

translated by Google

Cette page a été traduite par l'API Cloud Translation
([//cloud.google.com/translate/?hl=fr](https://cloud.google.com/translate/?hl=fr)).

Automation Ready

Switch to English

Images et éléments graphiques



Bien que votre application puisse ne contenir que du texte et des couleurs, vous pouvez ajouter d'autres éléments visuels, tels qu'un logo ou une illustration. Android propose des bonnes pratiques spécifiques pour ajouter des éléments graphiques à votre application, ainsi que différentes bibliothèques pour créer des effets graphiques ou ajouter du mouvement.

Un élément Android est appelé drawable

(<https://developer.android.com/reference/android/graphics/drawable/Drawable?hl=fr>), un type de ressource qui est dessiné à l'écran. Cela inclut, sans s'y limiter, les bitmaps, les formes et les vecteurs.

Lorsque vous créez des images et des graphiques, tenez compte des points suivants :

- Évitez d'inclure du texte immuable dans les composants.
- Utilisez d'abord les formats vectoriels, dans la mesure du possible.
- Fournissez des composants pour les buckets de résolution.
- Fournissez un voile suffisant entre les images de fond et le texte.
- Bien qu'Android soit capable de produire différents effets d'image, comme des dégradés, des colorisations et des flous, certains sont plus coûteux en termes de performances.
- Les drawables vectoriels animés fournissent un format évolutif pour les petites animations d'interface utilisateur.

Exporter des composants pour Android

- Mettez les noms des composants en minuscules.
- Définissez les éléments simplistes à exporter au format SVG.
- Définissez les images complexes, comme les photos, pour qu'elles soient exportées au format WebP, PNG ou JPG.
- Définissez la bonne mise à l'échelle de la résolution, comme indiqué dans le tableau suivant.



Taille de l'écran	Nombre d'utilisateurs
-------------------	-----------------------

mdpi	x1
------	----

hdpi	x1,5
------	------

xhdpi	x 2
-------	-----

xxhdpi	x3
--------	----

xxxhdpi	x4
---------	----

Vous pouvez également convertir des SVG en drawables vectoriels (VD) à l'aide d'Android Studio. Organisez les composants dans des répertoires correspondant à la résolution pour le transfert, comme illustré dans l'image suivante. Par exemple, incluez la taille de l'écran dans les noms de dossiers.

