

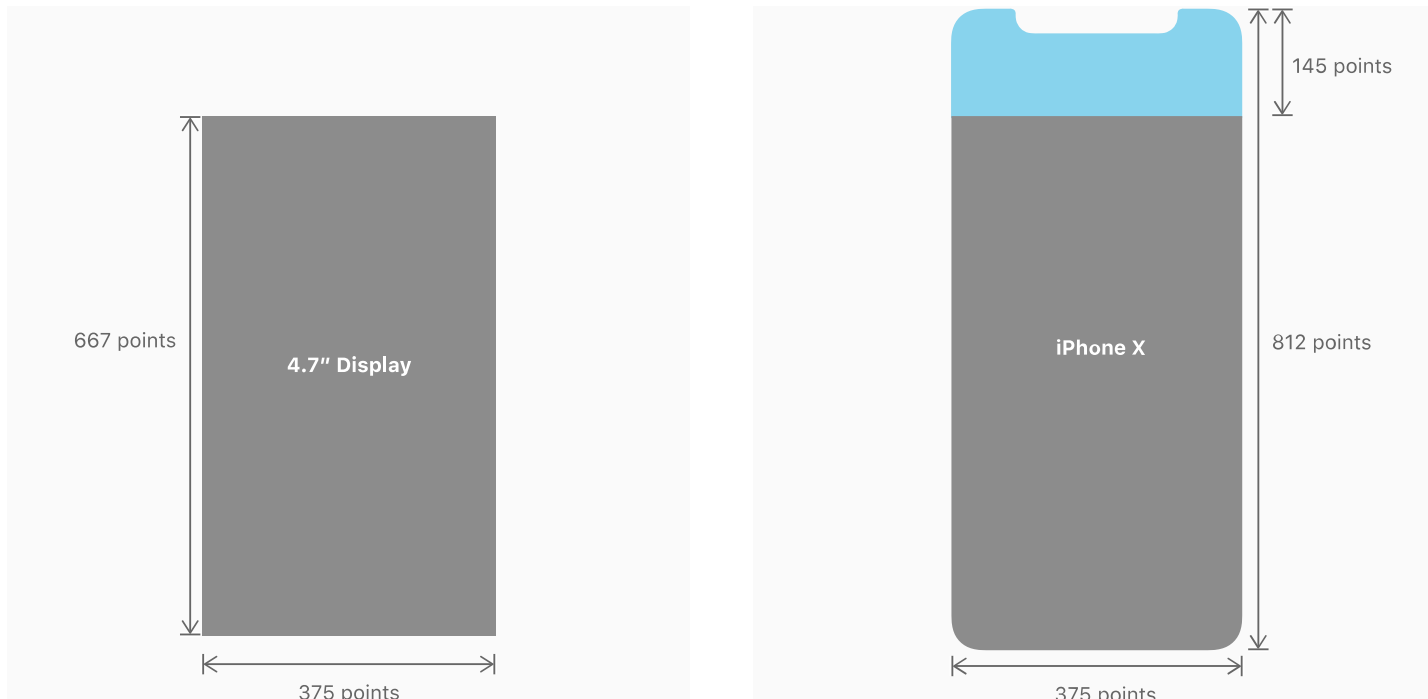
# iPhone X

iPhone Xには、大型で丸みを帯びた、本体の端から端まで広がる高解像度のSuper Retinaディスプレイが搭載され、今までにないほどの没入感でコンテンツを楽しめます。



## 画面サイズ

縦向きの場合、iPhone Xディスプレイの幅はiPhone 6、iPhone 7、iPhone 8の4.7インチディスプレイの幅と同じです。縦方向に145pt広がっているため、コンテンツの表示領域が約20%増加しています。



#### 縦向き表示のサイズ

1125px × 2436px (375pt × 812pt @3x)

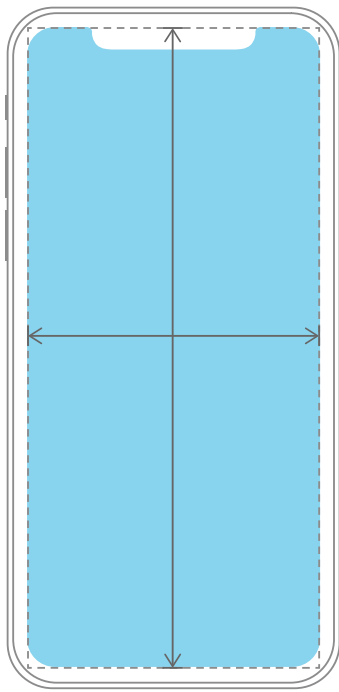
#### 横向き表示のサイズ

2436px × 1125px (812pt × 375pt @3x)

