

★ Obtenez un accès illimité au meilleur de Medium pour moins de 1\$ /semaine. [Devenir membre](#)



Android Compose - Créer un écran de démarrage avec Lottie



Ridvan Özcan · [Suivre](#)

4 minutes de lecture · 17 mars 2024

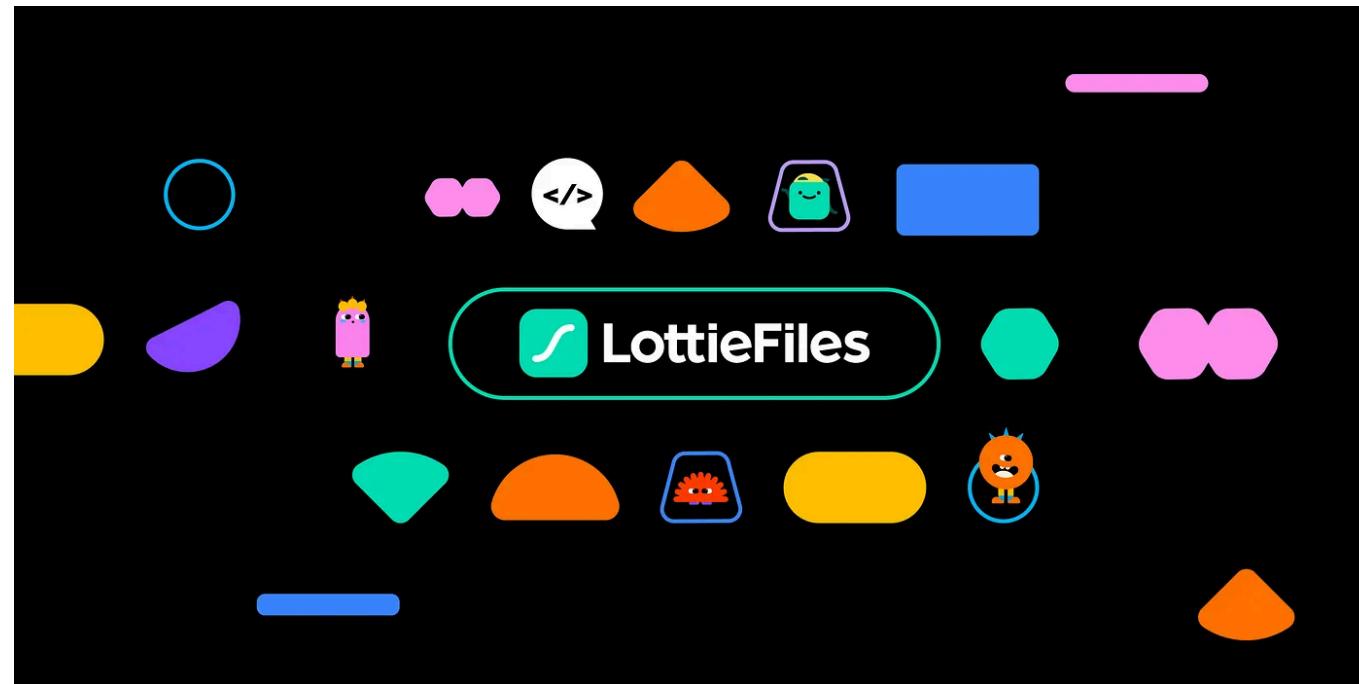


65



...

L'écran de démarrage Android sert d'interface initiale qui accueille les utilisateurs lors du lancement d'une application. Dans ce didacticiel, nous allons créer un écran de démarrage dynamique en utilisant [Lottie](#), une puissante bibliothèque d'animation pour Android, en conjonction avec Navigation-Compose.



