30TH EMBARCADERO DEVELOPER CAMP

第30回 エンバカデロ・デベロッパーキャンプ・ツアー

【F5】Delphi/C++テクニカルセッション

RAD Studioで始めるモバイル開発 ～ コンポーネントで簡単入門！ 勘所も押さえよう

株式会社シリアルゲームズ
取締役　細川　淳

embarcadero

アジェンダ

- はじめに
- はじめての FireMonkey で作るアプリケーション
- iOS アプリの作法
- Android アプリの作法
- 共通の作法
- まとめ

はじめに

Delphi / C++Builder

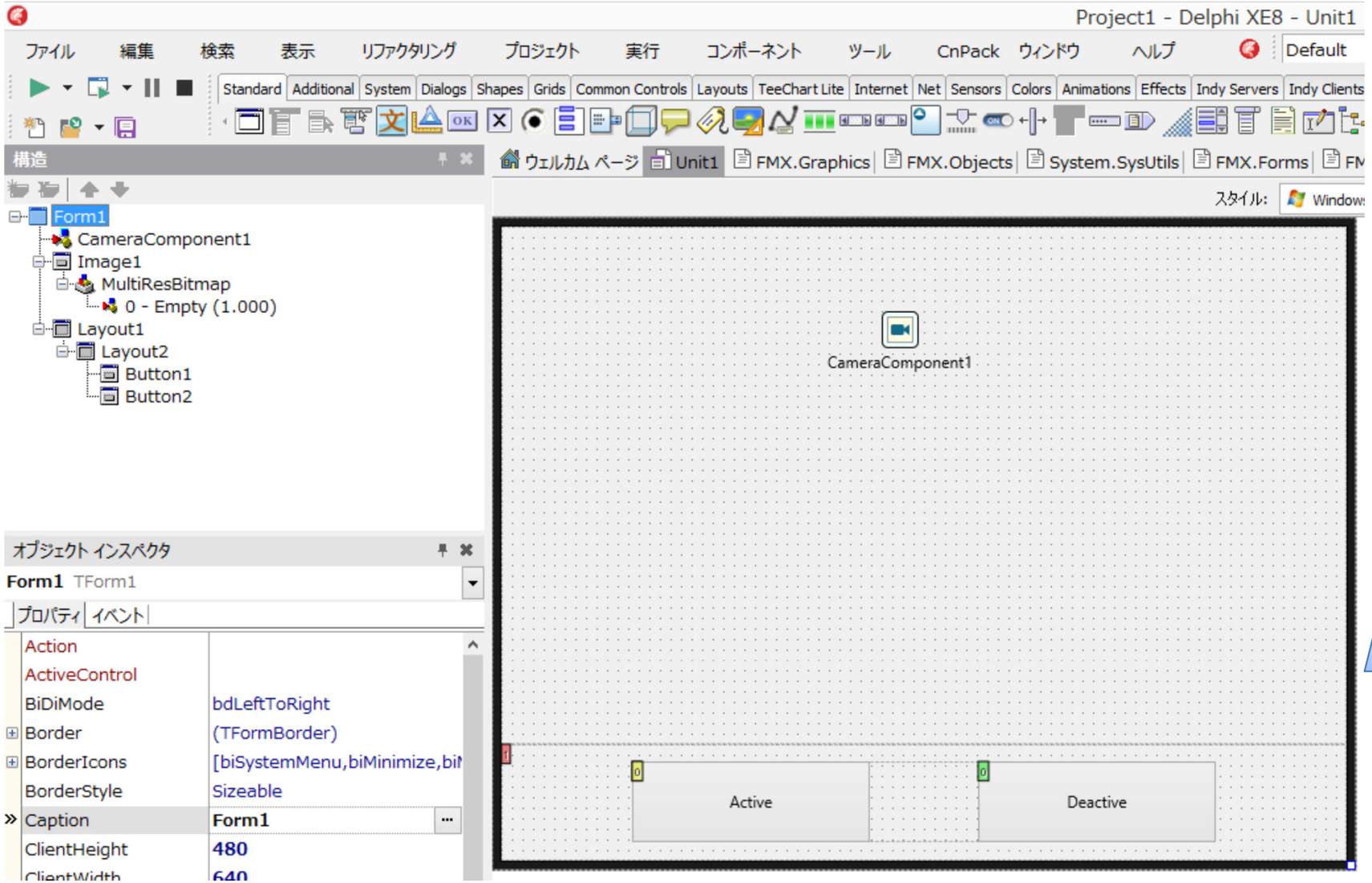
- このセッションの対象
  - Delphi or C++Builder User
    - ただし説明は全て Delphi で行います
  - 初級者～中級者
    - これから FireMonkey でアプリを作る方々のための作法を説明します。