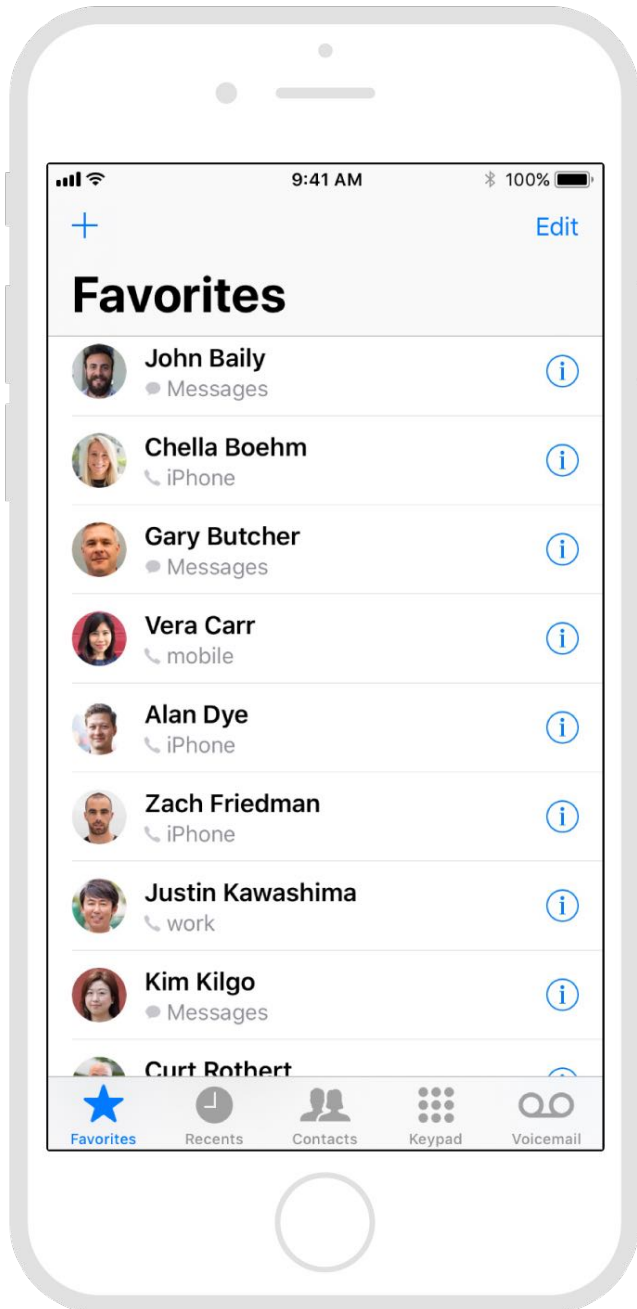
**アプリケーションのすべてのアートワークについて、高解像度の画像を用意する。** 高解像度のSuper Retina ディスプレイを搭載したiPhone Xでは、拡大縮小倍率が@3xとなっています。グリフをはじめとするフラットなベクトルアートワークについては、解像度に依存しないPDFを用意することを推奨します。ラスタアートワークについては、@3xと@2xの両バージョンのアートワークを用意します。詳しくは、「画像のサイズと解像度」および「カスタムアイコン」を参照してください。

## レイアウト

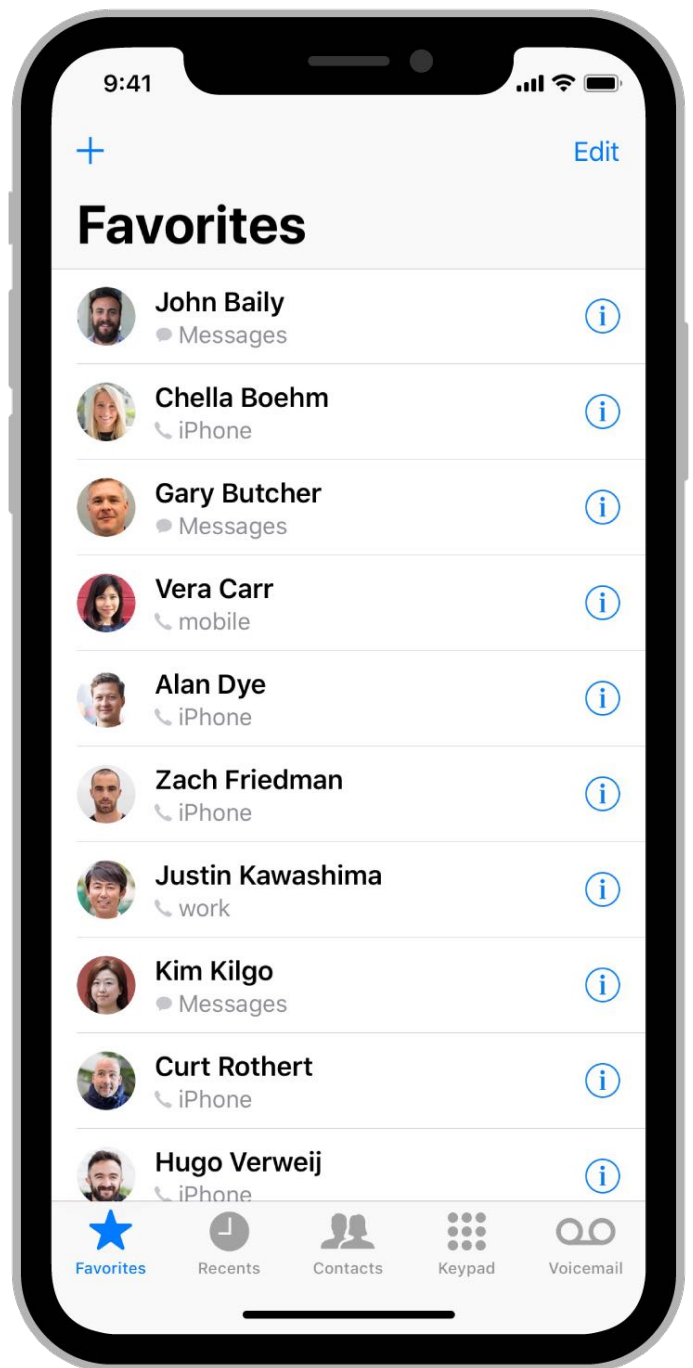
iPhone X用のデザインを検討する際は、コンテンツを画面いっぱいにレイアウトし、丸みを帯びた画面四隅、センサーハウジング、ホーム画面移動用のインジケータで要素が隠れないようにすることが必要です。