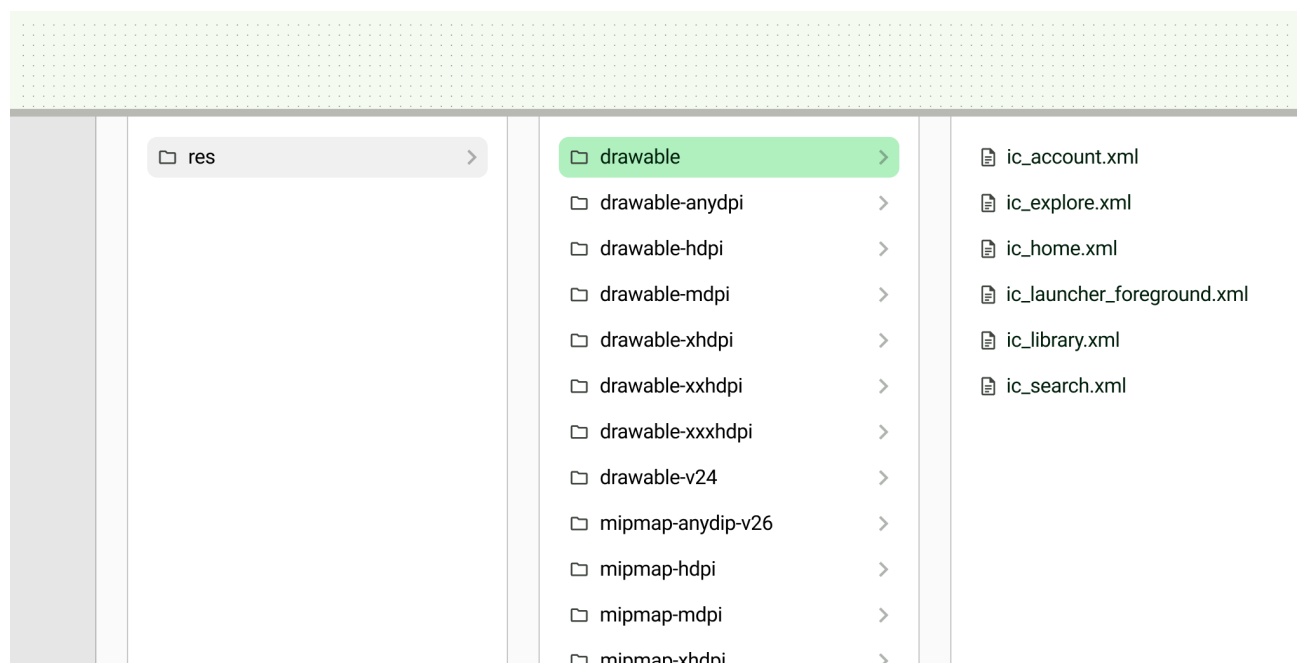


Figure 2. Répertoire res organisé.

Types de ressources

Android est compatible avec les types de composants suivants.

Vectoriel

Un graphique vectoriel est un format sans perte, ce qui signifie qu'il ne perd pas d'informations visuelles lorsqu'il est mis à l'échelle. Les vecteurs sont composés de points mathématiques qui sont remplis pour créer une image.

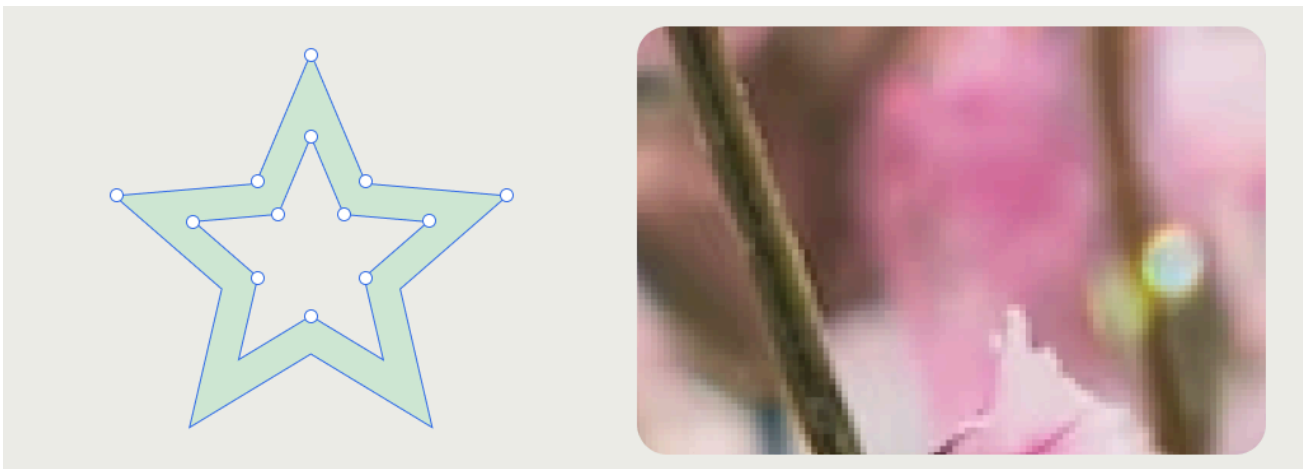


Figure 3. L'image de gauche montre une image composée de points Bézier vectoriels, par opposition à une image matricielle agrandie à droite.


Formats vectoriels

Android est compatible avec les formats d'image vectorielle suivants : SVG et Vector Drawables.