はじめての FireMonkey アプリケーション

カメラアプリ

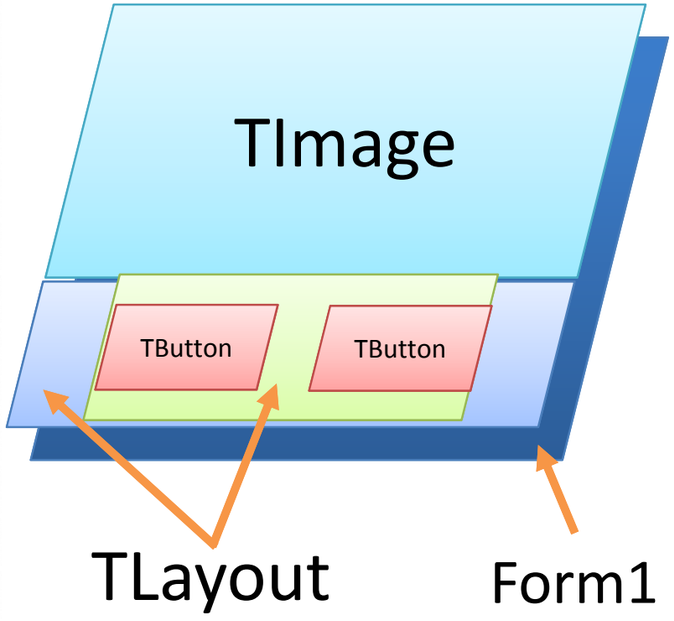
- Delphi でカメラアプリとか飽きたよお！とか言わない！
- TTakePhotoAction を使わずに
- TCameraComponent で
- TImage にカメラの映像を出します

Form デザイン



使用したコンポーネント

- TCameraComponent
- TImage
- TLayout
- TButton



embarcadero

イベントハンドラ

```
// Button1 を押してカメラを Active に
procedure TForm1.Button1Click(Sender: TObject);
begin
  CameraComponent1.Active := True;
end;

// Button2 を押してカメラを Deactive に
procedure TForm1.Button2Click(Sender: TObject);
begin
  CameraComponent1.Active := False;
end;
```

イベントハンドラ

```
// カメラの映像が取得可能になったとき呼ばれます
procedure TForm1.CameraComponent1SampleBufferReady(
  Sender: TObject;
  const ATime: TMediaTime);
begin
  // TImage のビットマップサイズを TImage いっぱいまで広げます
  // BitmapScale は画素密度です。たとえば xxdpi(FullHD) では 3 になります
  // FMX は全て仮想サイズですが、画像(Bitmap)は仮想では無いので補正が必要になります
  Image1.Bitmap.SetSize(
    Trunc(Image1.Width * Image1.Bitmap.BitmapScale),
    Trunc(Image1.Height * Image1.Bitmap.BitmapScale));

  // カメラの映像を TImage.Bitmap に描画します
  CameraComponent1.SampleBufferToBitmap(Image1.Bitmap, False);
end;
```

FireMonkey の作法

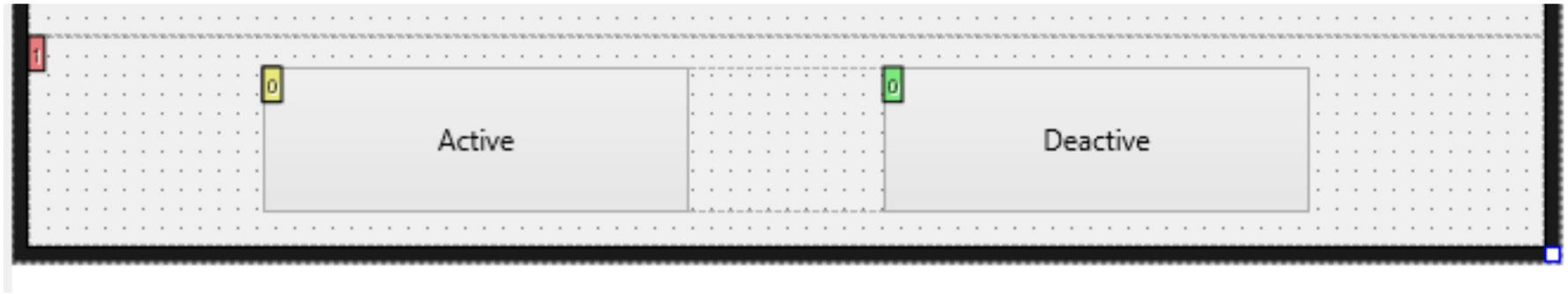
- VCL と同じように考えてはダメ！
  - デバイスによって画面解像度も比率も異なる
- レイアウト！
- スタイル！

