

Starterユーザーのための アップグレードしやすいアプリ構築術

第33回 エンバカデロ・デベロッパーキャンプ

エンバカデロ・テクノロジーズ
セールスコンサルタント 相蘇 和貴
Twitter : @kazais_



embarcadero®
DEVELOPER CAMP

■ Starterユーザーのための アップグレードしやすいアプリ構築術

- Starterの機能と上位バージョンとの違い
- マルチデバイス開発を視野に入れたアプリ開発



アジェンダ

- Starter Editionの機能
 - 上位版との機能差
 - 上位版でやれること・Starter Editionでやれること
- アップグレードしやすいアプリ構築の必要性
 - マルチデバイス・クロスプラットフォーム開発
 - マルチデバイス向けUI設計の有意点
- 使えるコンポーネントとプログラミング
 - マルチデバイス向けに使えるコンポーネントとその使用方法を覚えよう！



Starter Edition (無料) と上位版の機能差

- Windows 32-bit アプリの開発がターゲット

機能	Starter	Pro	Enterprise
32-bit Windowsアプリケーションの開発	✓	✓	✓
64-bit Windowsアプリケーションの開発		✓	✓
Windows Store アプリの開発		✓?	✓
iOSアプリの開発		追加購入パック	✓
Androidアプリの開発		追加購入パック	✓
Mac OS Xアプリケーションの開発		✓	✓
ローカル/組み込みデータベースアプリ構築		✓	✓
C/Sデータベースアプリケーション構築		追加購入パック	✓



上位版でできること

- クロスプラットフォーム開発が可能
 - Windows 64bitアプリの開発
 - UWP Windows Storeアプリの開発
 - macOS Xアプリの開発
 - iOS, Androidアプリの開発
- データベースアクセスコンポーネントの利用
 - FireDACフレームワークが容易なデータベースアクセスを提供
- Starterで可能なこと
 - Windows 向け マルチデバイス 開発



マルチデバイス・クロスプラットフォーム開発とは

- **クロスプラットフォーム**：複数のOS対応をはかる
 - 複数のOS向けの開発を実施（同時に実施できればなおよい）
 - 複数のOS向けのUI,コンポーネント、開発環境、開発言語に対応
 - 対応OS: Windows, iOS, Android, Mac OS
 - 開発環境: Visual C#, Xcode, Android Studio, Eclipse, RAD Studio
 - 使用開発言語: C++, C#, Swift, Objective-C, Java, Delphi(Object Pascal)
- **マルチデバイス**：多様なデバイス対応をはかる
 - 多様なデバイス向けの開発を実施（同時に実施できればなおよい）
 - 多様なデバイス向けのUIに対応
 - タブレット、デスクトップ、2-in-1, スマートフォン、ウォッチ...



アップグレードしやすいアプリ構築の必要性

- マルチデバイス向けUI設計は、今やWindows向けに必要
 - タブレット型へ対応 → 小さな画面、縦長ディスプレイ
 - 2-in-1デバイスへ対応 → 大きめの画面 + 縦横 変換 + キーボード脱着
 - 異なるディスプレイサイズ → 4 K, High DPI対応, カスタムの拡大/縮小率
 - 異なるディスプレイ解像度



これらのマルチデバイス向け開発はStarter Editionで可能!



アップグレードしやすいアプリ構築の必要性

- いずれiOS, Android もやってみたくなる
 - 順調にスタディが進めばいずれ動かしてみたくなる
- Windows 32向けに開発したプロジェクトはそのまま利用可能？
 - ほぼそのまま、iOS, Android, Mac OSで動作させることが可能
- Delphi / C++Builder / RAD Studio のメリット
 - UI設計はマスタUIを中心とした ほぼ一つのUI設計でOK
 - 開発言語も一つでOK
 - もちろん開発環境も Delphi / C++Builder / RAD Studio の一つでOK!

