**4. 버튼과 컨트롤**

**[WPF 에서 Control Class]**

* WPF 에서는 Window Program Interface보다 조금 더 한정적으로 사용된다.

윈도우 프로그램에서는 사용자가 화면상에 보이는 모든 것이 컨트롤로 간주했다면

WPF에서는 사용자와 상호적용을 하는 엘리먼트로 제한을 한다.

**[WPF 에서의 상호작용이란?]**

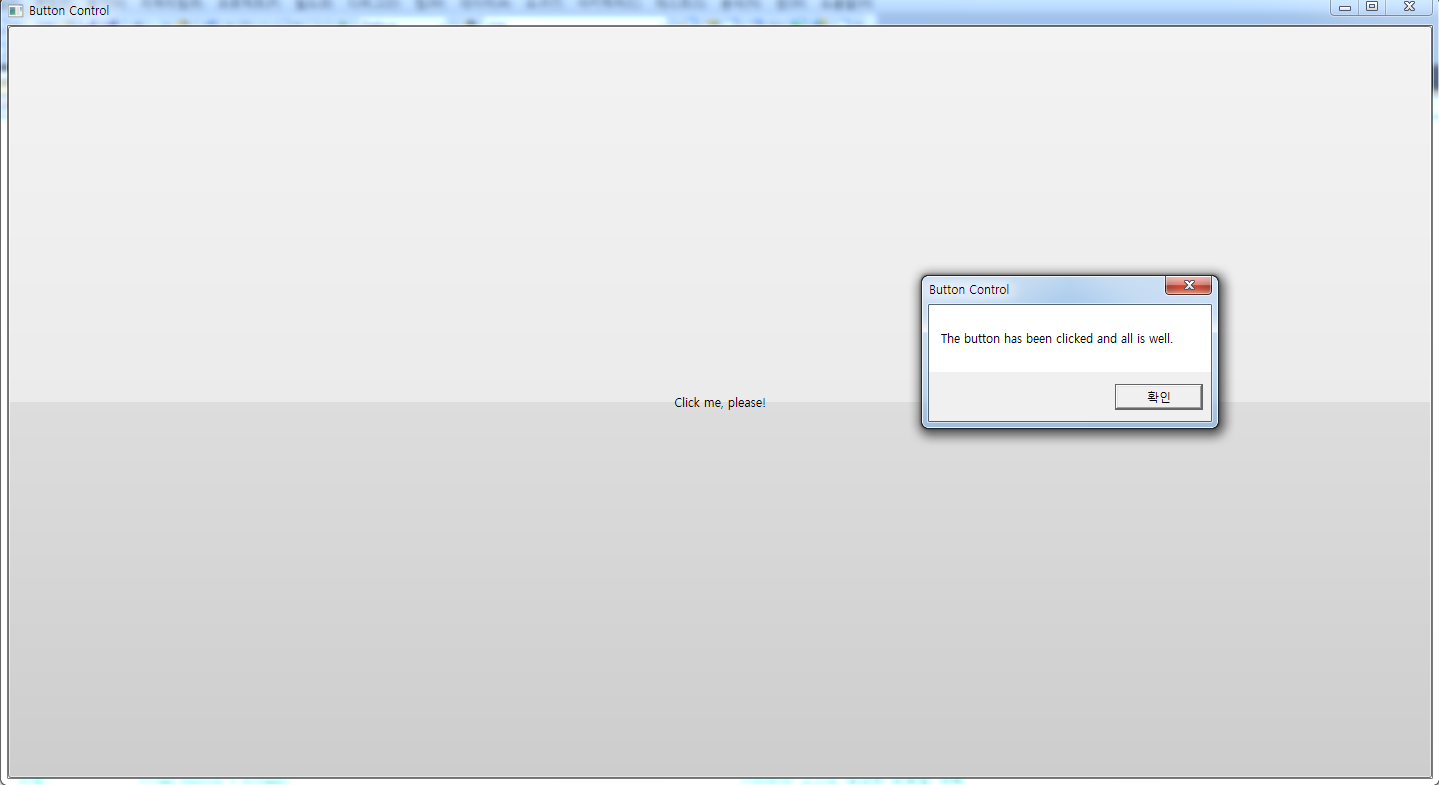
* 사용자가 마우스나 키보드를 눌렀을 때 어떤 종류의 피드백을 제공하는 것을 의미한다.
* TextBlock, Image, Shape 엘리먼트는 모든 키보드 마우스 스타일러스펜등의 입력을 받을수 있을 수 있었지만 입력을 따로 처리하지 않고 무시했었다.
* 이에 반해 Contorl은 능동적으로 사용자의 입력을 감지하고 처리하는 장점이 있다.