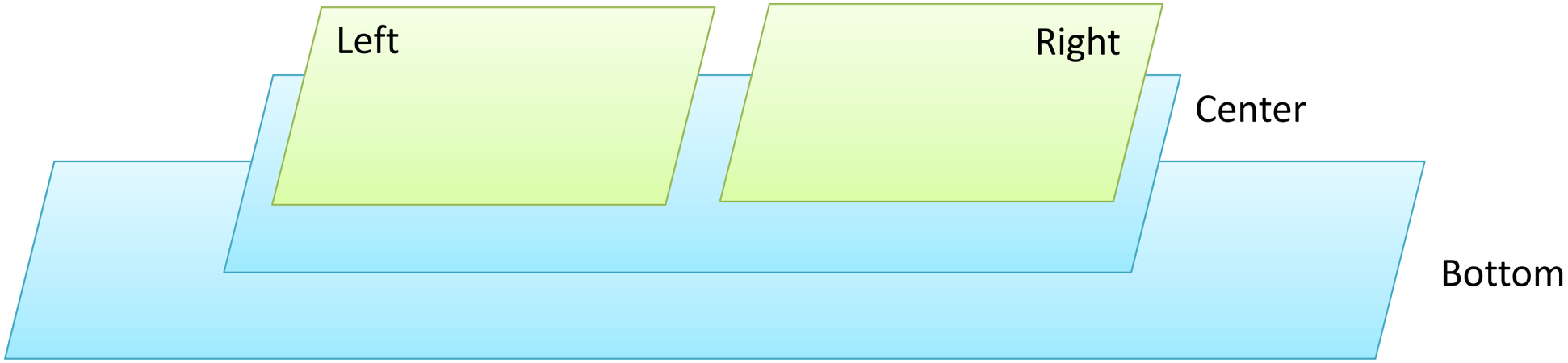
レイアウト

- デバイスごとに違いがある解像度や比率から解放してくれる！
- Align と Anchor を使いましょう！
- TLayout は何も描画しない純粋なコンテナ

例

- たとえば複数のコントロールをセンターに配置する





スタイル

- スタイル！スタイル！スタイル！
- コントロールの見た目要素はスタイルで！
  - スタイルの重要性がいまだに伝わってない感！

スタイル

FW	たとえば
VCL	HTML HTML では Font color プロパティ, Body bgcolor など個々のプロパティで
FMX	HTML5 + CSS 見た目の要素は HTML ではなく CSS で CSS ではリッチな見た目も定義できる！

- FMX では、エフェクト、α（透明度）、アニメーションなど見た目は全部スタイルに！
  - テキスト関係のプロパティ（TextSettings）だけ外出しされている
    - これによって混乱が助長されている点も…

スタイルの罠

- これだけスタイル！と言ってるのに
- **スタイルエディタのできが悪い！**
  - デベロッパーキャンプで、これ以上は…

スタイルの罠

- ビットマップを利用したスタイルを使う方法が解説されていない！
- しかも、コントロールが公開されていない！！
  - TCustomStyleObject 派生コントロールなど
- DEKO 氏がそれらをレジストするパッケージを「ハラヘッタウェア」で公開されています
  - http://ht-deko.minim.ne.jp/delphiforum/?vasthtmlaction=viewtopic&t=1501
    - TButtonStyleObject など、追加されていないものもあります

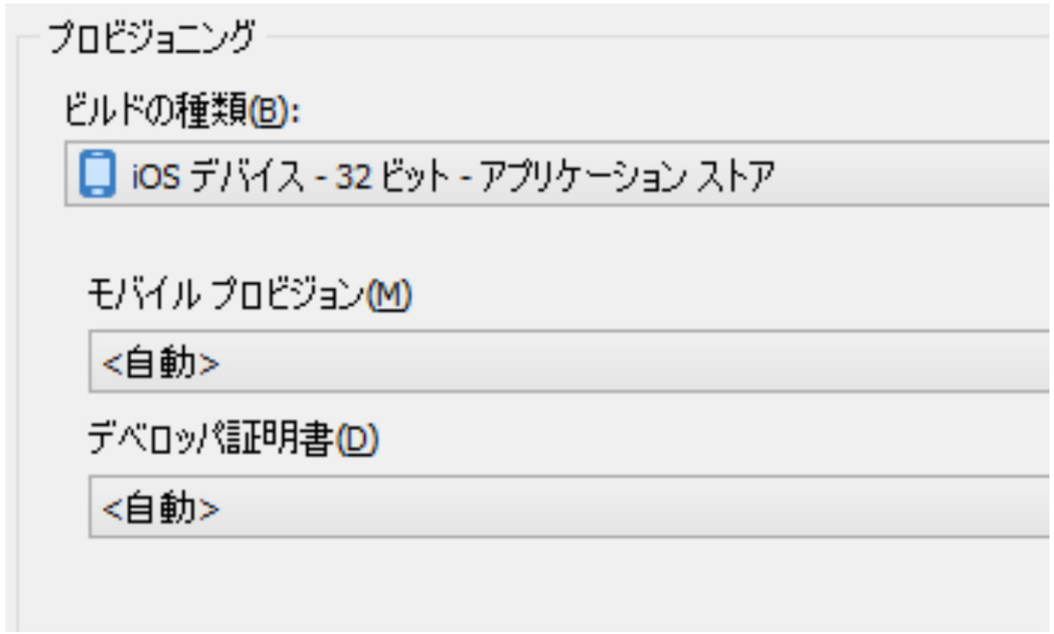
iOS の作法

iOS の作法

- モバイルプロビジョニングってナニ！！！
- デベロッパ証明書ってナニ！！！
  - Entitlement plist ってナニ！！！
  - CFBundleID ってナニ！！！