ナビゲーションバー、テーブル、コレクションなど、システムに用意されている標準のUI要素を使ったアプリケーションでは、ほとんどの場合、iPhone Xの新しいフォームファクタに合わせて表示が自動的に調整されます。背景素材はディスプレイの周縁部まで表示されるように拡大され、UI要素は、余白（inset）を適切に挿入した上で配置されます。



4.7インチiPhone



iPhone X

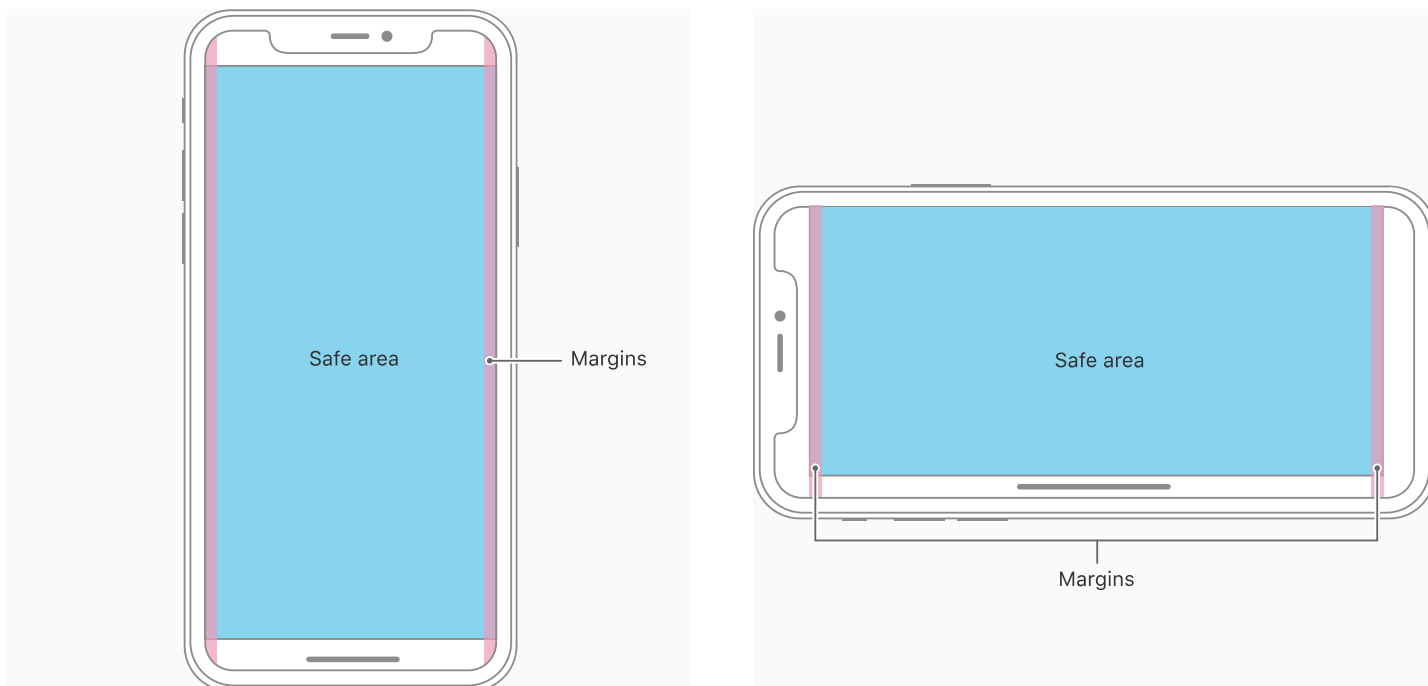
独自のレイアウトを採用しているアプリケーションの場合も、比較的容易にiPhone Xをサポートできます。アプリケーションでAuto Layoutを利用し、Safe Areaとマージンに関するレイアウトガイドを遵守している場合は、特に容易になります。

**アプリケーションをiPhone Xでプレビューする。** Xcodeに含まれているシミュレータを使ってアプリケーションをプレビューすると、画面要素の欠落など、レイアウト上の問題がないかどうかを確認できます。アプリケーションで横向きモードをサポートする場合は、デバイスを左右のどちらに回転したときも、レイアウト上の問題が生じないことを確認してください。上下逆のポートレートモードは、iPhone Xではサポートされません。広色域での画像表示など、一部の機能については実機でプレビューすることを推奨します。

**フルスクリーンでの利用に対応する。** 背景画像がディスプレイの周縁部まで表示されること、縦スクロール可能なレイアウト（テーブルやコレクションなど）で要素が末尾まで表示されることを確認します。

**重要なコンテンツは、余白（inset）を使って表示の欠落を防ぐ。** 一般的に、コンテンツは中央揃えで左右対称に配置して、どの向きでも適切に表示されるようにし、画面の四隅やデバイスのセンサーハウジングで表示が途切れたり、ホーム画面アクセス用のインジケータで要素が隠れたりしないようにします。最適な形で表示されるようにするには、システムに用