<https://lottiefiles.com/>

Ajoutons Navigation-Compose et Lottie au fichier *build.gradle* au niveau de votre application .

```
dépendances{  
    ....  
    implementation 'com.airbnb.android:lottie-compose:6.3.0'  
    implementation 'androidx.navigation:navigation-compose:2.7.7'
```

```
    ...  
}
```

Nous devrons définir la couleur d'arrière-plan et les couleurs de la barre système pour pouvoir jouer correctement nos animations. Nous ajouterons les propriétés ci-dessous au `themes.xml` fichier

```
< nom du style = "Theme.AnimatedSplashScreen" parent = "android:Theme.Material.  
    < nom de l'élément = "android:statusBarColor" > @color/black </ item >  
    < nom de l'élément = "android:windowBackground" > @color/noir </ élément >  
</ style >
```

Commençons par définir une classe scellée nommée `Destination` pour gérer les itinéraires dans notre graphique de navigation. Étant donné qu'il s'agit d'un exemple basique, nous allons rester simple avec seulement deux écrans : l'écran de démarrage et l'écran d'accueil.

```
classe scellée Screen ( val route: String) {  
    object Splash : Screen( "splash_screen" )
```

```
object Home : Screen( "home_screen" )  
}
```

Ici, nous avons créé une classe scellée appelée `Screen`. À l'intérieur, nous avons défini deux objets, `Splash` et `Home`, chacun représentant un écran dans notre application. De plus, nous avons attribué une `route` propriété à chaque objet, qui sera utilisée à des fins de navigation.

Ensuite, configurons notre graphique de navigation. Nous allons créer un nouveau fichier Kotlin nommé `SetupNavGraph.kt` pour gérer le graphique de navigation de nos écrans.

```
@Composable  
amusant SetupNavGraph (navController : NavHostController ) {  
    NavHost(  
        navController = navController,  
        startDestination = Screen.Splash.route  
    ) {  
        composable(route = Screen.Splash.route) {  
            SplashScreen(navController = navController)  
        }  
        composable(route = Screen.Home .route) {  
            Box(  
                modifier = Modifier  
                    .fillMaxSize()  
            )  
        }  
    }  
}
```

```
        .background(Color.White),  
        contentAlignment = Alignment.Center  
    ) {  
    Text(text = "Hello World" , color = Color.Black)  
}  
}  
}
```

Dans ce code, nous avons défini une fonction composable `SetupNavGraph` chargée de configurer notre graphique de navigation. Nous utilisons la bibliothèque de navigation de Jetpack Compose pour y parvenir.

Le `NavHost` composable configure un hôte de navigation avec une destination spécifiée `NavController` et une destination de départ. À l'intérieur de `NavHost`, nous avons défini deux destinations à l'aide du `composable` composable. Chaque destination est associée à un itinéraire de notre `Destination` classe scellée.

