

物を回転させるのが WPF

じゃありません!!!

えムナウ（兎玉宏之）



えムナウのC#
プログラミングのページ



アジェンダ

- Windows Presentation Foundation アーキテクチャ
- アニメーション
- フロードキュメント
- リソースとスタイルとテンプレート
- イベントとキーとコマンド

WPF

- WPF(Windows Presentation Foundation)
 - .NET Framework 3.0で提供される新たなフレームワーク
 - UI、2Dおよび3Dオブジェクトの描画、ベクトルグラフィックス、ラスタグラフィックス、アニメーション、音声および動画の再生などといった表現手法を統一的に利用することができる。
 - Windows Vistaにプリインストールされており、Windows XP SP2およびWindows Server 2003でも利用できる。

WPF

- XBAP(XAML Browser Application)
 - Internet Explorerをホストとする、アプレットタイプのWPFアプリケーション。
 - WPFアプリケーションをほとんど修正しないで実行できる。
 - Internet Zoneで動作するためセキュリティの関係で動作しない機能がある。
 - .NET Framework 3.0が必要、Windows Vista・Windows XP SP2・Windows Server 2003 。

WPF

- WPF/E

- 軽量なランタイムを事前に配布することで様々なブラウザで動作するリッチコンテンツアプリケーションを開発できる。
- 機能は制限されるが複数OS上で各種ブラウザで実行できる。
- Windows2000・2003・XP・Vista、マック
- IE5.5以上・Mozilla1以上・Firefox1以上・Opera 7以上・Safari1以上。

WPF

	WinForm	WPF	XBAP	WPF/E	ASP.NET AJAX
ブラウザ	—	—	IE	IE・Mozilla・ Firefox・ Opera・ Safari	IE・Firefox・ Safari
プラット フォーム	Windows 2000・XP・ 2003・Vista	Windows XP・2003・ Vista	Windows XP・2003・ Vista	Windows 2000・XP・ 2003・Vista マック	任意
ランタイムイ ンストール	.Net Framework 2.0	.Net Framework 3.0	.Net Framework 3.0	ブラウザプ ラグイン	なし

Windows Presentation Foundation アーキテクチャ

ドキュメント サービス

XPS文書

パッケージサービス

ユーザインターフェースサービス

アプリケーション
サービス

コントロール

データバインディング

配置サービス

レイアウト

メディア サービス

描画

2D

オーディオ

効果

3D

テキスト

ビデオ

アニメーション

構成エンジン

基本サービス

XAML

アクセシビリティ

入力とイベント

プロパティ

基本サービスーXAML

デザイナー

デベロッパー

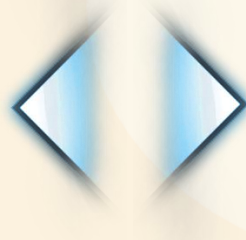
Expression

Visual Studio

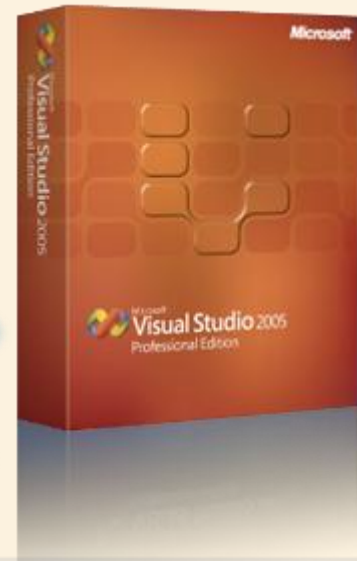
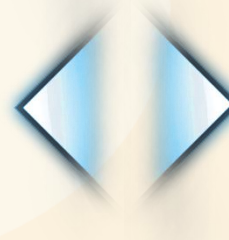


JPG
TIFF
PSD
PDF
GIF
PPT

C++
C#
VB



XAML



基本サービスプロパティ

Expression

XAMLを生成

XAML属性設定

Visual Studio

プログラムで動的生成

プロパティ設定



XAML

```
<Button Width="100"> OK
<Button.Background>
  LightBlue
</Button.Background>
</Button>
```

C#

```
Button b1 = new Button();
b1.Content = "OK";
b1.Background = new
SolidColorBrush(Colors.LightBlue);
b1.Width = 100;
```