Les drawables vectoriels ressemblent aux SVG, mais sont basés sur XML. Ils sont également compatibles avec différents attributs, comme les dégradés. Pour en savoir plus sur les fonctionnalités compatibles, consultez [VectorDrawable](https://developer.android.com/reference/android/graphics/drawable/VectorDrawable?hl=fr) (<https://developer.android.com/reference/android/graphics/drawable/VectorDrawable?hl=fr>). Vous pouvez convertir des SVG en Vector Drawables à l'aide de [Vector Asset Studio](https://developer.android.com/studio/write/vector-asset-studio?hl=fr#running) (<https://developer.android.com/studio/write/vector-asset-studio?hl=fr#running>).

Cas d'utilisation

En raison de leur petite taille, il est préférable de créer des icônes au format vectoriel. Un [drawable vectoriel animé](#)

(<https://developer.android.com/reference/android/graphics/drawable/AnimatedVectorDrawable?hl=fr>) peut être utilisé pour ajouter du mouvement à une icône.  Automation Ready

- Les illustrations sont des éléments graphiques qui aident à orienter les utilisateurs, à expliquer des concepts ou à exprimer des idées. Elles expriment généralement le style de la marque.
- Les illustrations de bannière sont très importantes par rapport au reste du contenu. Elles permettent de définir l'aspect général et d'expliquer les informations principales.
- Les illustrations ponctuelles sont plus petites, généralement intégrées, et complètent le contenu environnant.