iOS の作法

…っというのを話そうと思っていたのですが
XE8 ではあまり考えなくてもよくなりました！！



XE7 までの人は…頑張って！！
特に注意してほしいのは
「モバイル・プロビジョニング・ファイル」が良くわからなくて、
たくさんダウンロード＆インストールしてしまった場合。
同名のプロビジョニングファイルがあると
どのプロビジョニングファイルを使っていいか解らないため
署名に失敗します。

iOS の作法

- 自分のフォルダ以外には触れない！
  - Windows のように自由に触ることは不可能！！
- フォルダには意味がある！

iOS の作法

フォルダ	意味	バックアップ	自動削除
Documents	アプリのデータ保存ディレクトリ	○	×
Documents/Preferences	アプリの設定	○	×
Library/Cache	再取得・再生成できるデータを置く所	×	○
tmp	システムに自動削除されたくない 一時データ（削除は自分で） 容量に注意！	×	×

これを守らないと、リジェクトされます！

※どうしても！という場合

NSURLIsExcludedFromBackupKey を属性として付与すると、バックアップされなくなります

iOS の作法

- /StartUp/Documents
  - 配布マネージャを使ってアプリにデータを入れておく場所
  - ここに入れておくと、自動的に Documents ディレクトリに配置されます
  - iCloud にバックアップされます
  - ここに大量のデータを入れておくと初回起動時に大変重くなるので注意！！

Android の作法

Android の作法

- 開発について
  - まずは絶対にログを見るようにしましょう！！
  - ADB や DDMS で見れます
  - SERIALGAMES 製 Android LogViewer でも見れます
    - ※現在公開停止中

Android の作法

- ログには何が出ている？
  - もちろん、アプリ個々のログを出せます
  - システムのメッセージ
    - エラーが起きたときの**スタックトレース**！
    - Java の機能を使っているところでエラーが起きると場所まですぐにわかります！

Android の作法

- FMX.Types の Log は偽物
  - Log.d メソッドでなぜか Log.i が出力される仕様

- 正しい Log Level で出力されるライブラリを用意しました
  - FMX.Log
    - https://github.com/freeonterminate/delphi/tree/master/FMXLog

Android の作法

- Back キー
  - Delphi 製アプリケーションは Back キーを押されると終わります。
  - 終わります！！
    - 大切なことなので2回言いました。
    - →終わったように見せない工夫は、共通の作法に記載します

Android の作法

- Home キー
  - Pause 状態になります。
  - 最初に作ったアプリ、カメラが Active な状態で Home キーが押されると…？
    - **死にます！！！！！！！**
    - →解法は、共通の作法に記載します

Android の作法

- OS のバージョンに注意
  - Android 4.4 から
    - SD カードに自由にアクセスできなくなりました！
    - アプリ専用ディレクトリ以下のみにアクセスできます
  - Android も iOS も「非推奨 API」あり
    - とくに iOS の場合はリジェクト対象に！

Android の作法

- TMediaPlayer に注意！
  - TMediaPlayer.Stop メソッドを呼ぶと FileName を再設定（再オープン）するまで再生できなくなる罠！
  - Android の MediaPlayer#stop メソッドの仕様
    - play で使用したメモリなどを解放するため
    - 再度鳴らしたい場合は pause を使うようにと API ドキュメントに
      - ただし、TMediaPlayer は **pause メソッドを公開していない！！**
        - » →解法は、共通の作法に記載します

Android の作法

- ブラックアウト！
  - XE7, XE8 では、あまり起こらなくなりました
  - XE6 位をまだ使っている場合に一度試して頂きたいのが次のコードです。

Android の作法

```
case iAppEvent of
  TApplicationEvent.WillBecomeForeground:
  begin
    {$IFDEF ANDROID}
    // onWindowFocusChanged を呼ぶと TWindowManager.Render が呼ばれて再描画される
    TThread.Queue(
      TThread.Current,
      procedure
      begin
        PANativeActivity(System.DelphiActivity)^.
          callbacks^.
          onWindowFocusChanged(System.DelphiActivity, 1);
      end
    );
    {$ENDIF}
  end;
  :
  ;
end;
```

Android の作法

- デバッグ手法
  - 「設定」→「開発者向けオプション」
  - アクティビティを保持しない
    - これを設定すると裏に回った時に、アプリが終了する
    - 適切に非アクティブ処理がされているか？
      - されていない場合はログにエラーが出ているはず
  - バックグラウンドプロセスの上限
    - バックグラウンドプロセスが走らない設定も可能

Android の作法

- Delphi だけで実現できないことは Jar ファイルで
  - XE7 からプロジェクトマネージャに Jar ファイルを追加可能に
  - Java2OP を使えばアクセス用クラスも自動的に作成されます

共通の作法

共通の作法

- TPath の使用
  - TPath を使用すると OS 間のディレクトリ構造の違いを吸収してくれます（意味からディレクトリを返してくれます）
  - TPath.GetDocumentsPath で Documents
  - TPath.GetCachePath で Library/Cache
  - など

Docwiki サポートされているターゲット プラットフォームに適した標準の RTL パス関数
http://docwiki.embarcadero.com/RADStudio/XE8/ja/サポートされているターゲット_プラットフォームに適した標準の_RTL_パス関数