VB.NET

```
Dim b1 As New Button
b1.Content = "OK"
b1.Background = New _
  SolidColorBrush(Colors.LightBlue)
b1.Width = 100
```



Expression Blend Visual Studio

連携

DEMO

ユーザーインターフェースサービスレイアウト

Canvas	現在の GDI や GDI+ の機能に似た、座標軸上の配置が行えます。追加されたプロパティを使用して項目の位置を制御することもできます。
DockPanel	上下左右にDockすることで正確な 配置を考慮せずに項目を配置できます。
StackPanel	コンテンツを左から右、または上から下へと配置するスタック モデルを提供します。
Grid	行/グリッドを配置するためのモデルを提供します。
WrapPanel	エクスプローラ ウィンドウの右側のウィンドウ枠をモデリングし、項目が現在の行の行末に達すると、新しい行に折り返します。
ScrollViewer	子のコンテンツにスクロール バーを設定します。利用可能なスペースからコンテンツがオーバーフローすると、スクロール バーが表示され、ユーザーがコンテンツ領域を移動できるようになります。
ViewBox	コンテンツのサイズが親パネルに収まるように制限されます。これにより、一種の自動ズーム効果が得られます。

ユーザーインターフェースサービスコントロール

Windows Forms のみ

PrintPreviewDialog	ColorDialog
DateTimePicker	FolderBrowseDialog
NumericUpDown	FontDialog
DomainUpDown	DataGrid
ErrorProvider	DataGridView
HelpProvider	
ImageList	
LinkLabel	
MaskedTextBox	
MonthCalendar	
NotifyIcon	
PrintDocument	
PropertyGrid	

ユーザーインターフェースサービス—コントロール

• WPF のみ (ToolBoxで選択できるもの)

図形

Line

Ellipse

Rectangle

ビューアー

ViewBox

ScrollViewer

FlowDocumentPageViewer

FlowDocumentScrollViewer

パネル

DockPanel

StackPanel

VirtualModeStackPanel

ToolBarOverflowPanel

UniformGrid

Popup

BulletDecorator

ユーザーインターフェースサービス—コントロール

• WPF のみ (ToolBoxで選択できるもの)

Textやボタン

TextBlock

PasswordBox

RepeatButton

ToggleButton

インク

InkCanvas

InkPresenter

3-D

ViewPort3D

小物

Expander

Thumb

Border

ResizeGrip

ContentPresenter

HeaderedContentControl

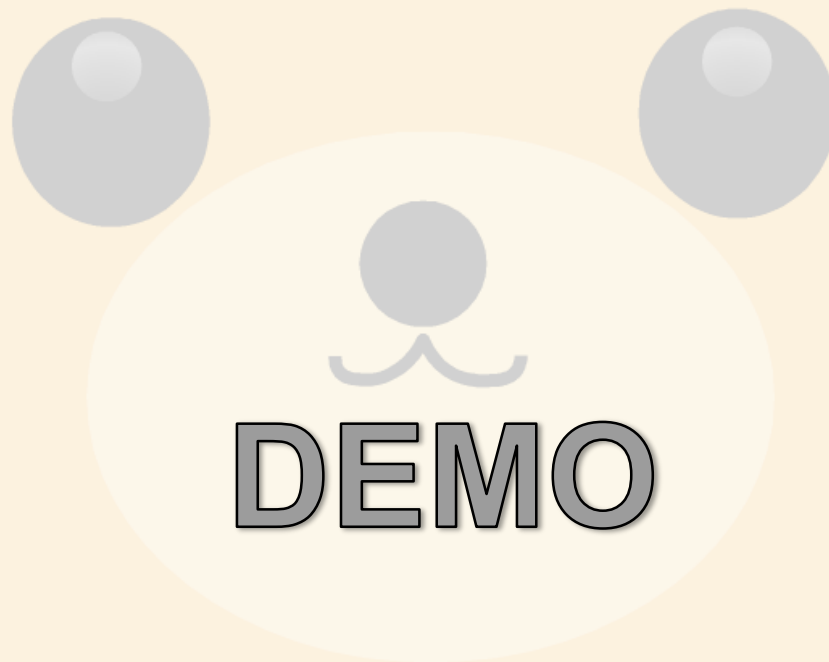
描画要素

- WPFではPenやBrushにはDisposeが不要になりました。
- System.Windows.Media 名前空間のものだけですがAnimatableクラスを継承している描画関係のクラス全般にわたってDisposeが不要です。