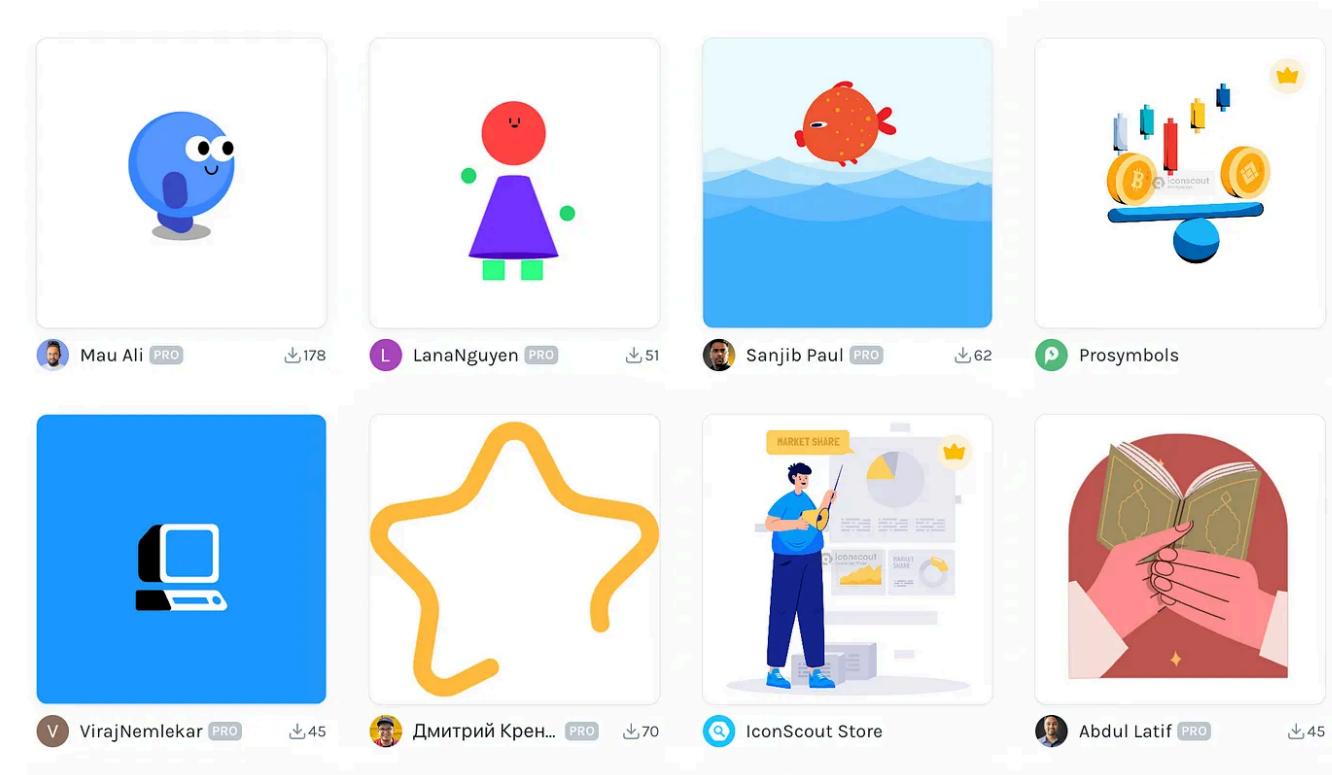
Pour lier le graphique de navigation à l'activité principale, nous appellerons la `SetupNavGraph` fonction à l'intérieur de notre activité principale.

```
class MainActivity : ComponentActivity () {
    override fun onCreate (savedInstanceState: Bundle ?) {
        super .onCreate (savedInstanceState)
        setContent {
            AnimatedSplashScreenTheme {
                val navController = RememberNavController()
                SetupNavGraph(navController = navController)
            }
        }
    }
}
```

Dans ce code, nous avons défini le contenu de l'activité sur une fonction composable qui initialise le graphique de navigation. Nous avons également utilisé `rememberNavController()` pour créer un contrôleur de navigation qui gérera la navigation au sein de l'application.

Désormais, notre activité principale est configurée pour afficher le graphique de navigation défini dans `SetupNavGraph.kt`, qui gère la navigation entre l'écran de démarrage et l'écran d'accueil.

Maintenant, téléchargez Lottie et construisons l'écran qui jouera notre animation, puis accédez à l'écran d'accueil. Nous allons simplement ajouter une vue Lottie et charger un fichier lottie dans la vue.



<https://lottiefiles.com/featured>

Nous pouvons maintenant créer un SplashScreen pour gérer l'interface utilisateur et la logique.