意されている標準のインターフェイス要素とAuto Layoutを使ってインターフェイスを構成します。アプリケーションでは、UIKitで規定されているSafe Areaとレイアウトマージンを遵守して、デバイスやコンテンツに応じて適切に余白が挿入されるようにします。Safe Areaを遵守すると、ステータスバー、ナビゲーションバー、ツールバー、タブバーの背後にコンテンツが隠れることもなくなります。

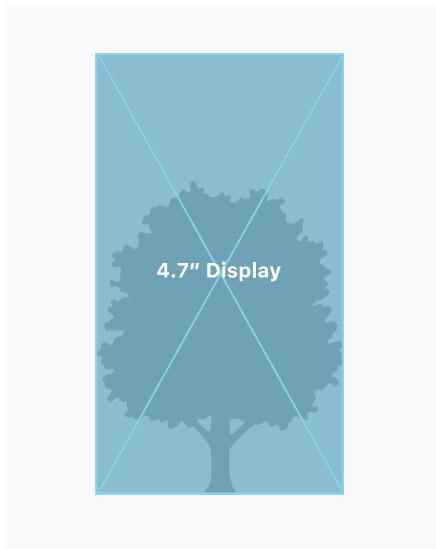


デバイスが横向きモードのとき、ゲームなど一部のアプリケーションでは、タップ可能なコントロールを画面の下側に配置して（Safe Areaを越えて下まで拡張）、コンテンツの表示領域を広く取るほうが妥当な場合もあります。コントロールを画面の最上部や最下部に配置する場合は、対応する余白（inset）を挿入して、ホームインジケータの周囲に十分なスペースを確保し、ユーザーがコントロールで操作しようとしたときに誤タップが生じないようにします。

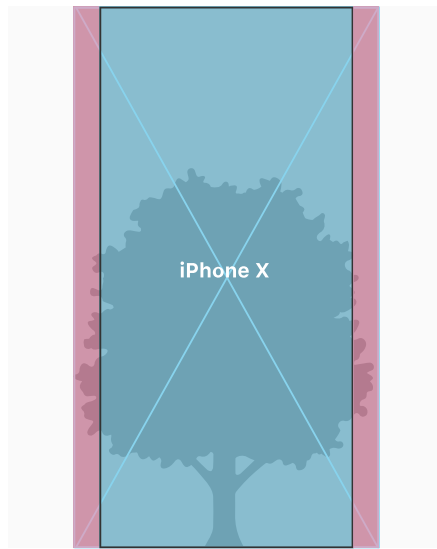


**ステータスバーの高さを念頭に置く。** iPhone Xのステータスバーは、ほかのiPhoneよりも高くなっています。コンテンツをステータスバーの下側に配置するとき、ステータスバーの高さが固定長であることを前提としている場合は、アプリケーションのコードを更新して、ユーザーのデバイスに応じて動的にコンテンツが配置されるようにする必要があります。iPhone Xのステータスバーは、音声録音や位置追跡などのバックグラウンドタスクがアクティブになっている場合に、高さが増えることはありません。

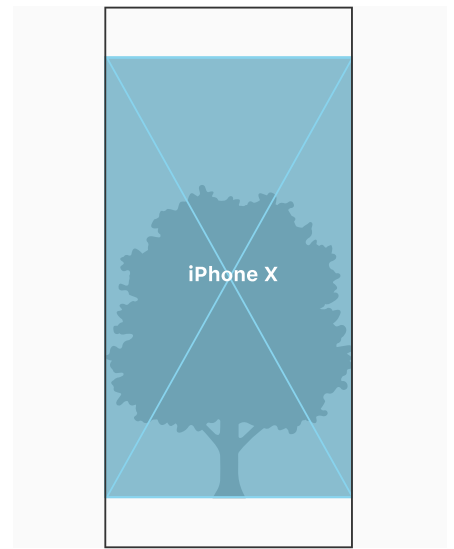
**アプリケーションでステータスバーを非表示にしている場合は、表示/非表示の判定をiPhone Xに委ねることを検討する。** このiPhoneでは、ディスプレイの高さが4.7インチiPhoneのディスプレイよりも大きく、コンテンツの表示領域が縦方向に広がっており、アプリケーションで十分に活用されていない可能性のある画面領域にステータスバーが表示されます。また、ステータスバーはユーザーにとって有用な情報が表示される領域でもあります。ステータスバーを非表示にするのは、何らかの付加価値が生まれる場合のみにしてください。