背景を透けさせる

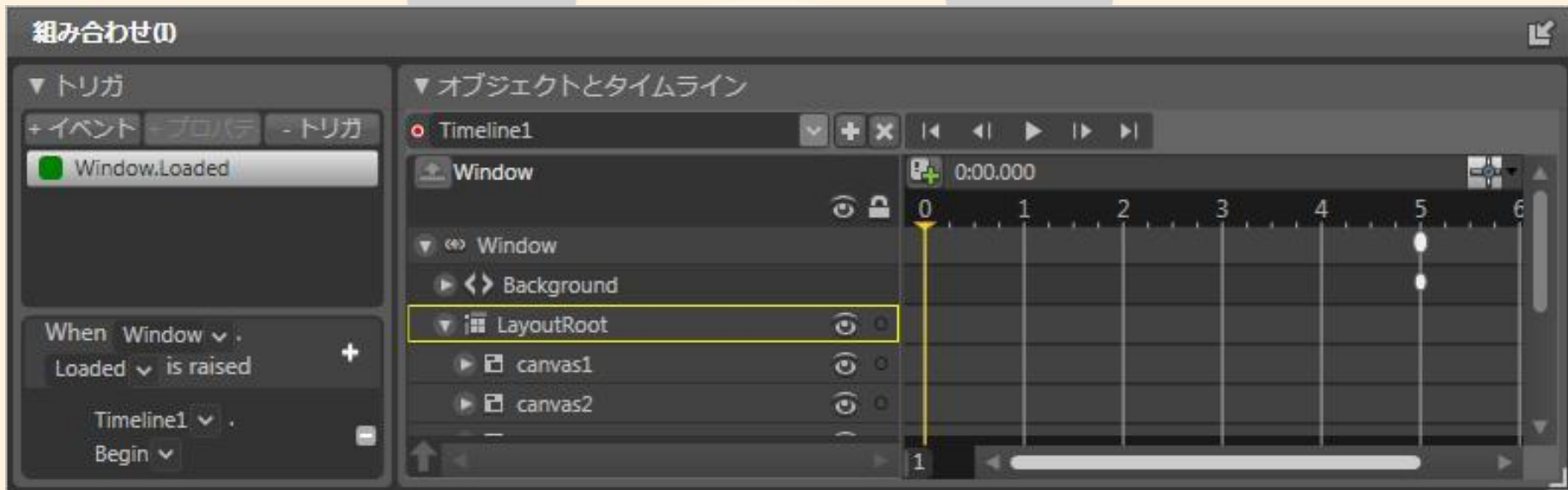
- dwmapi.dll の DwmExtendFrameIntoClientArea と DwmIsCompositionEnabled を使います。
- Window.OnSourceInitialized を、オーバーライドして Background と CompositionTarget.BackgroundColor を Transparent に指定します。
- DwmExtendFrameIntoClientArea で全 ClientArea に拡張します。

レイアウト



アニメーション

- Storyboard に Timeline を追加してアニメーションを作成し、イベントをトリガとして Timeline を起動します。
- 徐々に変化し指定時間には変化完了します。