共通の作法

- アプリの状態変化を見るためには？
  - TForm.OnActivate / OnDeactivate はアプリ内のイベント
    - アプリの自身のイベントではない
      - Form1, Form2 があったとき、Form2.Show とかで来るイベント

- **IFMXApplicationEventService**
  - このインターフェースを使うと状態変化が通知されます。

IFMXApplicationEventService

```
constructor TLifecycleManager.Create;
var
  AppEventService: IFMXApplicationEventService;
begin
  if
    TPlatformServices.Current.SupportsPlatformService(IFMXApplicationEventService)
  then
    try
      AppEventService :=
        IFMXApplicationEventService(
          TPlatformServices.Current.GetPlatformService(IFMXApplicationEventService)
        );
    except
      AppEventService := nil;
    end;

  if (AppEventService <> nil) then
    AppEventService.SetApplicationEventHandler(AppEvent);
end;
```

IFMXApplicationEventService

```
function TLifecycleManager.AppEvent(
  iAppEvent: TApplicationEvent;
  iContext: TObject): Boolean;
begin
  Result := False;

  case iAppEvent of
    TApplicationEvent.FinishedLaunching:
    begin
      // 起動完了
    end;

    TApplicationEvent.BecameActive:
    begin
      // フォーカスを取得
    end;
```

```
    TApplicationEvent.WillBecomeInactive:
    begin
      // フォーカスを失う
    end;

    TApplicationEvent.EnteredBackground:
    begin
      // 裏に回った
    end;

    TApplicationEvent.WillBecomeForeground:
    begin
      // 表に回った
    end;

    TApplicationEvent.WillTerminate:
    begin
      // アプリが終了する
    end;
```

IFMXApplicationEventService

```
    TApplicationEvent.LowMemory:
    begin
      // メモリが少ない！
    end;

    TApplicationEvent.TimeChange:
    begin
      // iOS のみ
      // 時刻に変化あり
      // 日が変わったり夏時間などで
    end;

    TApplicationEvent.OpenURL:
    begin
      // iOS のみ
      // URL が開かれる
    end;
  end;
end;
```

共通の作法

- BecameActive
  - ここでアクティブ化の処理を
- WillBecomeInactive
  - ここで非アクティブ化の処理を
    - 先のカメラの例でいえば、ここで Active を False にします

共通の作法

- OnSaveState イベントの活用(XE7以降)
  - Android では Back キーでアプリが終わってしまいますが、これを使ってあたかも終わってないように見せることもできます
    - 状態を保存して、それを復帰させる
  - 詳しくは DocWiki http://docwiki.embarcadero.com/RADStudio/XE8/ja/FireMonkey_の状態保存

共通の作法

- TMediaPlayer.pause が公開されてない！
  - → よしリフレクション（拡張 RTTI ）を使おう！

- FireMonkey では、FMX 自身がリフレクションを使いまくっているので、臆することはありません…？
  - アニメーションとか
  - 将来のバージョンアップで動作しなくなる可能性があります

QualityPortal には報告済み https://quality.embarcadero.com/browse/RSP-10340

TMediaPlayerHelper

あとで削除も簡単にできる Class Helper を使って Pause を実装します

```
unit uMediaPlayerHelper;

interface

uses
  FMX.Media;

type
  TMediaPlayerHelper = class helper for TMediaPlayer
  private
    function GetPlayer<T: IInterface>(const iMedia: TMedia): T;
  public
    procedure Pause;
  end;
```

```
implementation

uses
  System.SysUtils
  , System.Rtti
  , FMX.Types
  {$IFDEF ANDROID}
  , Androidapi.JNI.Media
  , FMX.Media.Android
  {$ENDIF}
  {$IFDEF IOS}
  , iOSapi.AVFoundation
  {$ENDIF}
  ;
```

TMediaPlayerHelper

```
{ TMediaPlayerHelper }

function TMediaPlayerHelper.GetPlayer<T>(const iMedia: TMedia): T;
var
  RttiType: TRttiType;
  RttiField: TRttiField;
begin
  Result := nil;

  if (iMedia <> nil) then
  begin
    RttiType := SharedContext.GetType(iMedia.ClassType);
    if (RttiType <> nil) then
    begin
      RttiField := RttiType.GetField('FPlayer');
      if (RttiField <> nil) then
        Result := T(RttiField.GetValue(iMedia).AsInterface);
    end;
  end;
end;
```

