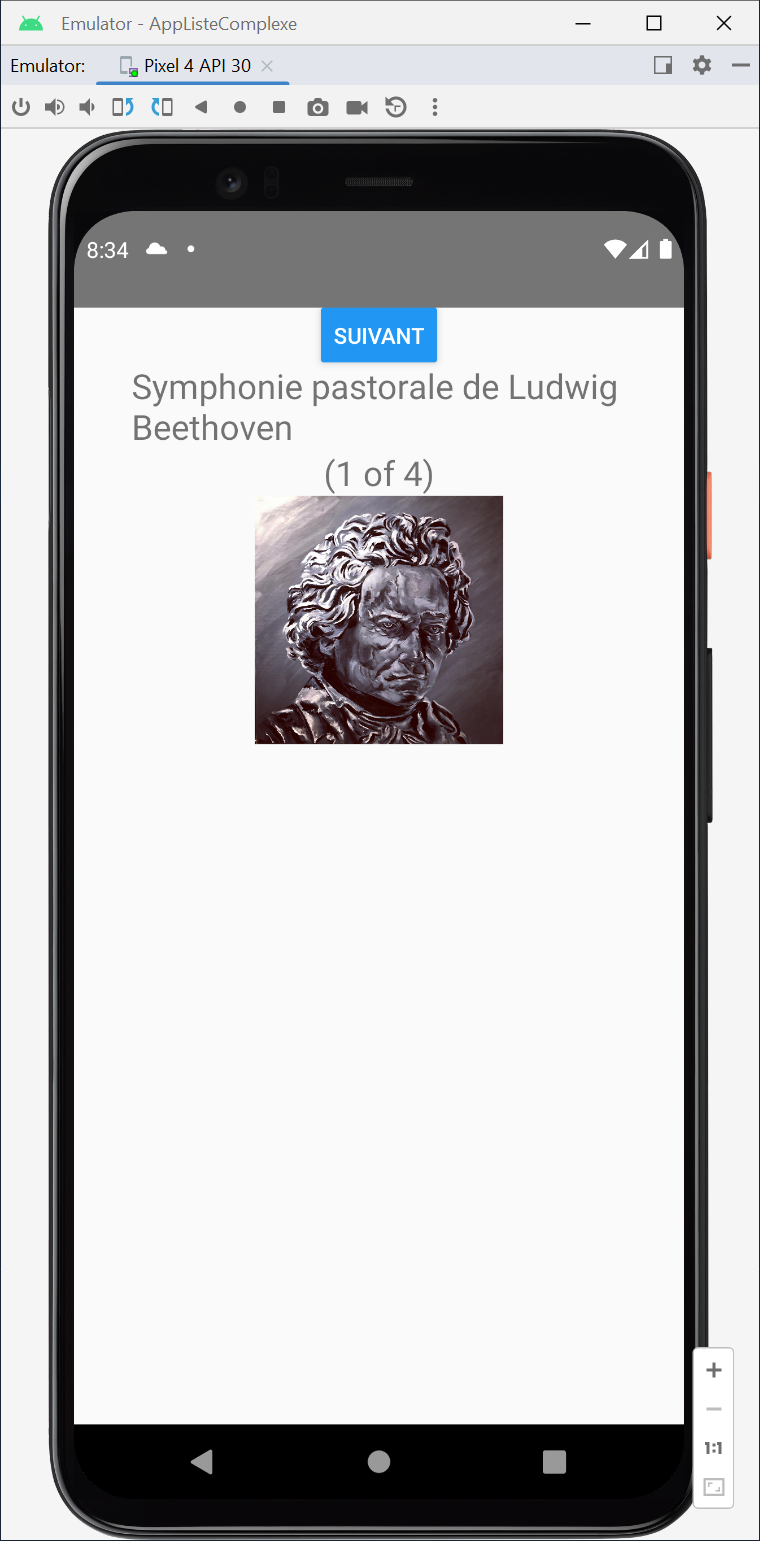
# Annexe 4 – Exercices formatifs

* Habiletés à maîtriser : Composants, useState, StyleSheet
* Vous utilisez le projet présent dans Annexe4Exercices sur LÉA. Vous le roulez ensuite en faisant d’abord npm install expo , puis npx expo start.
* Si vous avez étudié, essayer de le faire sans consulter vos notes / autres exercices. Bon travail !

