これらのアイコンカードのいずれかをクリックすると、アプリの最終バージョンで実際に色が変化し、操作され選択されたことを示すことがわかります。しかし現在、私たちのカードはクリックしても実際には何もしません。そして、私たちのカードまたはそれが基づいているものがiconContentまたはreusableCardにあることを見ることができます。それらのどれも実際にはユーザー入力を取得できません。 onPressedやonTapなどはありません。

このレッスンでこれを修正し、アイコンカードをインタラクティブにしましょう。これを行う1つの方法は、reusableCardをフラットボタンでラップすることです。そのため、OPTIONまたはALTを押しながらEnterキーを押します。

その後、新しいウィジェットでラップすると、ウィジェットはフラットボタンになります。

そしてもちろん、これにより、ユーザーがこれらの再利用可能なカードのいずれかを実際に押す瞬間にアクセスできる重要なonPressedにアクセスでき、色をそのように変更できるようになります。

今、フラットボタンの問題は、今すぐ保存を押すと、実際にカードのスタイルに影響することを確認できることです。

子のマージンやその他のさまざまなスタイルを変更しようとしています。

だから、それは非常に熱心なウィジェットです。

そして、フラットボタンのユースケースは、素材のフラットボタンのスタイルを実装することであり、さまざまな色のさまざまなテーマとその外観が含まれます。

しかし、私たちの場合、実際に必要なのは、すでにかなりスタイルの整った再利用可能なカードのタッチを検出する能力だけです。

そのため、フラットボタンの代わりに、ジェスチャ検出器と呼ばれるものを使用できます。ジェスチャ検出器は、ブリキに書かれていることをほとんど実行しますか？

ジェスチャを検出し、検出できるジェスチャはonTapのようなものです。これはonPressedと同じですが、さらに多くを検出することもできます。

そのため、たとえば、長押しやドラッグ、強制押しなどを検出できます。

このウィジェットが関係しているのは、ジェスチャーの検出だけです。

子ウィジェットにスタイルや外観、アニメーションを強制しようとはしません。

したがって、これは、ユーザーが私たちのウィジェットと対話するときを検出する非常に純粋な方法です。

そのため、再利用可能なカードをジェスチャ検出器の子として持つことで、検出可能なジェスチャを利用できるようになりました。

したがって、たとえば、onTapは画面上の古典的なシングルタップです。

したがって、フラットボタンのonPressedと同じです。または、垂直方向のドラッグ、水平方向のドラッグ、ダブルタップ、強制プレス、その他すべてを検出できます。

ただし、今必要なのはonTapジェスチャー検出器だけです。

そこで、古典的な匿名コールバックを追加し、ここに男性カードが押されたことを印刷します。保存ボタンを押して実行タブが表示されていることを確認してアプリにアクセスすると、その男性ボタンをクリックすると、押すたびに「男性カードが押されました」というメッセージが表示されます。

したがって、フラットボタンと同じように機能し、実装は非常に簡単です。ジェスチャ検出器ができたので、ユーザーが実際にカードをクリックしたかどうかを、男性であろうと女性であろうと検出できます。

そして、私たちが実現したいのは、男性カードをクリックしたときに、選択されたことを示すために色を変えたいということです。

そして、これを行うには、もちろん、セット状態を使用する必要があります。

そして、再利用可能なカードの色プロパティを更新します。

これが、ここにステートフルウィジェットがあり、カードの状態を実際に変更できる理由です。これを行う方法は、カードが少し暗い背景色で開始されることです。

タップすると、カードの残りの部分と同じ色に変わります。

アクティブな色。

最初に定数でここに、activeCardColorを作成し、inactiveCardColorの新しい定数を作成します。また、inactiveCardColorもカスタムカラーになり、16進値は0xFF111328になります。

そして、ここでは、溝の中でactiveCardColorから少しだけ暗くなっていることがわかりますが、メインの背景色とまったく同じではありません。

少し軽くなっています。

そして、私が先に進んで、例えば男性カードの色を非アクティブなカードの色に変更し、保存をクリックすると、アクティブな色よりも少し暗いが、背景色とまったく同じではないことがわかりますただ溶け込むだけではありません