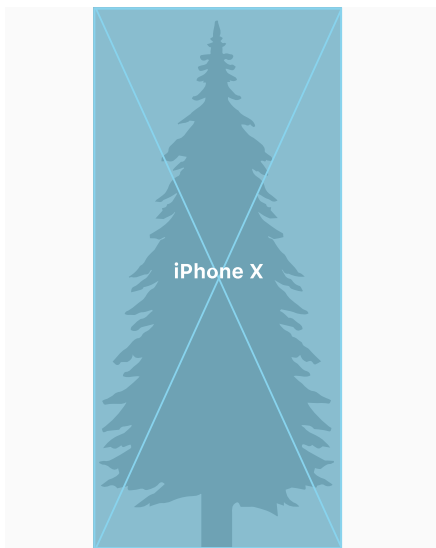
4.7インチデバイスのフルスクリーン表示画像



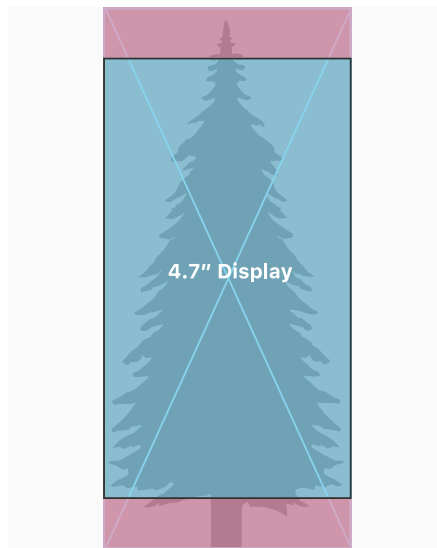
iPhone Xでのトリミング



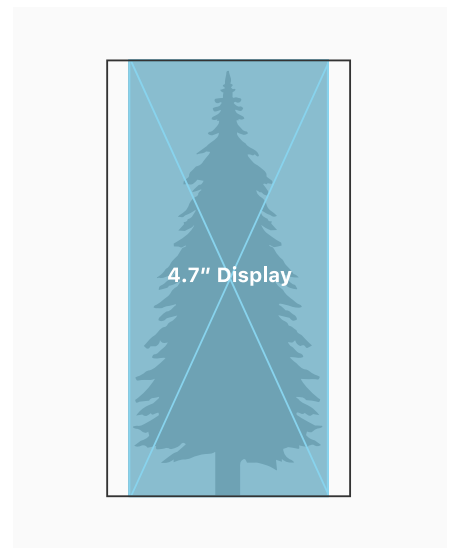
iPhone Xでのレターボックス表示



iPhone Xのフルスクリーン表示画像



4.7インチデバイスでのトリミング

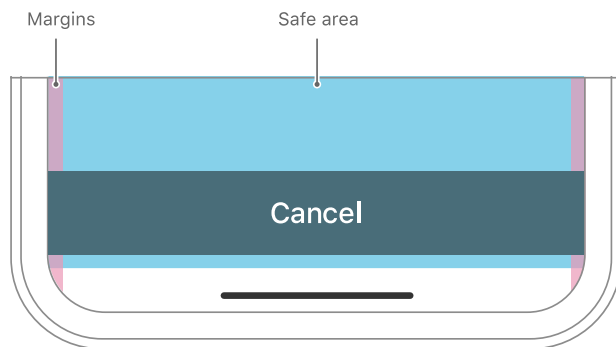
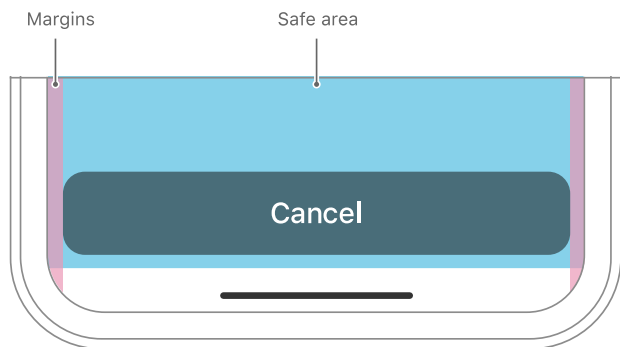


4.7インチデバイスでのピラーボックス表示

**既存のアートワークを再利用する場合は、アスペクト比の違いを念頭に置く。** iPhoneは、4.7インチiPhoneとはアスペクト比が異なります。結果として、4.7インチiPhoneでフルスクリーン表示に使うアートワークをiPhone Xでフルスクリーン表示すると、トリミングされるか、レターボックス表示になります。同様に、iPhone Xでフルスクリーン表示に使うアートワークを4.7インチiPhoneでフルスクリーン表示すると、トリミングされるか、ピラーボックス表示になります。重要な視覚コンテンツが、どちらのディスプレイサイズでも引き続き表示されることを確認してください。

**画面の最下部や四隅に、ユーザーとのやり取りに使われるコントロールを明示的に配置することは避ける。** ディスプレイの下端は、ホーム画面とアプリケーションスイッチャーにアクセスするためのスワイプジェスチャーで使われます。この領域に実装したカスタムジェスチャーは、これらのスワイプジェスチャーによって無効になる可能性があります。画面の四隅は、指でたどり着きにくい領域になり得ます。

**画面幅いっぱいのボタンには余白を設定する。** ボタンは、画面の端にまで広がっていると、ボタンのように見えない場合があります。画面幅いっぱいのボタンについては、UIKitで標準とされているマージンを両側に設定してください。画面幅いっぱいのボタンを画面の一番下に最適な形で表示するには、ボタンの四隅に丸みを持たせて、Safe Areaの下端に沿って配置します。このようにすると、ホームインジケータと重なることもなくなります。



**ディスプレイ上の重要な機能はマスクしない。または細心の注意を払う。** デバイスの丸みを帯びた四隅、センサーハウジング、ホーム画面アクセス用のインジケータについては、画面の最上部や最下部に黒色のバーを表示して覆い隠そうとしないでください。また、ブラケット、ベゼル、型材、指示テキストなどの視覚的な装飾は使わず、これらの領域の処置には細心の注意を払ってください。