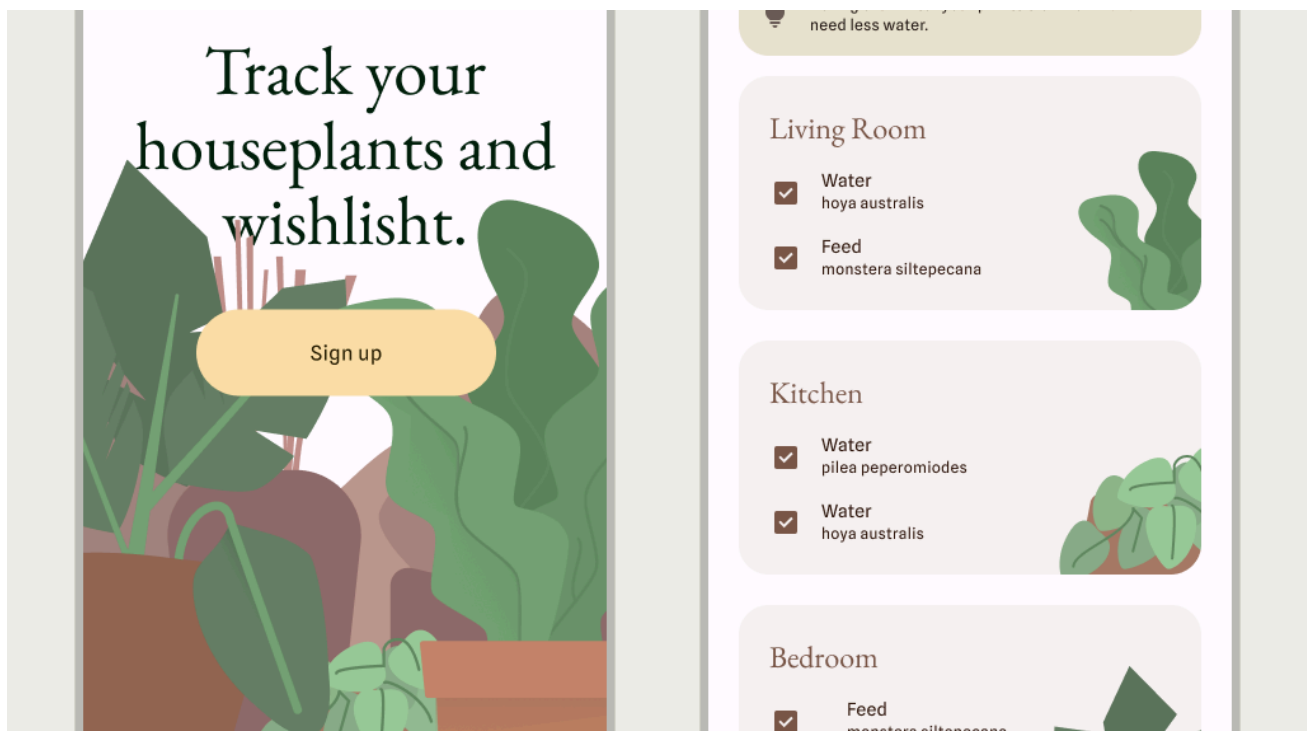


Figure 4 : Illustration principale et illustration intégrée.

Trame

Un graphique avec perte, ou un graphique qui perd des détails lorsqu'il est manipulé par compression ou mise à l'échelle, est composé de pixels qui forment l'image. Les graphiques raster sont généralement utilisés pour les images détaillées telles que les photos ou les dégradés complexes. Comme elles perdent en détail lorsqu'elles sont mises à l'échelle, exportez-les à plusieurs résolutions.

Formats raster

Android prend en charge les formats d'image raster suivants : PNG, GIF, JPG et WebP.

Cas d'utilisation

Les cas d'utilisation incluent les images qui présentent une gamme de textures entraînant une large palette de couleurs et une gradation, ou les images qui auraient un ensemble de points de Bézier trop complexe. Les cas d'utilisation peuvent également inclure des composants photo très détaillés, tels que des photos de produits, des informations sur les lieux, etc.

Taille

Lorsque vous créez des composants, tenez compte des points suivants.

Buckets de résolution

Votre application doit fournir des graphiques bitmap basés sur des plages ou des groupes de densité d'écran. Le système d'exploitation affiche automatiquement le bon graphique sur l'appareil en question à l'aide de ces buckets. Assurez-vous que des graphiques haute fidélité s'affichent sur tous les appareils en fournissant des composants pour chaque bucket.

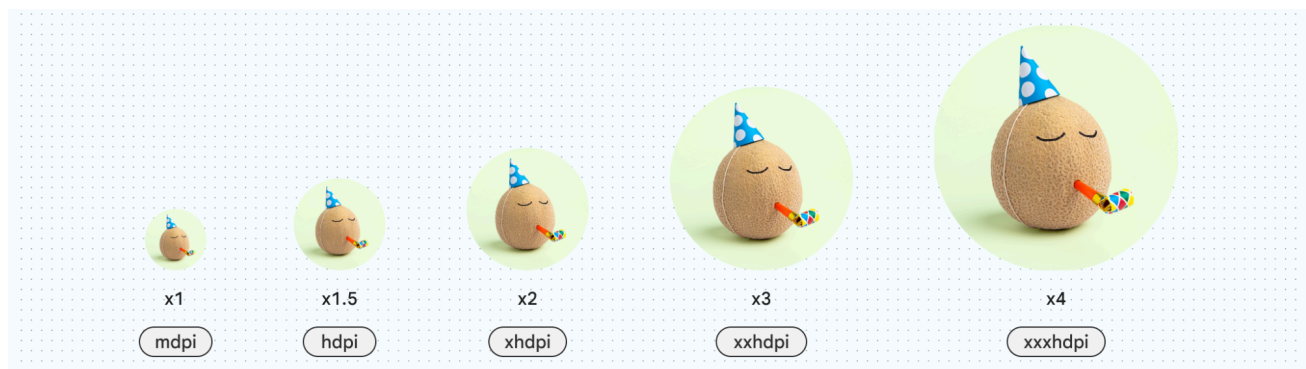


Figure 5 : Cantaloups de fête dans leurs densités respectives et à l'échelle pour l'exportation.

Remarque : Vous n'avez pas besoin de fournir des drawables vectoriels pour toutes les densités, car les vecteurs sont automatiquement mis à l'échelle en fonction de la densité.

Marges intérieures

Les icônes et les petits composants similaires doivent inclure une marge intérieure (intégrée) pour leur donner suffisamment d'espace de cible tactile et assurer une taille cohérente.