TMediaPlayerHelper

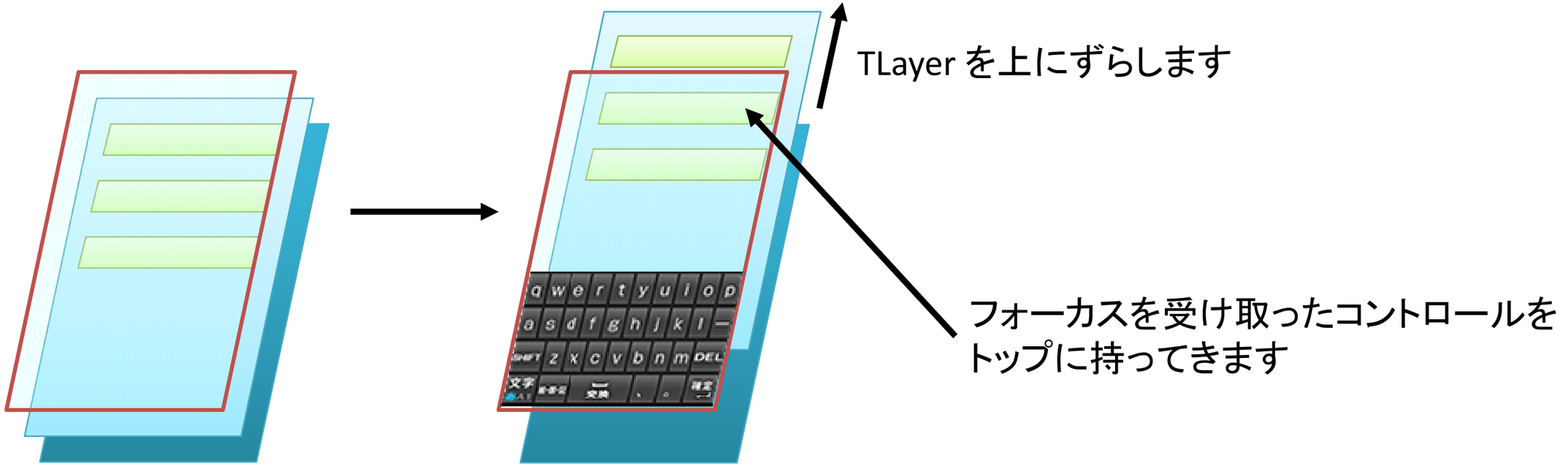
```
{$IFDEF ANDROID}
procedure TMediaPlayerHelper.Pause;
var
  Player: JMediaPlayer;
begin
  Player := GetPlayer<JMediaPlayer>(Media);
  if (PLayer <> nil) then
    Player.Pause;
end;
{$ENDIF}

{$IFDEF IOS}
procedure TMediaPlayerHelper.Pause;
var
  Player: AVPlayer;
begin
  Player := GetPlayer<AVPlayer>(Media);
  if (PLayer <> nil) then
    Player.Pause;
end;
{$ENDIF}

end.
```