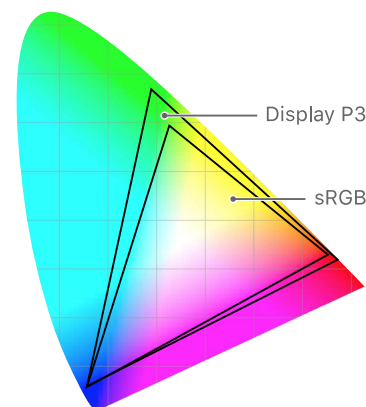
**ホーム画面アクセス用インジケータの自動非表示を許可することは控える。** 自動非表示が有効になっていると、ユーザーが一定の秒数にわたって画面にタッチしない場合、インジケータが非表示になります。ユーザーが画面にタッチすると、インジケータが再び表示されます。この挙動を有効にするのは、ビデオの再生や写真のスライドショーなど、ユーザーによる操作があまり生じない表示形態の場合のみにしてください。

詳しくは、「アダプティビティとレイアウト」を参照してください。

## 色

iPhone Xのディスプレイは、sRGBよりも色彩が豊かで鮮明な表示を実現する、P3の色空間に対応しています。

**広い色域を活かして、視覚体験をいっそう豊かなものにする。** 広い色域を活かすことで、写真やビデオはいっそう真に迫るものになり、視覚的なデータやステータスインジケータは、いっそう強い印象を与えるものになります。詳しくは、「カラーマネジメント」を参照してください。



## ビデオ

システムに用意されているビデオプレーヤーには、フルスクリーン（アスペクト比維持で画面全体を埋める）と、画面の大きさに合わせる（アスペクト比維持）という2つの表示モードがあります。デフォルトでは、表示モードはビデオのアスペクト比に基づいて選択され、ユーザーは再生中にモードを切り替えることができます。デベロッパ向けのガイダンスについては、「AVPlayerViewController」を参照してください。

**フルスクリーン（アスペクト比維持で画面全体を埋める）表示モード：**ビデオが、ディスプレイ全体を埋めつくすように拡大されます。上下または左右の辺で、表示のトリミングが生じることがあります。ワイドビデオ（縦横比が2:1~2.40:1）の場合は、これがデフォルトの表示モードです。デベロッパ向けのガイダンスについては、

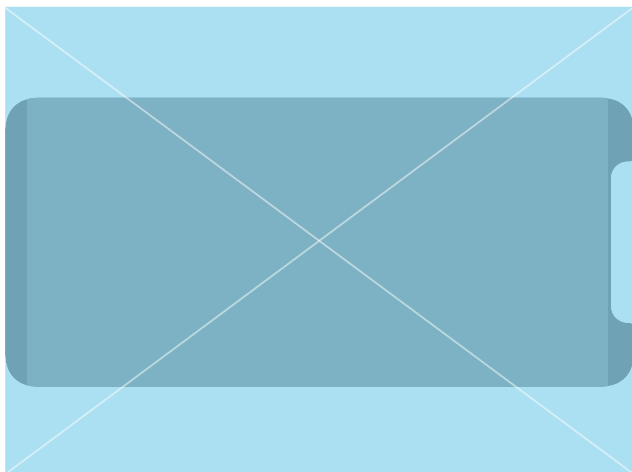
**画面の大きさに合わせる（アスペクト比維持）表示モード：**ビデオの映像全体が欠けることなく画面に表示されます。レターボックス表示またはピラーボックス表示になります。標準ビデオ（4:3、16:9、および2:1までのすべての縦横比）およびウルトラワイドビデオ（縦横比が2.40:1を超えるもの）の場合は、これがデフォルトの表示モードで