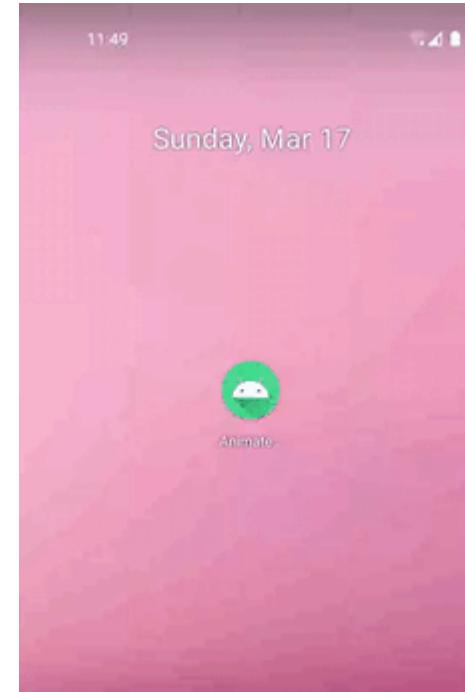
```
@Composable
amusant_SplashScreen (navController: NavHostController ) {
    Box(
        modifier = Modifier
            .fillMaxSize()
            .background(MaterialTheme.colors.primary)
    ) {
        val composition par RememberLottieComposition(LottieCompositionSpec.RawR
        val logoAnimationState =
            animateLottieCompositionAsState (composition = composition)
        LottieAnimation(
            composition = composition,
            progress = { logoAnimationState.progress }
        )
        if (logoAnimationState.isAtEnd && logoAnimationState.isPlaying) {
            navController.navigate(Screen.Home.route)
        }
    }
}
```

Le code fourni configure une composition à utiliser dans notre vue Lottie. L'affectation directe du fichier « logo.lottie » depuis le dossier brut dans les ressources est facilitée. Par la suite, une variable d'état est initialisée pour surveiller la progression de l'animation dans le composable, permettant ainsi des vérifications ultérieures de l'achèvement de l'animation.

Enfin, l'intégration de la vue LottieAnimation se fait en associant les composants composés et la progression. Lors de la transition de l'état vers « lecture » et lorsque l'on atteint l'image finale, une navigation transparente vers le composable suivant est exécutée, obtenant ainsi la fonctionnalité souhaitée.

Pour le code opérationnel complet, veuillez vous référer au lien suivant :
<https://github.com/ridvanozcan/AnimatedSplashScreen>

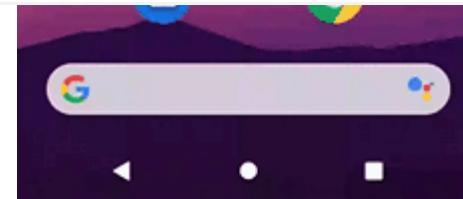
Aperçu



Recherche



Écrire



Si vous avez des questions ou des commentaires, n'hésitez pas à me contacter. Cordialement, Ridvan Özcan.

ridvanozcan — Aperçu

Vous ne pouvez pas effectuer cette action pour le moment. Vous vous êtes connecté avec un autre onglet ou une autre fenêtre. You...

github.com

<https://www.linkedin.com/in/ridvanozcan/>

<https://www.instagram.com/mr.softwareengineer/>

#kotlin #kotlindeveloper #android #androiddeveloper
#softwaredevelopment #androiddevelopment #kotlinandroid #developers
#developercommunity #softwareengineer #googledevs #googlegroups
#androidgroups #googleandroid #gradle #gradleandroid #community
#linkedinforcreators #creators #linkedin #linkedinlearning #linkedinfamily
#linkedinconnections #animated #splashscreen #lottie

Lottie

Écran de démarrage

Android

Composer

Lottie Animation



Écrit par Ridvan Özcan

107 abonnés

Développeur Android | Sahibinden.com

Suivre

Plus de Ridvan Özcan



Google Maps



Jetpack
Compose



Ridvan Özcan



Ridvan Özcan

Comment utiliser Google Maps dans Jetpack Compose : guide...

Dans ce tutoriel, nous vous guiderons dans l'intégration de Jetpack Compose et de...

6 minutes de lecture · 27 mars 2024



57



Ridvan Özcan

Entretien rapide sur le cycle de vie des activités Android

Comprendre le cycle de vie des activités

4 minutes de lecture · 17 avril 2024



28



Google Play Store vous propose 20 tests politiques et pratiques?