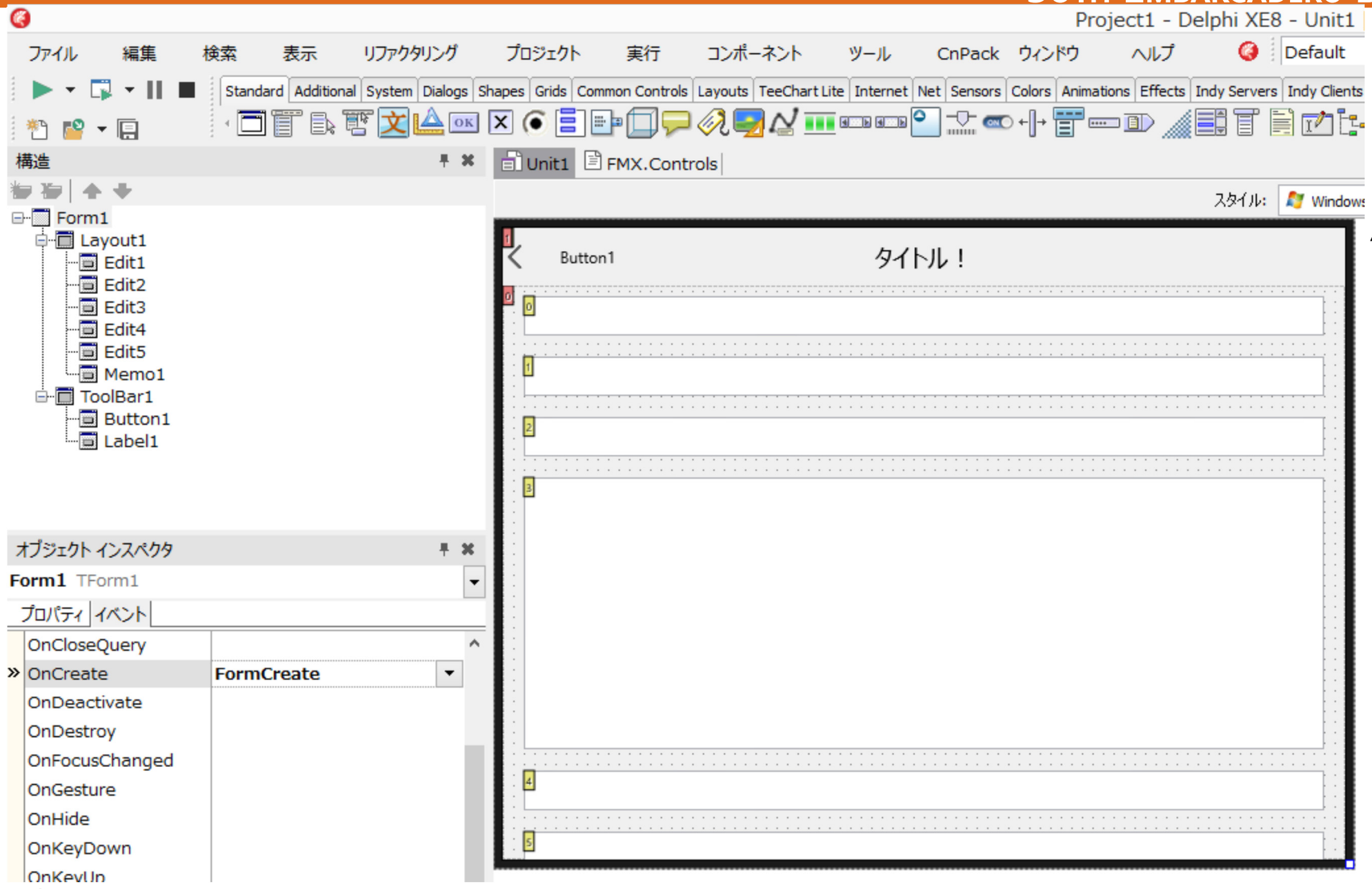
adjustpan / adjustresize

- TEdit などがフォーカスを受け取ったときソフトキーボードで隠れてしまうのを防ぎたい！
  - 普通の Android アプリでは adjustpan / adjustresize の2つで制御できます
- AdjustPan
  - 画面全体を上に移動
- AdjustResize
  - 一部のコントロールをリサイズ

