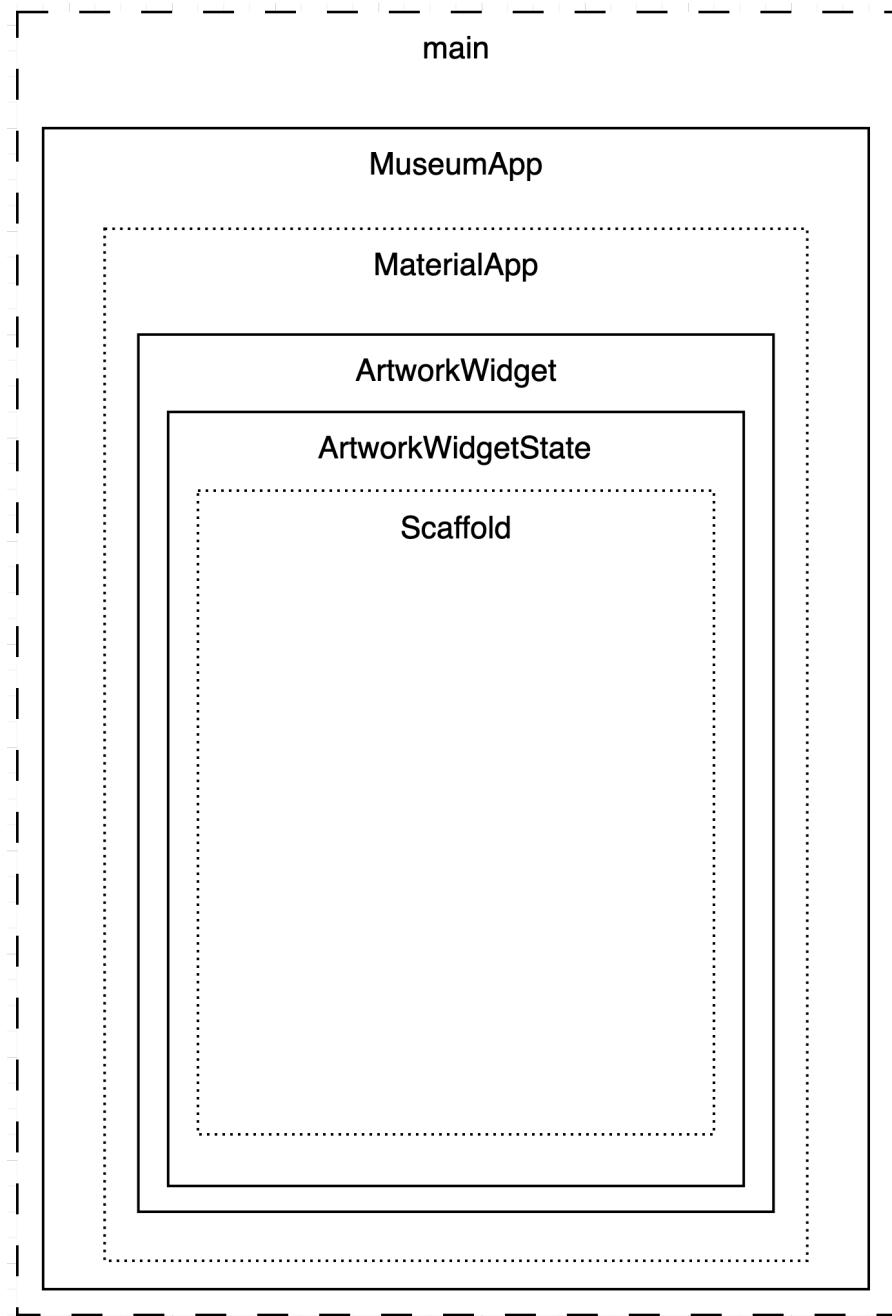
La Joconde est devenue un tableau éminemment célèbre car, depuis sa réalisation, nombre d'artistes l'ont pris comme référence. À l'époque romantique, les artistes ont été fascinés par ce tableau et ont contribué à développer le mythe qui l'entoure, en faisant de ce tableau l'une des œuvres d'art les plus célèbres du monde, si ce n'est la plus célèbre : elle est en tout cas considérée comme l'une des représentations d'un visage féminin les plus célèbres au monde. Au xxie siècle, elle est devenue l'objet d'art le plus visité au monde, devant le diamant Hope, avec 20 000 visiteurs qui viennent l'admirer et la photographier quotidiennement.