そのため、ここにはカードがありますが、明らかに選択もアクティブ化もされていませんが、このカードは他のカードと同じように見えます。

そのため、アクティブ化されたカードまたは選択されたカードのように見えます。

これが私たちが目指している外観です。

そして、両方のカードが非アクティブなカードの色を持ち始めるはずですが、それらの1つがタップされると、その色をアクティブなカードの色に変更する必要があります。

そのため、プロパティを変更および変更できるように変更可能な状態ウィジェットに移動します。したがって、このクラスのプロパティはfinalである必要はありません。

それらは、変化する可能性のある変数にすることができます。

したがって、最初に作成するのは色タイプです。

そして、maleCardColorと呼ばれ、inactiveCardColorと等しくなるように設定されます。

したがって、これは、男性カードの色と女性カードの色の両方の開始値になります。

そして、この変数、femaleCardColorおよびmaleCardColorを再利用可能なカード内で使用します。

このようにして、選択されたカードに応じて、これらのカードの色を選択的に変更できます。

ここで、カードをonTapに変更するために、updateColorというメソッドを作成します。

これは、どの性別がクリックされたかを何らかの形で教えてくれる入力を受け取ります。

ですから、性別と呼ばれ、今は単に整数になります。

そして、たとえば、1つを渡すと男性に等しく、2つを渡すと女性に等しいと言うことができます。このメソッド内では、IFステートメントを使用して、性別が1に等しいかどうかを確認します。

したがって、値1を渡すと、これは男性カードが押されたことを意味します。

次に、この場合、maleCardColorが現在非アクティブなCardColorと等しいかどうかを確認します。

その場合、それをactiveCardColorに変更します。

そこで、カードの色を切り替えます。

ただし、既にactiveCardColorにある場合は、inactiveCardColorになる反対の値に変更します。

したがって、onTapに進み、「男性カードが押されました」を印刷する代わりに、セット状態を使用して、色を決定するために使用されるmaleCardColorプロパティの状態を更新します。

次に、今作成したメソッドupdateColorを呼び出します。

そして、数字の1は男性の性別を表すと言った。

だから今保存を押すと、私のカードが両方とも非アクティブな色であることがわかります。

したがって、これらはこれとまったく同じではありませんが、クリックするとすぐにアクティブな色に変わります。

それは、このメソッドに1を渡し、maleCardColorが非アクティブになり始めたときに、タッチを検出するとすぐにアクティブな色に更新するためです。

ただし、現在はアクティブな色なので、もう一度クリックすると、非アクティブな色に戻ります。

このようにして、カードの背景色を切り替えて、何かが選択されているかどうかを表すことができます。

それでは、女性カードについても同じことをしましょう。性別が1と等しくない場合、またはアプリに複数の性別が必要な場合は、別のELSEを追加するだけで、性別が2に等しいかどうかを確認するIFステートメントを追加するだけです。

したがって、2は押された女性カードに等しいと言います。

そしてこの場合、私たちはほぼ逆のことをします。

したがって、femaleCardColorが非アクティブであるかどうかを確認します。

その場合は、アクティブな色に変更しますが、そうでない場合は非アクティブに切り替えます。

そのため、今度は女性カードの中に、再利用可能なカードを新しいジェスチャー検出器の中にラップします。

また、これには、タップが検出されるとトリガーされる匿名関数であるonTapもあります。

それが起こったら、set stateを呼び出して色を更新し、性別を2番として渡し、女性カードが押されたと伝えます。保存して、アプリをチェックしてみましょう。

そのため、男性カードをクリックすると、男性カードが点灯し、もう一度クリックして非アクティブ化できます。

そして、女性のカードをクリックすると、女性のカードが有効になります。

これで、BMI計算機で性別を選択できるようになりました。

現時点では、女性カードが選択されているときに男性カードをクリックしても、女性カードは選択されたままになりますか？

しかし、私が間違いを犯し、実際にこれを選択するつもりだったとしましょう。クリックすると、おそらく両方を同時に選択するべきではないため、これをオフにする必要があります。

では、どうすればこれを行うことができますか？

既に非常に冗長な方法を追加しますが、それは非常に簡単です。

心配する必要はありません。少し後で目に見えるようになります。

しかし、最初に機能をマッピングして、何が起こっているのかを理解しています。

したがって、updateColorが呼び出され、押された男性カードであるため、性別が1の場合、maleCardColorをオンにするだけでなく、femaleCardColorもオフにします。したがって、これをinactiveCardColorに変更します。また、女性のカードが押された場合もその逆です。次に、maleCardColorをオフにします。

そのため、これを非アクティブに変更します。

次に、保存をクリックしてみましょう。男性のカードをクリックすると、女性のカードがオフになります。

女性のカードをクリックすると、男性は切れます。

そして、この振る舞いはもっと意味があります。これからのレッスンでは、このロジックのすべてを、もっと短く、簡潔で、目に優しいものに単純化します。

現時点では、それは非常に冗長で、実際に何が起こっているのか非常に混乱しているためです。

しかし、これを達成するには、Dartについてもう少し学ぶ必要があります。

そのため、次のレッスンでは、Dart列挙型について説明します。それ以外にも、

会いましょう。