1. Vous reconnaissez l’app sur les chefs d'œuvre de musique classique dont on a discuté dans l’annexe 3. Étudiez le code. Le bouton ne permet pas de passer de la description d'une œuvre à l'autre car aucun useState n'est utilisé.

Faites en sorte que la variable index devienne une variable d'état, un peu comme la mémoire du composant.

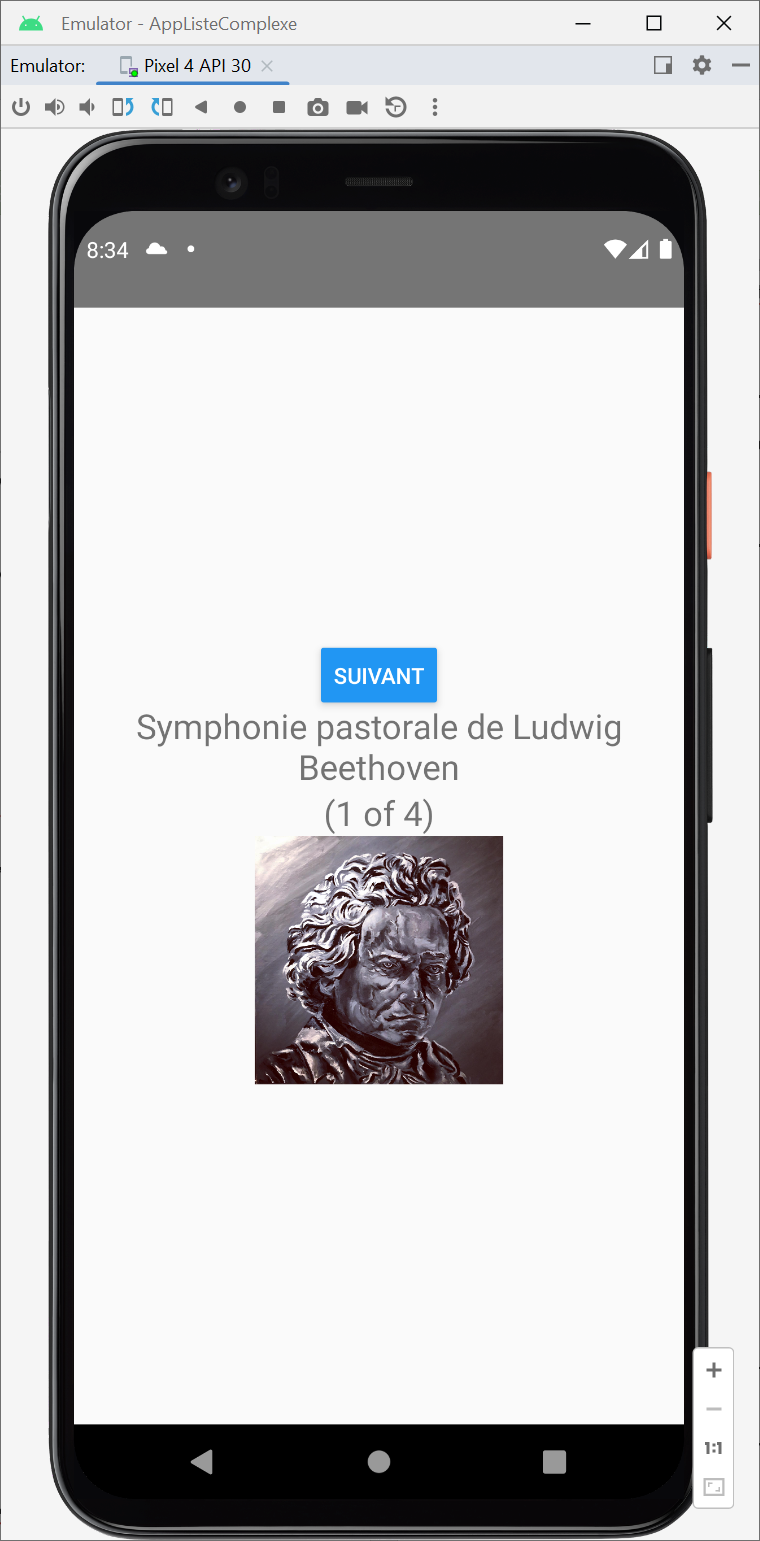
De cette façon on peut passer d'un chef d'œuvre à l'autre :