「resizeAspectFill」を参照してください。

す。デベロッパ向けのガイダンスについては、「resizeAspect」を参照してください。

#### 4:3ビデオ

---



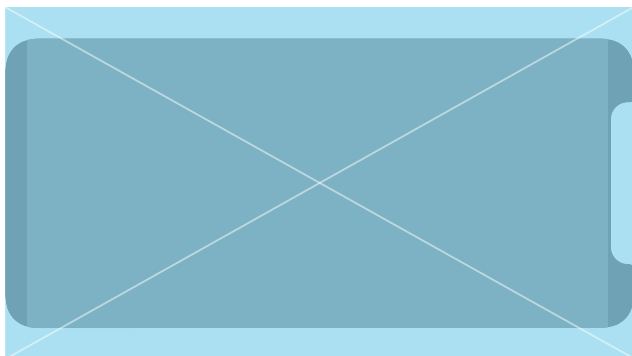
フルスクリーン表示モードの4:3ビデオ



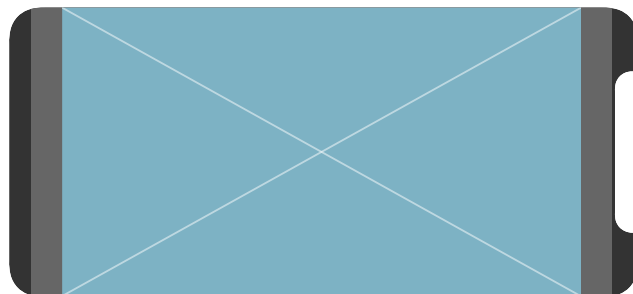
画面の大きさに合わせる表示モードの4:3ビデオ（デフォルト）

#### 16:9ビデオ

---



フルスクリーン表示モードの16:9ビデオ

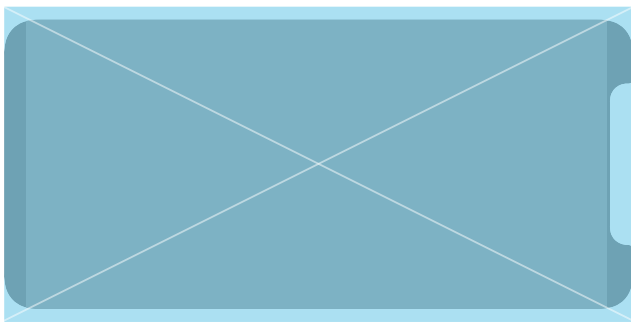


画面の大きさに合わせる表示モードの16:9ビデオ（デフォルト）

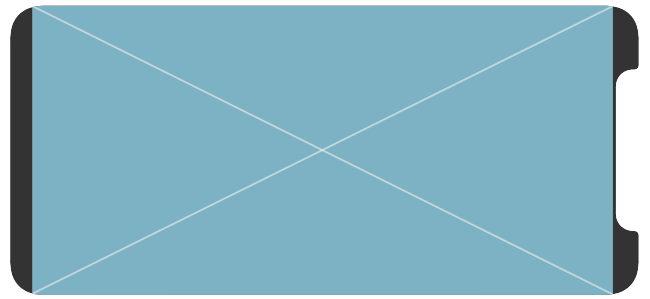
#### 2:1ビデオ

---



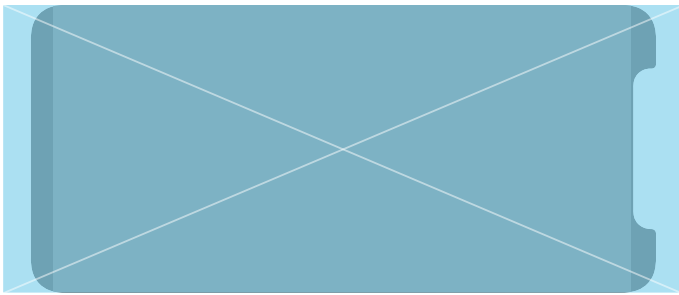


フルスクリーン表示モードの2:1ビデオ（デフォルト）

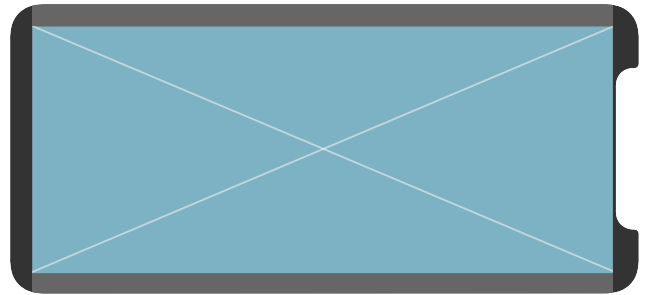


画面の大きさに合わせる表示モードの2:1ビデオ

## 21:9ビデオ



フルスクリーン表示モードの21:9ビデオ（デフォルト）



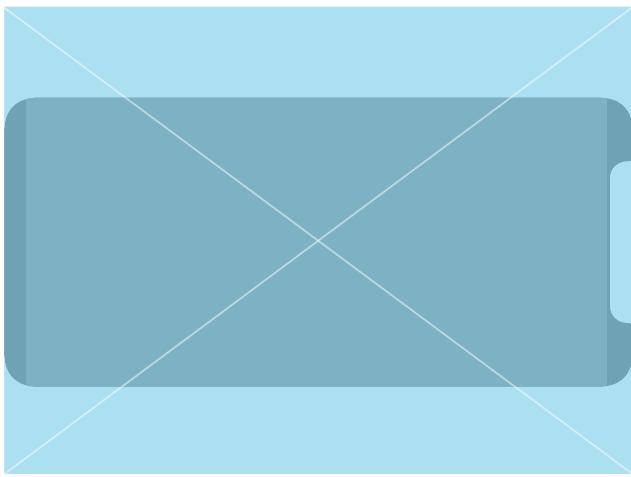
画面の大きさに合わせる表示モードの21:9ビデオ

● AVKit safe area ● Video

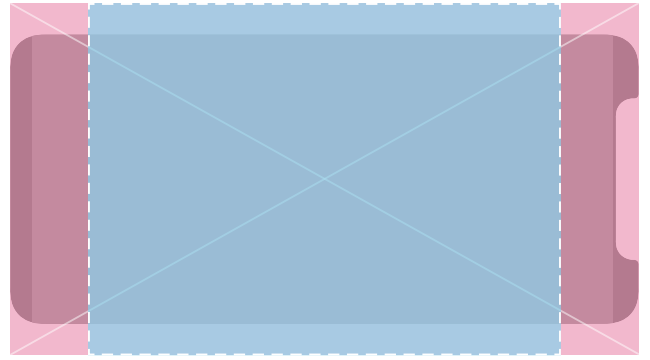
**独自のビデオプレーヤーが想定どおりに動作することを確認する。**確認の目的は、iPhone Xでビデオコンテンツが再生される際に、デフォルトでビデオをフルスクリーン表示することです。ただし、トリミングされる領域があまりにも大きくなる場合は、画面の大きさに合わせて表示します。また、フルスクリーン表示と画面の大きさに合わせる表示を、ユーザーが好みに応じて切り替えられるようにしてください。デベロッパ向けのガイダンスについては、「AVPlayerLayer」を参照してください。