**[Control Class의 계층도]**

|  |
| --- |
| **Object**  **DispatcherObject**  **DependencyObject**  **Visual(absetract)**  **UIElement**  **FrameworkElement**  **Control** |

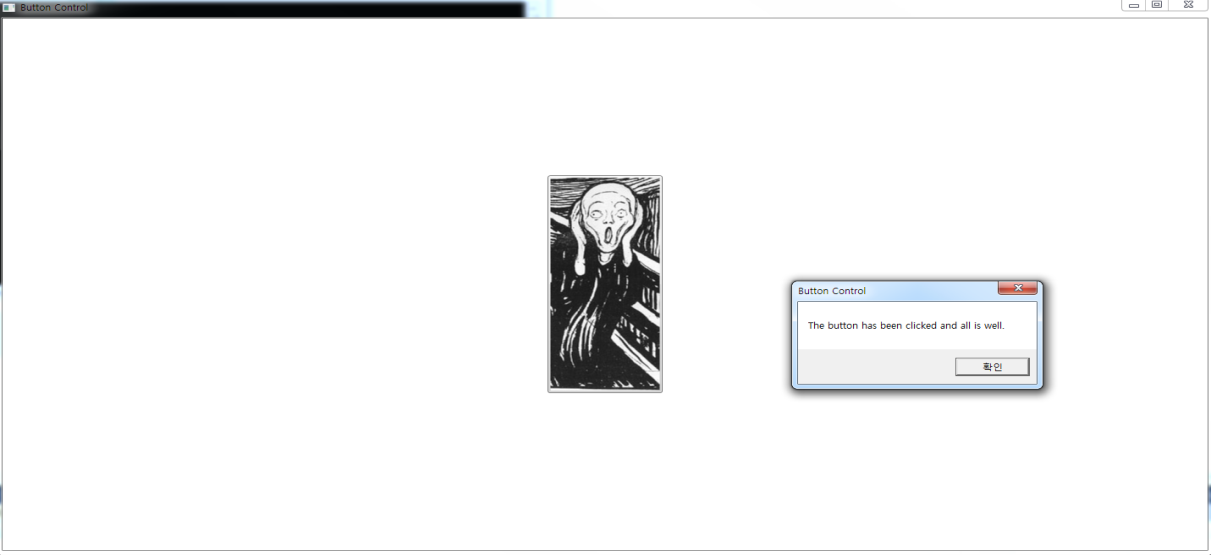
* Control Class를 상속 받는 클래스는 50개 이상이며 Button, ListBox, EditFild등이 있다.
* 이들의 클래스는 System.Windows.Controls와 System.Windows.Controls.Primitives 네임스페이스에 존재하며 Primitives 네임스페이스에는 Control를 상속하지 않는 다른 클래스도 정의 되어있다. (TogleBox등…)

**[ButtonClass]**



* WPF에서 전형적인 컨트롤 중에 하나인 버튼은 Class로 표현하며 ButtonClass에는 Content 프로퍼티와 Click이라는 이벤트가 존재한다.
* Window 와 ButtonBase 모두 동일한 Content 프로퍼티를 가지고 있다
* 이것이 의미하는 바는 Windows의 Contents에 대힙할수 있는 모든 객체는 버튼의 Content로 사용될 수 있으며 버튼을 비트맵, Shape 객체나 서식이 있는 텍스트로 표시하거나 Button 객체의 Content에 다른 객체를 연결할 수도 있다