➔ マルチデバイスを視野にいれておかないなんて、もったいない！



マルチデバイス向けのアプリ構築の考慮すべき点

- 一画面内に多くを配置しない
 - 画面サイズの小さいデバイスでも操作できるように情報量を少なく
- 画面の切り替えを利用
 - 画面遷移、ページ切り替えを利用し、進捗を意識してもらう
- タッチ操作
 - キーボード、マウスを前提としない入力と、ジェスチャー、スワイプ対応
- 画面の縦 / 横の切り替えを考慮
 - UI設計とコンポーネントの配置を検討するとともに、適切にプロパティを設定して追従



マルチデバイス向けに使えるコンポーネントを覚えよう！

■ 使用頻度の高いコンポーネント

用途	コンポーネント
画面切り替え	タブコントロール、マルチビュー、フォーム
タッチ操作	ジェスチャーマネージャー + アクションリスト
タッチで使いやすいコンポーネント	スピンドボックス、コンボボックス、スイッチ、トラックバー
一画面で見やすい配置	リストビュー、リストボックス
画面の大きさ、解像度の違い	グリッドパネルレイアウト、スケールレイアウト
配置・レイアウト	(プロパティ) アンカー、アライメント、パディング、マージン



コンポーネントを使いこなすプログラミングを覚えよう！

- 画面切り替えのポイント：画面遷移を意識してもらう
 - タブ、マルチビュー、フォーム

Common Controls

TTabControl

- 項目エディタでタブを追加し切り替え可能なタブ表示を行える
- プロパティのTabPositionで表示形式設定。Noneでタブなし、Bottomで画面下に表示
- SetActiveTabWithTransitionで TTabTransition.Slideを設定するとスライド切り替えになりGood

Common Controls

TMultiView

- プロパティのMasterButtonで表示ボタンを指定し別パネルを表示できる
- プロパティのModeで表示形式を指定：Drawerが画面遷移に使える
- 標準で画面端からのスワイプに対応していて、引き出し、格納が可能

マルチデバイスフォーム(W)

Form(W)

- プロジェクトマネージャで右クリック - [新規追加] - [マルチデバイスフォーム]で追加して新フォーム追加可能
- [ファイル] - [使用するユニット]で、メインフォームで呼び出し準備
- Form.show で表示、Form内でForm.Closeで閉じることができる

コンポーネントを使いこなすプログラミングを覚えよう！

■ タッチ操作

- ジェスチャーマネージャー + アクションリスト

☐ Gestures

TGestureManager

- 他コンポーネントの[Touch]-[GestureManager]プロパティに指定して使える
- 各ジェスチャーにどんなアクションを行うか指定 (ActionListと合わせて使う)
- ※OnTouchesイベントのほうが反応は良いけれど...

☐ Standard

TActionList

- アクションリストにアクション名を登録しどんな動作をするかプログラムできる
- タブコントロール向けのNextTabAction、PreviousTabActionが用意されていてTab動作に丁度良い
- ※スマホのカメラ機能の呼び出し等が利用可能



コンポーネントを使いこなすプログラミングを覚えよう！

- タッチで使いやすいコンポーネント
 - スピンボックス、コンボボックス、スイッチ、トラックバー

Additional

TSpinBox

- 左右の矢印をクリック・タップすることで数値を上下させることができる
- プロパティのMaxとMinで上限・下限の数値を設定
- 現在の設定値はプロパティのValueとして数値 (Single型)で取得できる

Standard

TComboBox

- クリック・タップすることで選択肢を表示して選択させることができる
- 選択肢はプロパティのItemsにStrings (文字列リスト)で登録するか項目エディタでListBoxItemとして追加しテキストを編集
- 現在の設定値はプロパティのItemIndexでインデックス (番号) で取得できる