AdjustPan

- TForm の一番下に TLayer を置きます
- 入力系コンポーネントがフォーカスを受け取ったら TLayer の Position.Y を変更します





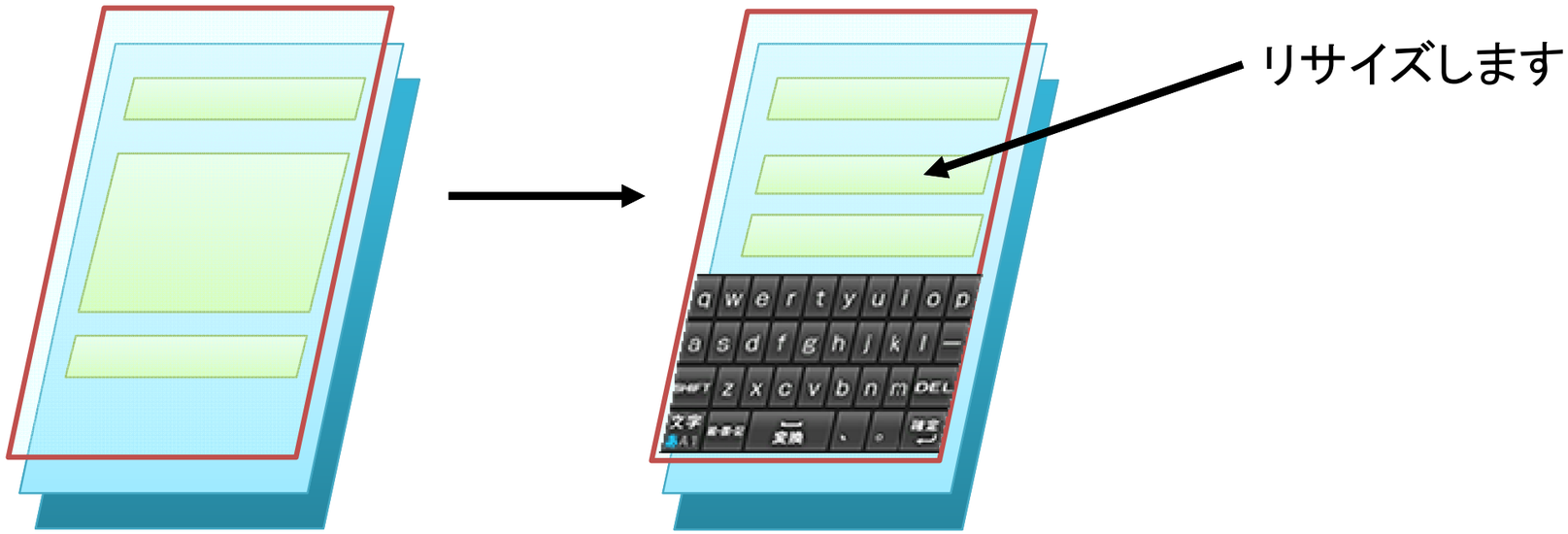
全体を上に
ずらします

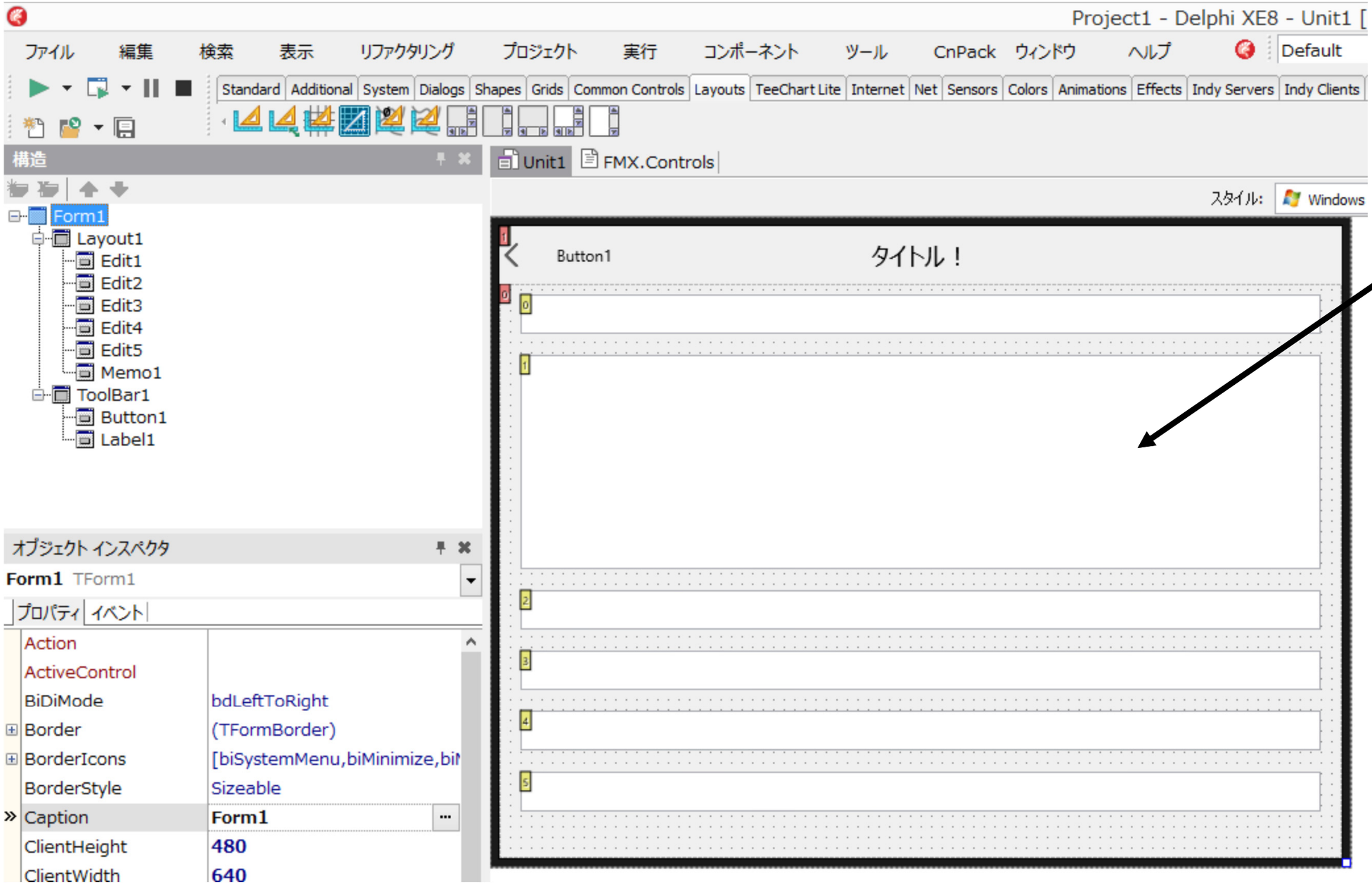
サンプルコード

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  FOrgY := Layout1.Position.Y;
end;

procedure TForm1. ForcusEnter(Sender: TObject);
var
  Pos: TPointF;
begin
  Layout1.Align := TAlignLayout.None;
  Layout1.Position.Y := FOrgY - TControl(Sender).Position.Y;
end;

procedure TForm1.FocusExit(Sender: TObject);
begin
  Layout1.Align := TAlignLayout.Client;
end;
```

AdjustResize

- 入力系コンポーネントがフォーカスを受け取ったら大きなコントロールをリサイズします





ここの Height を狭めます

サンプルコード

```
procedure TForm1.FormCreate(Sender: TObject);
begin
  FOrgMemoHeight := Memo1.Height;
end;

procedure TForm1.ForcusEnter(Sender: TObject);
begin
  if (TControl(Sender).Position.Y > Memo1.Position.Y) then
    Memo1.Height := 20 // 適当な値を入れてますが、本当はちゃんと計算してください
  else
    Memo1.Height := FOrgMemoHeight;
end;

procedure TForm1.FocusExit(Sender: TObject);
begin
  Memo1.Height := FOrgMemoHeight;
end;
```

まとめ

まとめ

- FireMonkey はスタイルとレイアウトが重要
- iOS は審査があるので注意
- Android はデバッグを十分に
  - ログ
- デバイス非依存になるように考慮して作成しよう
  - モバイルアプリはライフサイクルにも注意！

30TH EMBARCADERO DEVELOPER CAMP

第30回 エンバカデロ・デベロッパーキャンプ・ツアー

Thank you!

embarcadero®