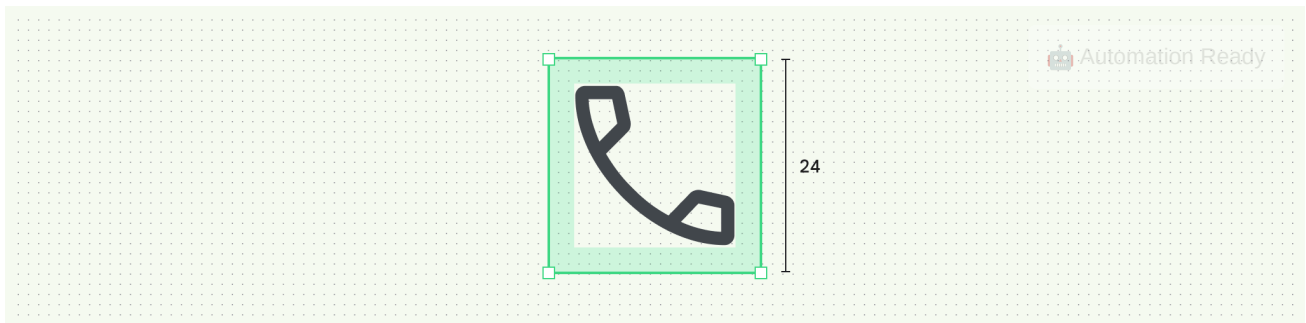


Figure 6 : Icônes de 24 dp avec une marge intérieure intégrée aux composants.

Noms des fichiers

Les composants Android sont en minuscules et n'incluent pas de suffixe de résolution.

Adoptez une convention de nommage et une structure cohérentes pour organiser vos fichiers et vos projets. Par exemple, nommer les icônes avec le préfixe "ic_..." peut vous aider à organiser toutes les icônes de votre projet et à les identifier rapidement pendant le développement.