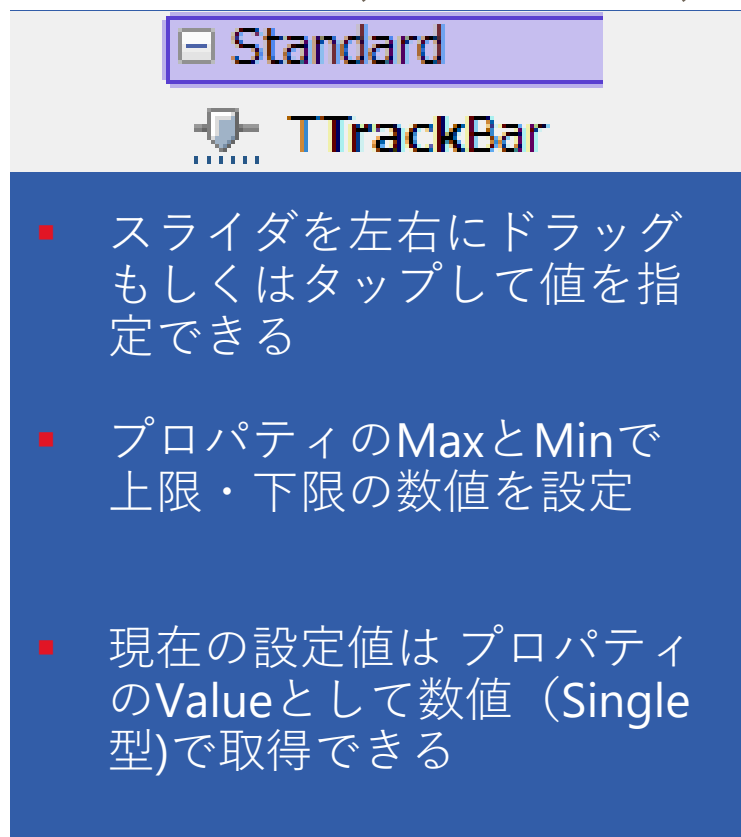
Standard

TSwitch

- クリック・タップすることでON/OFFを切り替え。
- プロパティのIsCheckedでTrueならON、FalseならOFFと判断できる
- 視覚的にわかりやすく、オススメ

コンポーネントを使いこなすプログラミングを覚えよう！

- タッチで使いやすいコンポーネント
 - スピンボックス、コンボボックス、スイッチ、トラックバー



- スライダーを左右にドラッグもしくはタップして値を指定できる
- プロパティのMaxとMinで上限・下限の数値を設定
- 現在の設定値は プロパティのValueとして数値 (Single型)で取得できる



コンポーネントを使いこなすプログラミングを覚えよう！

- 一画面で見やすい配置
 - リストビュー、リストボックス

Standard

TListBox

- 項目を整列して一覧表示できる。表示する項目数が決まっている設定画面などに向いている
- 項目エディタでヘッダ、フッタ、表示アイテムを並べ、各項目にコントロールなどを表示できる
- 表示内容が長くなってもスライド表示ができるようになっていて便利

Standard

TListView

- 項目を整列して一覧表示できる。表示する項目数が決まっていない一覧リストなどに向いている
- プロパティのAppearance設定で多様な表示内容を設定できる



コンポーネントを使いこなすプログラミングを覚えよう！

- 画面の大きさ、解像度の違い
 - グリッドパネルレイアウト、スケールレイアウト

Layouts

TGridPanelLayout

- 等しく分割してレイアウト可能な「枠」。枠内に設置したコントロールを等しい大きさに整列させることができる
- 均等に2分割、4分割...など行って、かつ画面のサイズやウィンドウの大きさ変更に対応したいときに便利

Layouts

TScaledLayout

- レイアウト内に設置されたコントロールをレイアウトのサイズの拡大縮小に合わせて、拡大縮小できる「枠」
- 画面のサイズやウィンドウの大きさに合わせてコントロールを拡大縮小したいときに便利



コンポーネントを使いこなすプログラミングを覚えよう！

- 配置・レイアウト
 - アンカー、アライメント

[-] Anchors	[akLeft,akTop]
akLeft	<input checked="" type="checkbox"/> True
akTop	<input checked="" type="checkbox"/> True
akRight	<input type="checkbox"/> False
akBottom	<input type="checkbox"/> False

- 親項目のそれぞれの辺と結び付けて配置を設定するもの
- 親項目の枠サイズ変更に従って位置が変化（結び付けられた一辺の移動に従）

>> Align	None
	Bottom
	Center
	Client
	Contents
	Fit

- コントロールを親項目内の特定の位置に整列・配置

[+] Margins	(TBounds)
[+] Padding	(TBounds)

- Margins
項目間（および親項目）の余白
- Padding
親から見た子項目に対する余白



本日のサンプルプログラムとサマリー

- Github

<https://github.com/kazais/SlidePuzzle>

※Delphi (Object Pascal)で書いてあります (C++は後日アップロード予定)

- 実施内容サマリー

- Community embarcadero (コミュニティ エンバカデロ)にWebリンク、サンプルコード情報等
<http://community.embarcadero.com/>

「エンバカデロ」で検索



本日のセミナー内容は
弊社ブログに掲載予定

[コミュニティ]
↓
[日本人ブログ]



まとめ

- Starter Editionの機能
 - 上位版との機能差
 - 上位版でやれることStarter Editionでやれること
- アップグレードしやすいアプリ構築の必要性
 - マルチデバイス・クロスプラットフォーム開発
 - マルチデバイス向けUI設計の有意点
- 使えるコンポーネントとプログラミング
 - マルチデバイス向けに使えるコンポーネントとその使用方法を覚えよう！



DELPHI / C++ BUILDER STARTER
チュートリアルシリーズ



THANKS!

www.embarcadero.com/jp

第33回 エンバカデロ・デベロッパーキャンプ