**[ButtonCotent에 이미지 삽입]**

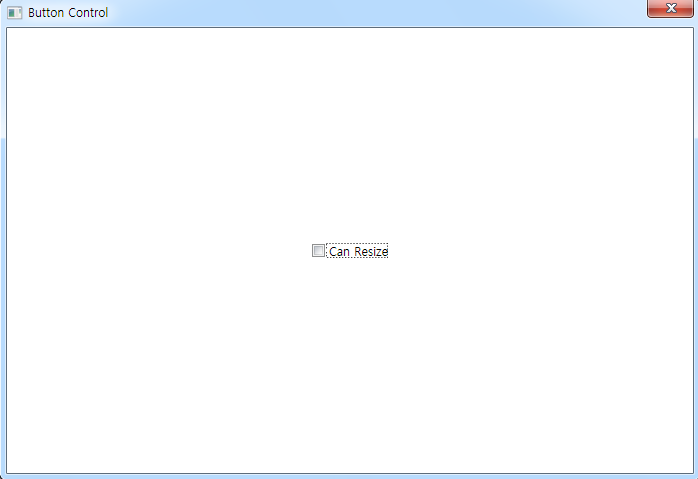
****

* WPF에서의 리소스 개념(어셈블리 리소스) 프로젝트의 일부분 (이진파일) 보통은 실행파일이나 DLL 의 일부가 된다.
* 이미지를 추가하기 위해서는 System.Windows.Media.Imaging를 추가 하여야 한다.
* 추가 방법
* 프로젝트 이름 -> 추가 -> 기존항목 -> 항목을 찾아서 추가 -> 추가된 항목을 선택 -> 속성 -> 빌드작업 을 리소스로 변경
* Uri생성자를 통하여 경로를 지정
* Uri uri = new Uri("pack://application:,,/munch.png");

**[Command Binding]**

* 특정한 명령을 이벤트 핸들러와 결합시키는 방법
* WPF에서 제시하는 바인딩 유형중에 하나.
* UIElemenent 클래스 안에 CommandBindings라는 프로퍼티로 존재 이는 CommandBindings의 컬렉션의 개념
* Control클래스와 Windows클래스는 이 CommandBinding을 UIElemenent에서 상속한다.

**[Toggle Button]**



* Toggle Button의 계층도

|  |
| --- |
| **Control**  **ContentControl**  **ButtonBase (abstract)**  **Button**  **GridWiewColumnHeader**  **RepeatButton**  **ToggleButton**  **CheckBox**  **RadioBox** |

* ToggleBotton 은 추상 클래스가 아니라서 ToggleButton의 타입 객체도 생성할 수 있다
* ToggleBotton을 사용하기 위해서는 System.Windows.Controls.Primitives 추가해 줘야 한다.
* 여러가지 항목중에서 하나만 생성할수 있는 Cotrol RadioBox, CheckBox 등이 이에 해당함

**[Data Binding]**

* 기본적으로 표현값이 Boolean인 컨트롤 등과 Boolean프로퍼티는 결합이 될 수 있는데 이를 데이터 바인딩이라고 한다.
* 데이터 바인딩이 적용된 컨트롤등은 해당 프로퍼티를 초기화를 하지 않아도 되며 이벤트 핸들러도 따로 제작할 필요가는 장점이 있다.
* 데이터 바인딩을 사용하기 위해서는 FramewirkElement 클래스에서 상속한 SetBinding 메소드를 사용해야한다.
* 데이터 바인딩후 DataContext프로퍼티에 바인딩될 Windows 객체를 지정해야한다.

**[TextBox]**

* Text박스는 컨텐츠가 존재하지만 항상 텍스트라 ContentControl가 Content 대신 Text 프로퍼티를 사용한다
* 이 아니다
* TextBox는 Control를 상속 받기 때문에 폰트지정이나 브러시도 지정할수 있다.
* RichTextBox는 텍스트 일부를 다양한 서식으로 변경할수 있다.
* RichTextBox와 TextBox의 차이는 워드프로세서와 메모장
* RichTextBox는 일반 텍스트,서식있는 텍스트(RTF) XAMEL의 XML파일등을 지원한다.
* TextBox의 계층 구조

|  |
| --- |
| **Control**  **TextBoxBase (abstract)**  **TextBox**  **RichTextBox** |