Autres composants Application

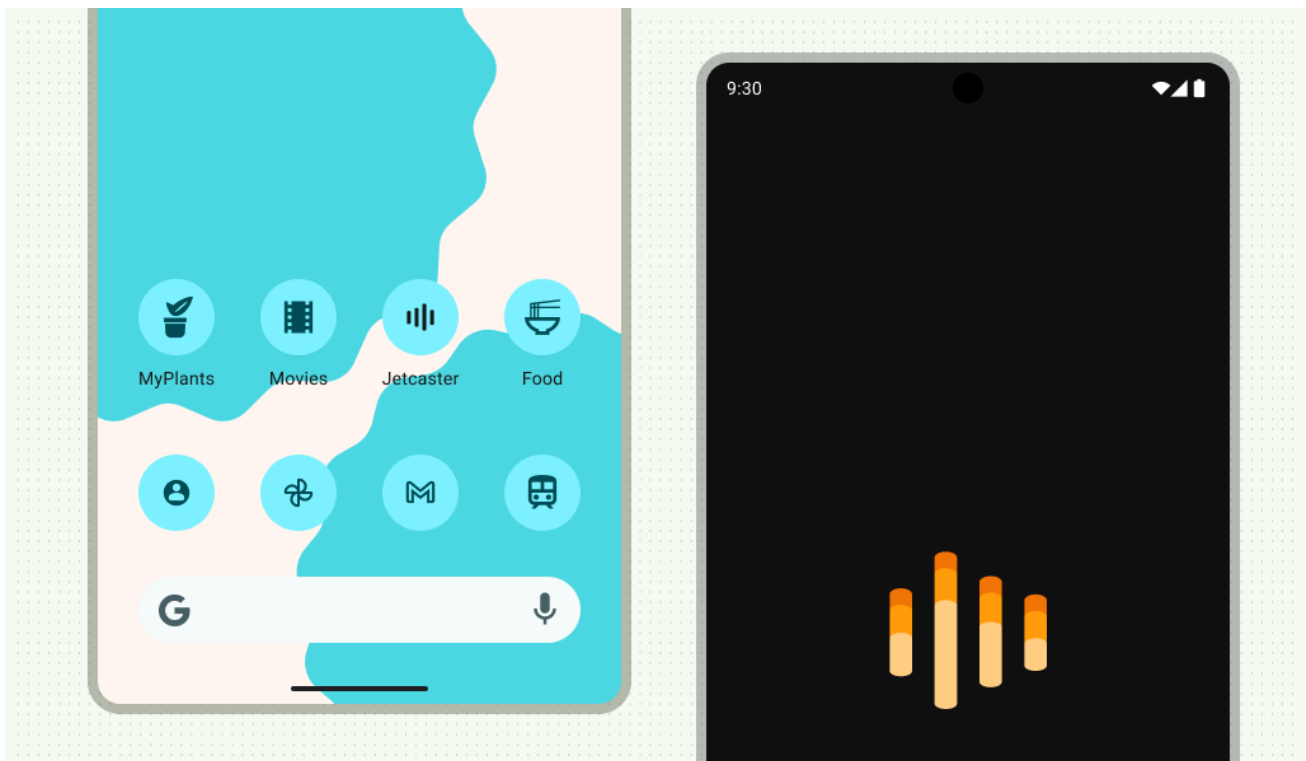


Figure 7 : Icônes de lanceur et écrans de démarrage monochromes.

Icônes d'application

Les icônes du lanceur d'applications se trouvent sur l'écran d'accueil. Recherchez les icônes monochromes dans l'interface utilisateur du système, y compris les notifications, la barre d'état et les widgets monochromes.

Mettez en forme les icônes d'application

(https://developer.android.com/develop/ui/views/launch/icon_design_adaptive?hl=fr) en tant que drawables vectoriels pour les icônes adaptatives et au format PNG pour les anciennes icônes. Pour en savoir plus sur la création et la prévisualisation de l'icône de votre application, consultez Concevoir et prévisualiser les icônes de votre application (<https://codelabs.developers.google.com/design-android-launcher?hl=fr>).

Écrans de démarrage

À partir d'Android 12, votre application peut afficher un écran de démarrage

(<https://developer.android.com/develop/ui/views/launch/splash-screen?hl=fr>) animé qui présente l'icône de l'application pendant son ouverture.

Emplacement


Notez comment les images doivent être mises à l'échelle

(<https://developer.android.com/design/ui/mobile/guides/layout-and-content/layout-basics?hl=fr>) et positionnées sur l'écran. **Ajuster, Recadrer, FillHeight, FillWidth, FillBounds, Inside** et **Aucune** sont disponibles pour définir la mise à l'échelle

(<https://developer.android.com/develop/ui/compose/graphics/images/customize?hl=fr#content-scale>) d'une image.



Figure 8 : Exemples de recadrage.

Vous pouvez également découper des images selon une forme pour créer des effets supplémentaires.  Automation Ready

Recadrage responsif

Pour afficher des images de manière responsive, définissez la façon dont elles seront recadrées pour différentes plages de points d'arrêt. À différents niveaux de points d'arrêt, le recadrage peut :

- Maintenez un ratio fixe.
- S'adapter à différents ratios.
- Conservez des hauteurs d'image fixes.

Conserver un ratio

Le dimensionnement des images peut conserver un ratio fixe pour les différentes plages de points d'arrêt.