*Après 2 clics, on affiche le 3e de 4 chefs d'oeuvre*

On veut faire en sorte que le composant soit centré verticalement, on suggère d'ajouter justifyContent:center dans la règle main de l'objet styles. Est-ce que ça fonctionne ?

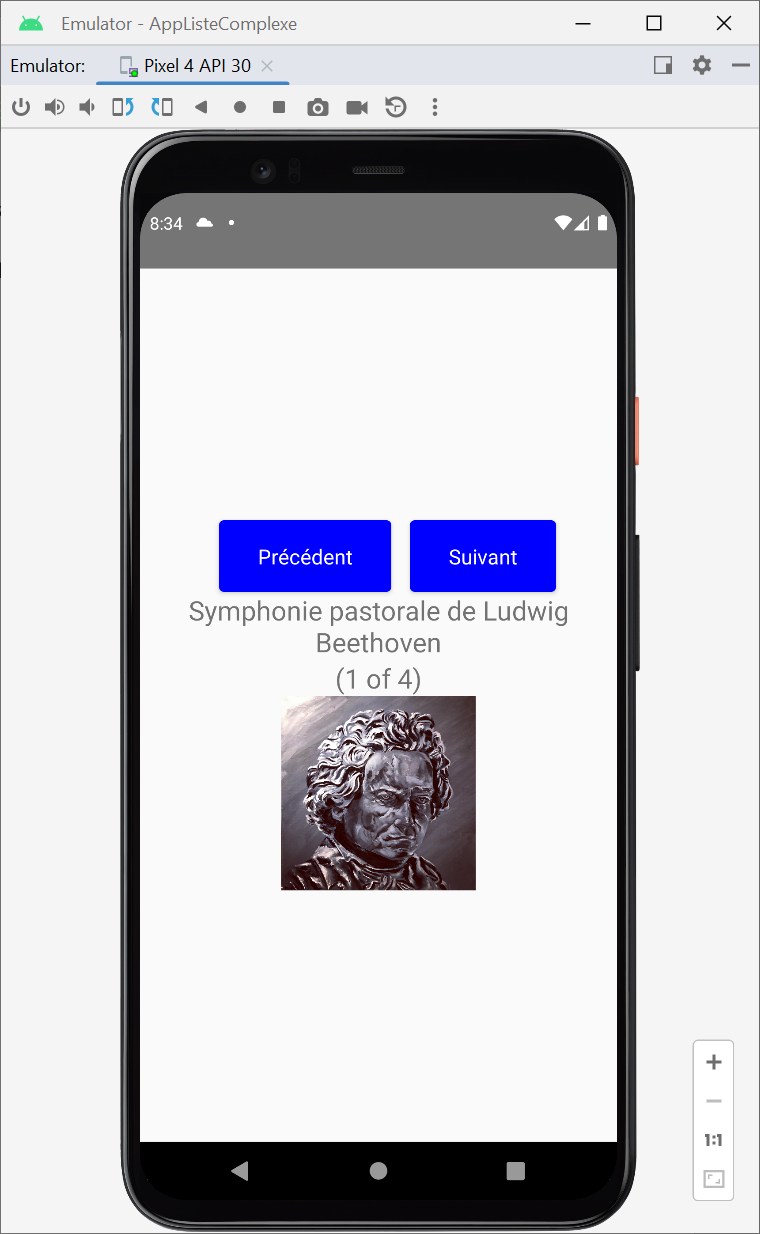
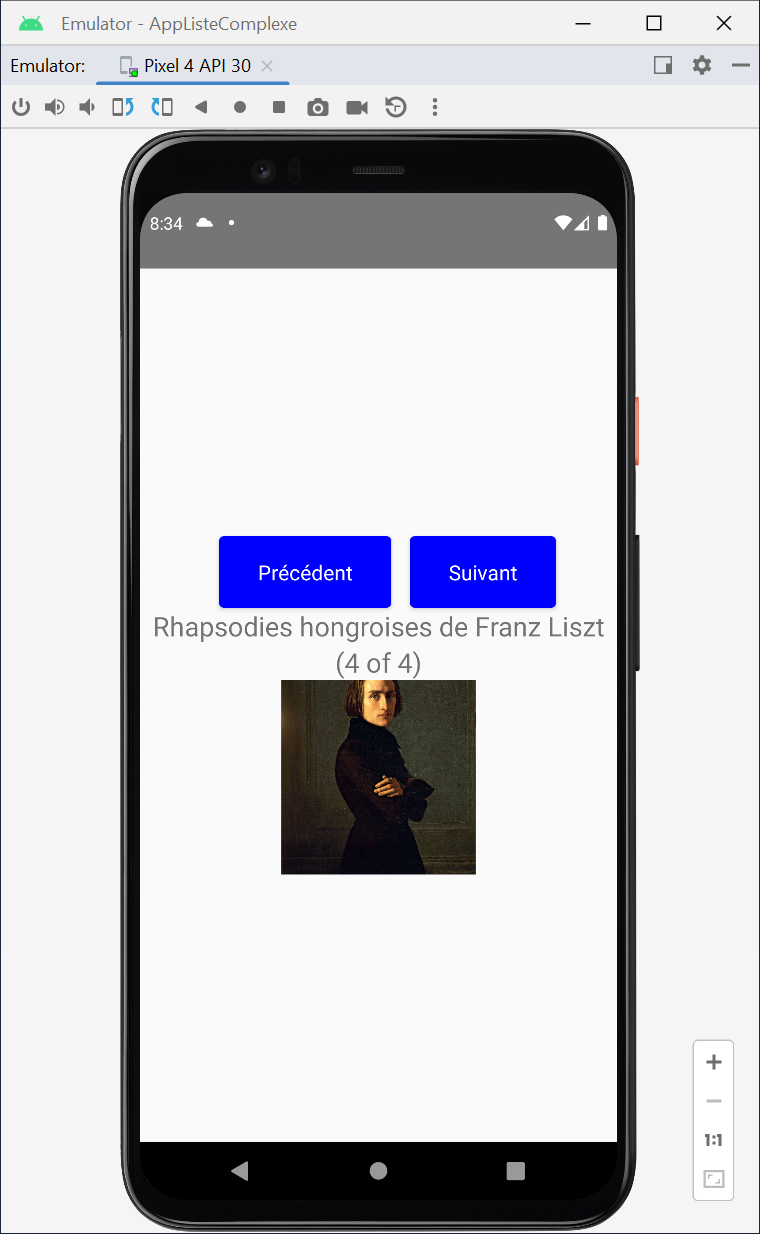
Quelle autre règle peut-on ajouter à celle-ci-haut pour que le composant soit centré verticalement ? Pourquoi est-ce nécessaire ?



Toutes les dimensions en ReactNative sont sans unités et représentent des dp ( density independant pixels )

1. L'app plante lorsqu'on est rendu à index =4 car il n'y a pas de 5ème image. Faites en sorte qu'on revienne au début lorsqu'on est rendu à la fin de la liste.

