31장 비트맵, 브러시, 드로잉

목표

- WPF에서 다양한 비트맵 알아보기

- WPF에서 래스터 그래픽과 벡터 그래픽을 혼합하는 방법 알아보기

- 2장에서 시작된 브러시에 관한 부분 마무리(Drawing 객체, 비트맵, Visual 객체)

p.1233 **전통방식과 WPF 방식의 차이**

**전통적인 컴퓨터 그래픽**

- 래스터 그래픽(raster graphic) – 비트맵(실세계의 이미지를 인코딩..) / Image 클래스로 그림

- 벡터 그래픽(vector graphic) – 라인,커브,색칠된 영역 / Shape 파생 클래스로 그림

**하지만**, WPF의 그래픽 기능은 전통적인 컴퓨터 그래픽 방식과는 달리 확실히 나누기 어렵다.

Image클래스는 비트맵 뿐만 아니라 DrawingImage도 지원

\*Drawing - 벡터 그래픽 엘리먼트들로 구성된 그림들을 일컫음

Drwaing 클래스도 벡터 그래픽만뿐만 아니라 래스터 그래픽, 동영상비디오를 지원

p.1234

**WPF에서 제공하는 BitmapSource 추상클래스와 그 파생클래스**

Object

DispatcherObject(abstract)

DependencyObjct

Freezable(abstract)

Animatable(abstract)

**ImageSource(abstract) – Height, Width(장치 독립적 단위 사용) 프로퍼티 존재**

**BitmapSource(abstract) – PixelWidth, PixelHeight, DpiX, DpiY, Format, Palette**

**(읽기 전용 프로퍼티) 존재**

BitmapFrame(abstract)

**BitmapImage – Uri 객체를 이용한 생성자로 많이 사용**

CachedBitmap

ColorConveredBitmap

CroppedBitmap

FormatConvertedBitmap

RenderTargetBitmap

TransformedBitmap

**DrawingImage – 벡터그래픽을 그리기 위해 사용됨**

p.1236 (이미 존재하는 비트맵을 메모리에 불러오는 것을 보았음..)

**새로 만든 비트맵에 이미지를 넣는 방법**

1. 비트맵 위에 그림을 그리는 방법(BitmapEncoder 파생 클래스의 Save메소드 사용)

① RenderTargetBitmap 객체 생성

② DrawingVisual 객체 생성

③ RenderTargetBitmap 객체의 Render메소드 호출(DrawingVisual 객체), 지우고 싶으면 Clear 호출

④ Image 객체를 만들고 해당 객체의 Source 프로퍼티를 설정

\*RenderTargetBitmap 객체란?

현재 화면에 출력되고 있는 Visual객체나 임의로 생성한 Visual 객체를 Image로 변환 해야 할 때 사용

예제 DrawGraphicsOnBitmap, DrawButtonsOnBitmap 참고…

2. 이미지를 구성하는 비트맵의 실제 비트를 직접 설정하는 방법

- BitmapSource의 정적 메소드 Create를 이용한다.

(컬러 테이블을 필요로 하는 비트맵 경우 BitmapPalette 타입의 객체를 사용해야 한다)

예제 CreateIndexedBitmap, CreateFullColorBitmap 참고…

p1245

CroppedBitmap – BitmapSource를 자름

FormatConvertedBitmap – BitmapSource용 픽셀 형식 변환 기능 제공

TransformedBitmap – BitmapSource의 크기를 조정하고 회전

p1247

Image가 벡터그래픽 묶음을 그려준다는 개념을 확인…

예제 ImageDisplaysVectorGraphics.xaml 참고…

Object

DispatcherObject(abstract)

DependencyObjct

Freezable(abstract)

Animatable(abstract)

Drawing(abstract)

**DrawingGroup – Children 프로퍼티를 가져서 다른 Drawing 객체들을 저장하는 컬렉션(하나의 드로잉으로 결합하는 방법, 조합된 드로잉을 제어할 수 있는 다른 프로퍼티를 가짐)**

**GeometryDrawing**

GlyphRunDrawing

ImageDrawing

VideoDrawing

\* DrawingGroup 클래스 - Children 프로퍼티를 갖는다

(다른 Drawing 객체들을 저장할 수 있다)

(∴ 벡터그래픽과 비트맵을 하나의 드로잉으로 결합 가능)

- GlypeRunDrawing 클래스 - GlypeRun 객체와 전경색 브러시를 결합

(특정한 포트와 크기를 사용하는 문자들의 집합, 사용하기 어려움)

- ImageDrawing 객체 - ImageSource 프로퍼티가 가리키는 비트맵을 참조(특정한 크기 부여하기 위해서 Rect 프로퍼티 설정)

- VideoDrawing 클래스 - MediaPlayer 타입의 Player 프로퍼티와 크기를 지정하는 Rect 프로퍼티를 갖음

\* DrawingGroup 클래스의 역할

- 이미지를 조합(ImageDrawing 객체, GemoetryDrawing 객체를 조합할 때 많이 사용)

- ImageDrawing이 Rect 프로퍼티를 갖고 있음(비트맵 크기 지정과 2차원 좌표 공간 안에서의 위치도 지정)

- 조합된 드로잉을 제어함(ClipGeometry 프로퍼티에 Geomtry 객체 설정, OpacityMask 프로퍼티를 이용한 투명도)

- BitmapEffect - UlElement와 DrawingGroup 양쪽 모두에 존재, Visual Effect 효과를 줌

- Transform 프로퍼티 (드로잉 전체에 변환을 적용)

p1253

- Children 프로퍼티 뿐만 아니라 조합된 드로잉을 제어할 수 있는 다른 프로퍼티들도 제공

- ClipGeometry : 특정 부분 잘라냄

- Opacity : 투명도 설정

- OpacityMask : Brush객체를 할당하여 알파 채널의 값을 이용해 투명도를 설정, Opacity보다 다양한 효과

- BitmapEffect : 이미지에 원하는 효과를 넣어줌(워터마크, 겉테두리, 뿌연느낌 등등..)

- Transform : 크기 조절

- RenderTransform : 애니메이션 효과를 넣을 수 있음

p1256

DrawingContext 클래스 - 로우레벌 드로잉 메소드(WPF App의 기능이라고 부를만한 기능들이 정의되어있음)

==========================================================================

p.1258

Object

DispatcherObject(abstract)

DependencyObject

Freezable(abstract)

Animatable(abstract)

Brush(abstract)

GrdientBrush(abstract)

LinearGradientBrush

RadialgradientBrush

SolidColorBrush

TileBrush(abstract)

**DrawingBrush - GeometryDrawing객체(Drawing,미디어 플레이어)**

**ImageBrush - ImageSource(BitmapImage,DrawingImage,그래픽 오브젝트)**

**VisualBrush - DrawingVisual 객체, 컨트롤, 사용자 인터페이스의 다른 부분 그리기위해**

- TileBrush에 정의된 8개의 프로퍼티를 모두 제어 가능

p1273

INotifyPropertyChanged 인터페이스 – 속성값이 변경되었음을 알리는 클래스  
(Public 이벤트인 PropertyChanged를 발생함)

DispatcherTimer 클래스 – 지정된 시간 간격과 우선 순위로 처리되는 Dispatcher큐로 통합되는 타이머

TimeSpan – 시간간격(수치)