アニメーション

- 例えば5秒後までに徐々に色が変わるようにするには以下のようなタグを使います。

ColorAnimationUsingKeyFrames
SplineColorKeyFrame

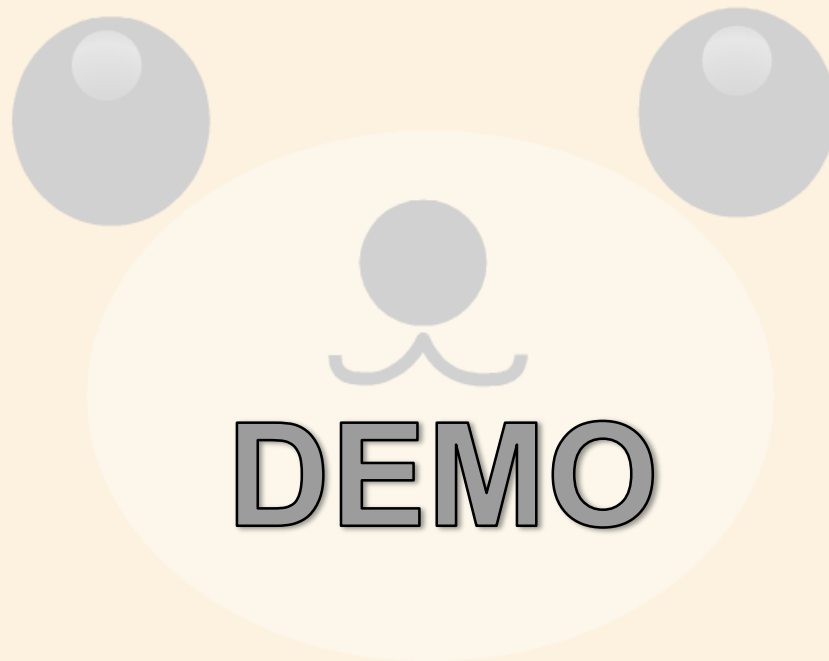
```
<Storyboard x:Key="Timeline1">  
  <ColorAnimationUsingKeyFrames BeginTime="00:00:00"  
Storyboard.TargetName="ellipse"  
Storyboard.TargetProperty="(Shape.Fill).(SolidColorBrush.Color)">  
  <SplineColorKeyFrame KeyTime="00:00:00" Value="#FF66132E"/>  
  <SplineColorKeyFrame KeyTime="00:00:05" Value="#FF6A6A6A"/>  
</ColorAnimationUsingKeyFrames>  
</Storyboard>
```

アニメーション

- アニメーション可能なクラスがある型

Boolean	Point3D
Byte	Point
Char	Quaternion
Color	Rect
Decimal	Rotation3D
Double	Single
Int16	Size
Int32	String
Int64	Thickness
Matrix	Vector3D
Object	Vector

アニメーション



DEMO

メディアサービス

	Windows フォーム	PDF	GDI+	Windows Media Player	Direct3D	WPF
グラフィカル インター フェイス	○					○
画面上のドキュメント	○					○
固定書式ドキュメント		○				○
画像	○		○			○
ビデオとオーディオ				○		○
2D のグラフィックス	○		○			○
3D のグラフィックス					○	○

メディアサービス—テキスト

- フロードキュメント
 - FlowDocument タグで定義したドキュメント

ビューワーで表示します。

- FlowDocumentViewer
- FlowDocument ScrollViewer

エムナウのプログラミングのページ

エムナウのプログラミングのページ
 エムナウの とどけもの エムナウ
 Blog@AllLight エムナウ
 Blog@Wankuma 目次ホーム ブログ
 (わんくま) ブログ(AllLight) 連絡する
 本サイトについて 利用状況カウンタ
 イメージ ギャラリー 似顔絵 カテゴリ
 VS2005 C# VS2003 C# VS2003 C++
 SQL2000 SQLサーバー Windowsサー
 バー 告知と雑談 Links #Blog(わん
 くま) #Blog(AllLight) @Programing! -
 ASP.NET By C# .NET - AllLight - C#の
 研究、調査してます!! 宇宙仮面の
 C# プログラミング ペがらぼ(ペガサス
 研究室) もっとドットNET C#,

VB.NET 入門 FITEA福井情報技術者協
 会 主婦と .NET と犬の記録
 HIRO's.NET MSMVP Visual C#
 2005/01-2007/12 2007年01月
 2005C# エムナウコントロール サ
 ンプル1 Windows Forms のための検
 証コントロールプロパティを追加して
 検証グループと比較条件の機能を実装
 しました。 posted @ 2007年01月11日
 今年もマイクロソフトからVisual
 Developer - C#のカテゴリでMVP表彰
 される事になりました。マイクロソフ
 トから Visual Developer - C#のカテゴ
 リで、MVPの再表彰していただくこと
 になりました。よりいっそう自己の研
 鑽とコミュニティ発展の協力を努める
 よう頑張ります。どきどきものでした
 が応援・協力していただいた皆様あり