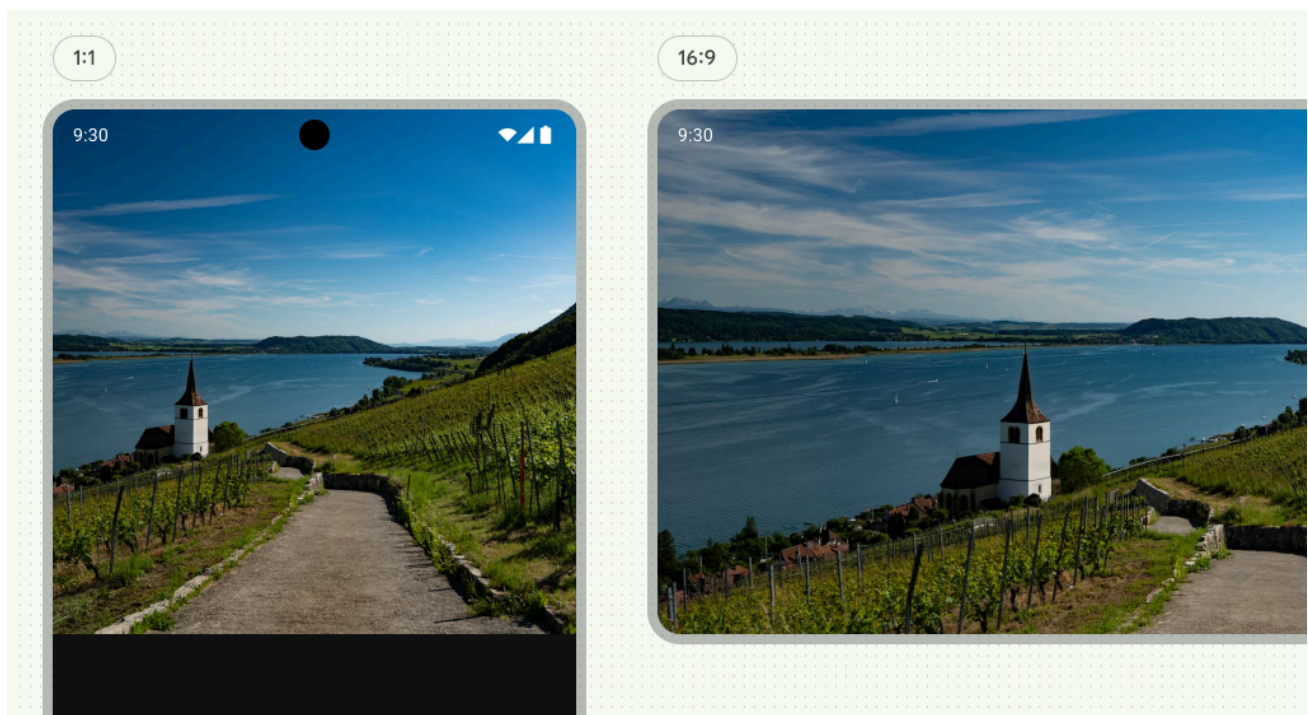


Figure 9. Image affichée dans différents formats.

S'adapter à différents ratios

Les ratios d'image peuvent changer en s'adaptant à différentes plages de points d'arrêt. Par exemple, dans la figure 9, le format de l'image passe de 1:1 à 16:9 entre les points d'arrêt.

Hauteurs d'image fixes

La taille des images peut conserver une hauteur fixe et une largeur fluide dans les plages de points d'arrêt et entre elles. L'image conserve une hauteur fixe, tandis que la largeur entre les points d'arrêt est fluide.

Effets

Android inclut divers effets d'image intégrés. Voici quelques effets courants :

Gradients

Dans Compose, utilisez un pinceau

(<https://developer.android.com/reference/kotlin/androidx/compose/ui/graphics/Brush?hl=fr>) pour dessiner quelque chose à l'écran. Différents pinceaux peuvent être utilisés pour dessiner des formes de différentes couleurs ou des dégradés. Utilisez les pinceaux de dégradé

(<https://developer.android.com/develop/ui/compose/graphics/draw/brush?hl=fr#gradient-brushes>) intégrés pour obtenir différents effets de dégradé. Ces pinceaux vous permettent de spécifier la liste des couleurs à partir desquelles vous souhaitez créer le dégradé.