**ビデオコンテンツは、常に本来のアスペクト比で表示する。**特定のアスペクト比に合わせるため、レターボックス形式またはピラーボックス形式でスペースをあらかじめパディングしているビデオコンテンツの場合、iOSでは、ユーザーが選択する表示モードに応じてビデオを適切に拡大縮小できなくなります。ビデオのフレーム内にあらかじめパディングを施すと、フルスクリーンモードまたは画面の大きさに合わせるモードで、適切な大きさよりも小さく表示される可能性があります。また、iPadのPicture in Pictureモードなどで、フルスクリーンではないビデオが枠内いっばいに正しく表示されなくなります。

## 4:3のビデオにパディングが施されている場合

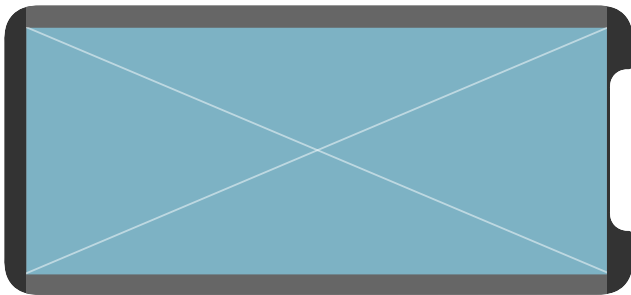


フルスクリーン表示モードの4:3ビデオ

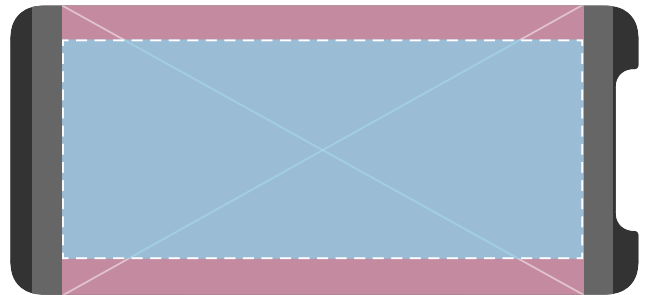


フルスクリーン表示モードの4:3ビデオ（パディングが施されている場合）

### 21:9のビデオにパディングが施されている場合



画面の大きさに合わせる表示モードの21:9ビデオ



画面の大きさに合わせる表示モードの21:9ビデオ（パディングが施されている場合）

● AVKit safe area    ● Video    ● Embedded padding

## ジェスチャー

iPhone Xのディスプレイでは、画面の端を使ったジェスチャーで、ホーム画面、Appスイッチャー、通知センター、コントロールセンターを表示することができます。

**システム全体にわたって有効になる、画面の端を使ったジェスチャーと干渉しないようにする。** このジェスチャーは、どのアプリケーションでも実行可能です。まれなケースとして、ゲームなど没入感のあるアプリケーションでは、システムジェスチャーよりも優先される画面の端を使ったジェスチャーが必要になる場合もあります。最初のスワイプではアプリケーション固有のジェスチャーが呼び出され、2回目のスワイプでシステムジェスチャーが呼び出されます。エッジプロテクトと

呼ばれるこの挙動は、ユーザーがシステムレベルのアクションにアクセスしにくくなるため、実装の頻度は控えめにしてください。デベロッパ向けのガイダンスについては、「[UIViewController](#)」の[preferredScreenEdgesDeferringSystemGestures\(\)](#)メソッドの項を参照してください。システムジェスチャーについては、「[ジェスチャー](#)」を参照してください。

## デザインに関するその他の考慮事項

**認証メソッドには正確に言及する。** iPhone Xは、Face IDでの認証に対応しています。Apple Payをはじめとするシステム認証機能をアプリケーションに組み込む場合、iPhone X上ではTouch IDに言及しないでください。同様に、Touch IDに対応するデバイス上では、アプリケーションでFace IDに言及していないことを確認してください。詳しくは、「[認証](#)」を参照してください。

**システムに用意されているキーボード機能と重複するものを実装しない。** iPhone Xでは、ユーザーがカスタムキーボードを使っている場合も、絵文字／グローブキーと音声入力キーがキーボードの下側に自動的に表示されます。アプリケーションからこれらのキーに影響を及ぼすことはできないため、これらのキーと同じものをカスタムキーボードに実装してユーザーを混乱させることは避けてください。詳しくは、「[カスタムキーボード](#)」を参照してください。

**構築したウェブサイトが、端から端まで画面が続くiPhone Xのディスプレイで適切に表示されることを確認する。** 詳しくは、[webkit.org](#)の「[Designing Websites for iPhone X](#)」を参照してください。

## リソース

PhotoshopとSketchで利用できるiPhone X UIデザインテンプレートは、「[Apple UI Design Resources](#)」でダウンロードしてください。

## さらに詳しく

関連するデザインガイダンスおよびデベロッパ向けのガイダンスについては、以下のビデオをご覧ください。

- Fall 2017 : 「[Designing for iPhone X](#)」
- Fall 2017 : 「[Building Apps for iPhone X](#)」
- WWDC 2017 : 「[Auto Layout Techniques in Interface Builder](#)」
- WWDC 2017 : 「[Designing Across Platforms](#)」
- WWDC 2017 : 「[Modern User Interaction on iOS](#)」
- WWDC 2017 : 「[Updating Your App for iOS 11](#)」