Ajoutez un bouton Précédent qui permettra d'afficher les oeuvres dans l'autre sens.



1. Vous remarquez que les boutons que j'ai fait n'ont pas la même apparence que les vôtres. En passant êtes-vous capable de donner une marge à vos boutons en utilisant une règle appliquée à eux afin de les espacer horizontalement ?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. On ne peut pas appliquer de styles à un composant Button; on peut seulement changer sa couleur avec le props color ( à passer comme attribut dans la balise )

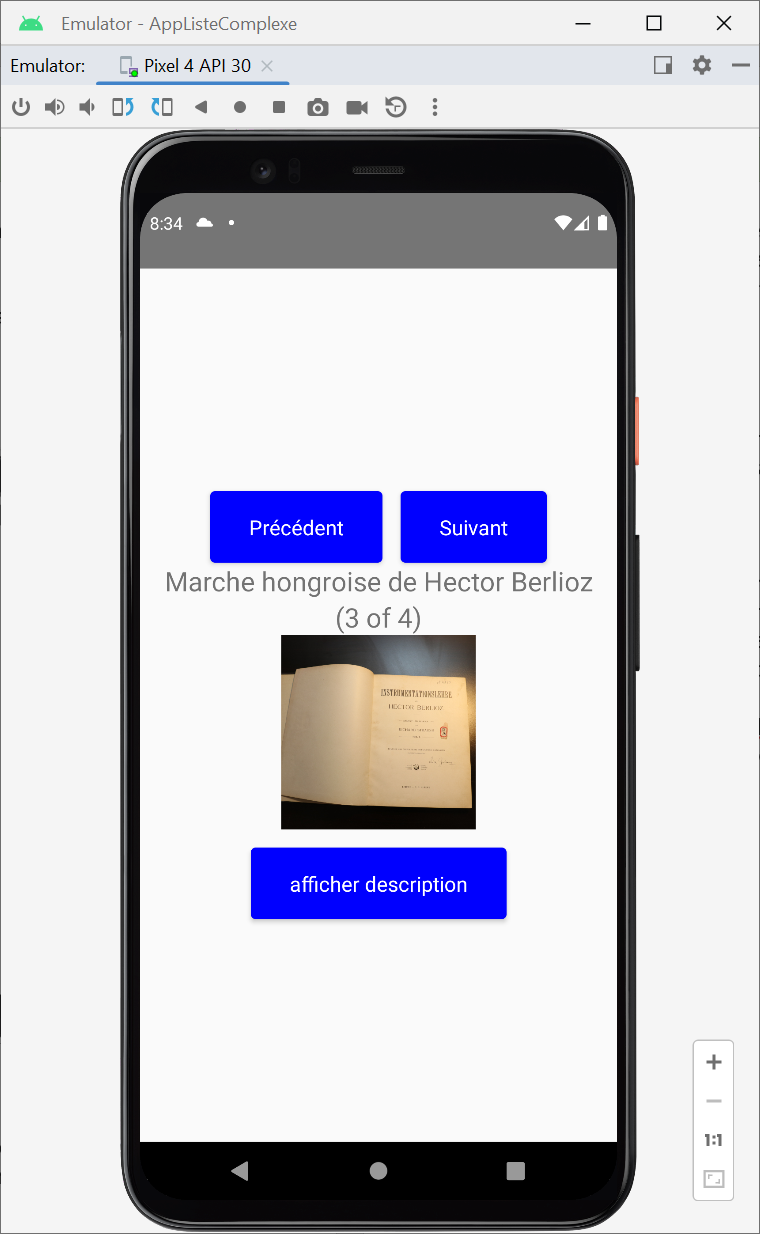
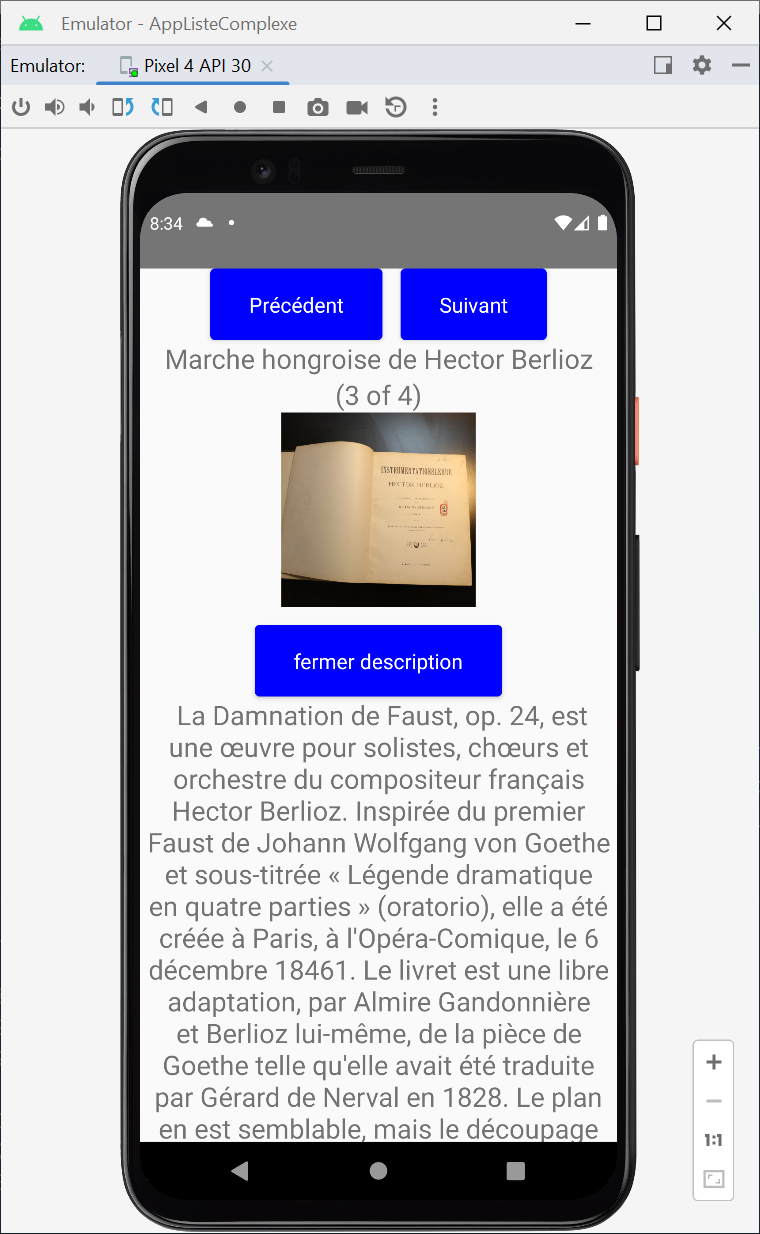
Une façon de faire des boutons personnalisés consiste à utiliser comme à l'annexe précédente le composant TouchableHighlight ou TouchableOpacity à la place du Button. Cependant…

* On doit placer un composant Text à l'intérieur
* On doit "dessiner" le bouton à l'aide d'une règle dans notre StyleSheet

Remplacer vos Buttons pour des composants que vous pourrez *designer* aisément pour avoir un résultat similaire au mien. Ajoutez la marge également !

1. Ouvrez le fichier donnees.js représentant les données des œuvres. Pour chaque objet du tableau, ajoutez un attribut desc représentant une brève description de l'œuvre en question ( vous pouvez utiliser le premier paragraphe de la page Wikipedia de l'œuvre, la marche hongroise de Berlioz fait partie de la damnation de Faust )

Ajoutez un bouton / TouchableOpacity en bas de la photo de manière à afficher ou non l'attribut desc de l'objet affiché :

Englobez votre composant d'un ScrollView pour pouvoir faire défiler les textes trop longs.

**La façon d'associer un style à un ScrollView est le props contentContainerStyle plutôt que style**