フロードキュメント

DEMO



リソースとスタイルとテンプレート

- <Window.Resources>の中に単独のプロパティをリソースとして保存して使い回しができます。
- <Window.Resources>の中に複数のプロパティをスタイルとして保存して使い回しができます。
- テンプレートはスタイルの拡張でオブジェクトのビジュアル ツリーを含めることができます。

リソースとスタイルとテンプレート

- リソース
 - 特別な背景色を保存して再利用する
- スタイル
 - ボタンのサイズや背景色・前景色を保存して再利用する
- テンプレート
 - ボタンの形状も含めサイズや背景色・前景色を保存して再利用する
 - ボタンに図とテキストを入れた形で保存して再利用する



リソースとスタイル とテンプレート

DEMO

イベントとキーとコマンド

やっぱりWPFって
見た目だけかよ

違います!!!

イベント

- イベントの追加はプログラムからでもデザイナーからでも作成できます。
- もちろんイベントの実装はプログラムで行います。

XAML

```
<RepeatButtonx:Name="UpArrow" Click="UpArrow_Click" />
```

C#

```
UpArrow.Click+= new RoutedEventHandler(UpArrow_Click);
```

VB.NET

```
Addhandler UpArrow.Click, AddressOf UpArrow_Click
```

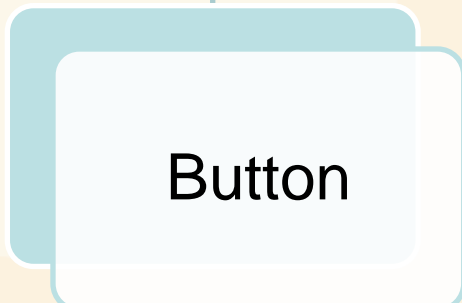
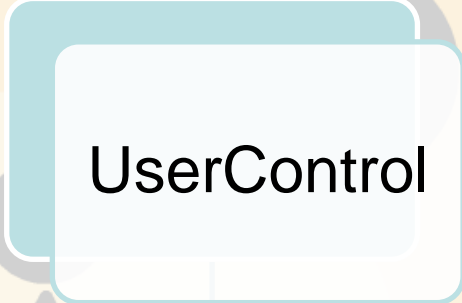
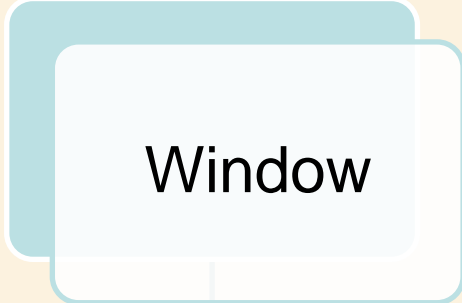
イベント

トンネル型
Previewが前に付く

1

2

3



6

5

4

バブル型



Buttonが処理をするイベントはバブル型では発生しない。

コマンド

- XAMLでコマンドを指定することによりプログラムのコマンドを実行できます。
- 以下の例では「Save」ボタンを「ApplicationCommands.Save」コマンドに、バインドしています。

```
<Button Name="buttonSave" Command=" Save" />
```

保存

```
</Button >
```


コマンド

```
CommandBinding SaveCommandBinding =
    new CommandBinding(
        ApplicationCommands.Save,
        SaveCommandExecuted,
        SaveCommandCanExecute);
this.CommandBindings.Add(SaveCommandBinding);

void SaveCommandExecuted(object target, ExecutedRoutedEventArgs e)
{
    // 実際の保存処理
}
void SaveCommandCanExecute(object sender, CanExecuteRoutedEventArgs e)
{
    e.CanExecute= true;
}
```

キー

- キーをコマンドにバインディングできます。
下は「コントロール+S」を
「ApplicationCommands.Save」コマンドに、
バインドした例です。

```
KeyBinding SaveKeyBinding =  
    new KeyBinding(  
        ApplicationCommands.Save,  
        Key.S,  
        ModifierKeys.Control);  
this.InputBindings.Add(SaveKeyBinding);
```

イベントとキーとコマンド

DEMO

Microsoft®

ready
for a **new day**

 **Office** Microsoft®

 **Windows Vista™**

Microsoft
Exchange Server 2007