Google Play Store vous permet d'obtenir des informations sur les produits qui vous...

3 minutes de lecture · 25 décembre 2023



36



Ridvan Özcan

Configurer Charles Proxy pour les émulateurs Android / MacOS

Les informations fournies dans la section d'aide des émulateurs Charles Proxy pour...

6 minutes de lecture · 19 avril 2023

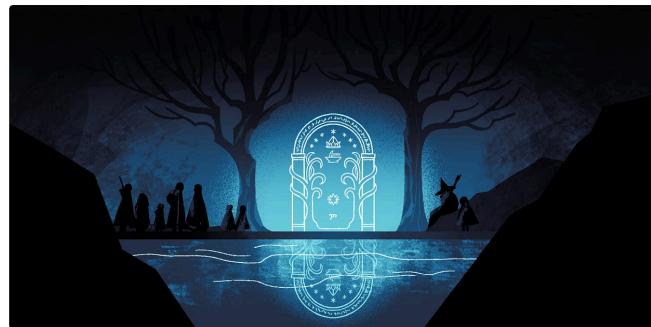


104



Tout voir de Ridvan Özcan

Recommandé à partir de Medium



Samed Harman

Android Compose - Crédit à l'API...

Bonjour, Dans cet article, nous allons créer cet exemple d'écran de démarrage dans...

5 minutes de lecture · 25 décembre 2023



58



...



sed magna elit....

Lore ipsum dolor sit amet, consectetur adipiscing elit. In sed magna elit....



Lore ipsum dolor sit amet,



Francesc Vilarino Güell

Transitions d'éléments partagés dans Jetpack Compose

Introduction

7 minutes de lecture · 21 avril 2024



455



2



...

Listes



Choix du personnel

634 histoires · 934 sauvegardes

Jetpack Compose

Setting Up Material Theme Color Schemes in Jetpack Compose



<https://medium.com/@rowaido.game>



Michihiro Iwasaki

Configuration des schémas de couleurs du thème matériel dans...

Simplifier les décisions de couleur grâce à la conception matérielle

6 minutes de lecture · 28 novembre 2023



62



Tdkms

Changement de thème dans Android Jetpack Compose

Dans ce tutoriel, je vais vous apprendre à basculer entre les thèmes dans Android...

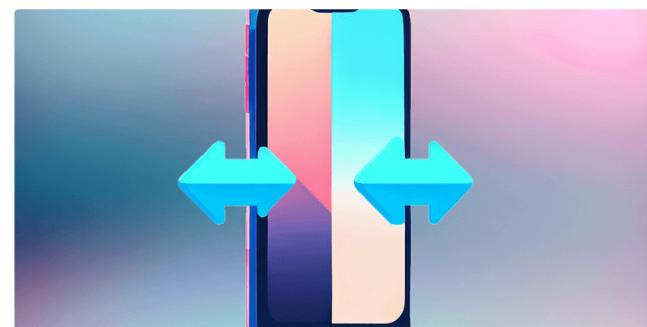
3 minutes de lecture · 26 février 2024



Jetpack Compose Series

Nested Lazy Column

Narayan Panthi dans Kt. Académie



Tomas Repčík

LazyColumn imbriquée dans Jetpack Compose

Lors de l'affichage de groupes d'éléments, nous utilisons généralement des colonnes e...

7 minutes de lecture · 17 avril 2024



169



4



...

Animations de composition et de transition d'écran Jetpack

Les animations de composition de l'accompagnateur sont devenues obsolètes...

5 minutes de lecture · 4 novembre 2023



146



...

Voir plus de recommandations

[Aide](#) [Statut](#) [À propos](#) [Carrières](#) [Blog](#) [Confidentialité](#) [Termes](#) [Texte pour parler](#) [Équipes](#)