8. Faites en sorte que le texte soit intégré dans un container adapté aux dimensions de l'image (cf. captures d'écrans en annexe) et puisse défiler verticalement (scroll), dans le cas où le volume de texte le nécessite. Superposez le texte à l'image.

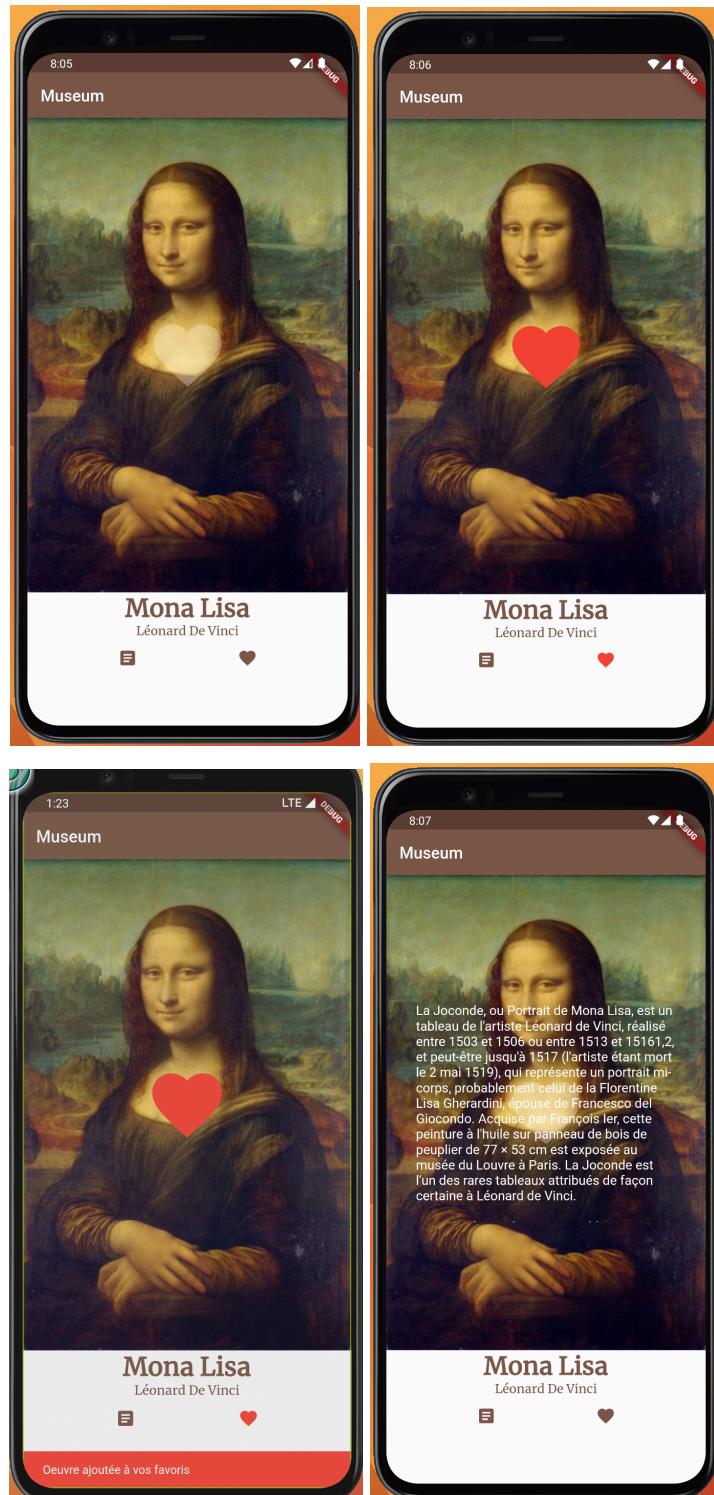
Etape 5 : Refactoring du code

- Procédez à une réorganisation du code de façon à disposer d'un seul fichier par entité logiciel :
 - 1 fichier main.dart,
 - 1 fichier par widget Stateless,
 - 1 fichier par widget Stateful
 - classe de widget Stateful
 - classe de State
- Adaptez les imports en conséquence (préférez les imports avec un chemin absolu, commençant par "package:...").

Structure de l'application



Captures d'écran



Flutter

Alexandre Leroux (alex@sherpa.one) - 2024

7/8

Crédits

- Texte et image : https://fr.wikipedia.org/wiki/La_Joconde