Les pinceaux de dégradé permettent de créer des dégradés plus avancés en ajoutant des arrêts de couleur et des mosaïques, à condition de fournir la liste des couleurs et les pourcentages où l'arrêt se produit. Utilisez des conteneurs ou des formes pour recadrer des dégradés, des remplissages de couleur unie ou des images.



Figure 10. Traduisez un dégradé depuis Figma à l'aide de modificateurs Compose.

Floutage

Appliquez des effets de flou aux images à l'aide de la méthode `Modifier.blur()` et en fournissant les ratios de flou. Utilisez les flous avec précaution, car ils peuvent affecter les performances et ne sont disponibles que sur les appareils équipés d'Android 12 ou version ultérieure. Pour en savoir plus, consultez Composable de flou d'image

(https://developer.android.com/develop/ui/compose/graphics/images/customize?hl=fr#blur_an_image_composable)

.

Modes de fusion

Android est capable de modifier des images à l'aide d'opérations booléennes et de modes de fusion similaires à ceux que vous pouvez trouver dans les logiciels de conception, comme l'union ou la multiplication. Pour en savoir plus, consultez [BlendMode](https://developer.android.com/reference/android/graphics/BlendMode?hl=fr) (<https://developer.android.com/reference/android/graphics/BlendMode?hl=fr>).

Teinte

Utilisez des modes de fusion et des remplissages pour teinter les composants. Cela vous permet d'avoir un seul composant et d'y appliquer différentes couleurs, ce qui peut réduire le nombre de composants produits.

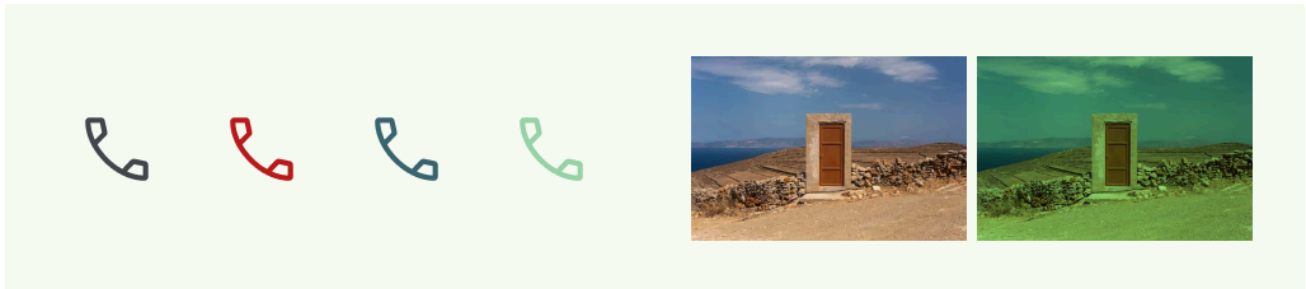


Figure 11. Ressources avec différentes teintes utilisées pour compléter les couleurs du contenu ou pour indiquer différents états.

Mouvement

Les graphiques peuvent être animés de manière programmatique pour créer des animations graphiques au lieu d'importer des fichiers de mouvement. Cela peut permettre une plus grande flexibilité et des ressources d'éléments plus petites.

Les drawables vectoriels animés vous permettent de créer de petites animations d'UI. Sinon, découvrez comment animer avec des images clés dans [Compose](https://developer.android.com/develop/ui/compose/graphics?hl=fr) (<https://developer.android.com/develop/ui/compose/graphics?hl=fr>). Pour en savoir plus sur les effets d'image, consultez [Personnaliser une image](https://developer.android.com/develop/ui/compose/graphics/images/customize?hl=fr#border-image) (<https://developer.android.com/develop/ui/compose/graphics/images/customize?hl=fr#border-image>).

Le contenu et les exemples de code de cette page sont soumis aux licences décrites dans la [Licence de contenu](https://developer.android.com/license?hl=fr) (<https://developer.android.com/license?hl=fr>). Java et OpenJDK sont des marques ou des marques déposées d'Oracle et/ou de ses sociétés affiliées.

Dernière mise à jour le 2025/08/28 (UTC).