

# 게임프로그래밍

---

## 봇

박종승

---

Dept. of CSE, Incheon Nat. Univ.  
jong@inu.ac.kr  
<http://ecl.inu.ac.kr>

# 목차

---

- 붓이란
- 컬러 기초
- 단색 붓 사용하기
- 선형 계조 붓 사용하기
- 방사형 계조 붓 사용하기
- 비트맵 붓 사용하기
- 확장 모드
- 붓 변환하기

# 붓이란

---

- 붓(brush)?
  - 붓의 출력물(예: 물감)로 그릴 영역을 색칠함
  - 붓의 공통 인터페이스: ID2D1Brush
    - 붓의 공통 특징들에 대한 구현 (예: 불투명도의 set/get, 붓의 변환)
  - 붓은 장치에 의존인 자원임
    - ID2D1RenderTarget를 통해 생성됨
    - 렌더타겟이 재생성될 때마다 붓도 재생성해야함.
  - 붓의 타입에 따라서 출력물의 형태가 다름
- 붓의 사용
  - 렌더타겟의 Draw/Fill 함수들의 인자로 사용

# 붓 타입

- D2D가 지원하는 붓 타입: 4가지 (ID2D1XXXBrush) 01.BrushTypesExample
  - 단색 붓: ID2D1SolidColorBrush
  - 선형 계조 붓: ID2D1LinearGradientBrush
  - 방사형 계조 붓: ID2D1RadialGradientBrush
  - 비트맵 붓: ID2D1BitmapBrush
- 붓 생성 방법
  - ID2D1RenderTarget::CreateXXXBrush 함수 호출
    - XXX=SolidColor,LinearGradient,RadialGradient,Bitmap

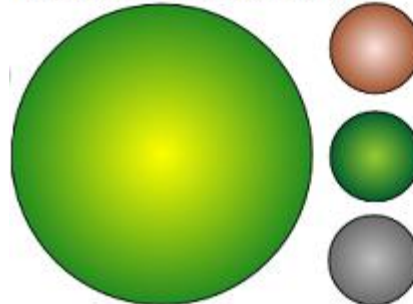
ID2D1SolidColorBrush



ID2D1LinearGradientBrush



ID2D1RadialGradientBrush



ID2D1BitmapBrush



# 컬러 기초

---

- 컬러의 표현
  - D2D1\_COLOR\_F 구조체
    - D3D에서의 D3DCOLORVALUE 구조체와 동일함
      - struct { float **r**; float **g**; float **b**; float **a**; } D3DCOLORVALUE;
  - 4개의 요소: red, green, blue, alpha
  - 각 실수의 범위: 0.0~1.0
    - r,g,b: 0은 색이 전혀 없음, 1은 색이 완전히 있음
    - a: 0은 완전히 투명함, 1은 완전히 불투명함

# 컬러 기초'

---

- 연산자를 지원하는 클래스 표현
  - 편리 클래스: D2D1::ColorF 클래스
    - 구조체 D2D1\_COLOR\_F를 상속하였음
    - 편리한 색 정의를 위한 상수들 및 생성자들을 제공함
    - 생성자에서, alpha를 지정하지 않으면, 디폴트 1.0이 사용됨
  - 생성자1: ColorF(float r, float g, float b, float a = 1.0);
    - 예: D2D1::ColorF(0.93f, 0.94f, 0.96f, 1.0f);
  - 생성자2: ColorF(Enum knownColor, float a = 1.0);
    - knownColor: 미리 정의한 컬러들 (UINT32값을 가짐)
      - 예: Red = 0xFF0000, Green = 0x008000, Lime = 0x00FF00, Blue = 0x0000FF
    - 예: D2D1::ColorF(D2D1::ColorF::Black, 1.0f);
  - 생성자3: ColorF(UINT32 rgb, float a = 1.0);
    - 예: D2D1::ColorF(0x9ACD32, 1.0f);

# 참고: 미리 정의한 컬러들

White #FFFFFF	Gainsboro #DCCDCD	CadetBlue #5F9EA0	DarkCyan #008B8B	RosyBrown #BC8F8F	Salmon #FA8072	AliceBlue = 0xf0f8ff,	GhostWhite = 0xf8f8ff,	NavajoWhite = 0xffdead,
LightGray #D3D3D3	Silver #C0C0C0	Teal #008080	Turquoise #40E0D0	IndianRed #CD5C5C	LightCoral #F08080	AntiqueWhite = 0xfaebd7,	Gold = 0xffd700,	Navy = 0x000080,
DarkGray #A9A9A9	Gray #808080	LightSeaGreen #20B2AA	DarkTurquoise #00CED1	DarkSalmon #E9967A	LightSalmon #FFA07A	Aqua = 0x00ffff,	Goldenrod = 0xdaa520,	OldLace = 0xfdf5e6,
DimGray #696969	Black #000000	MediumAquamarine #66CDAA	Aquamarine #7FFFD4	SandyBrown #F4A460	Orange #FFA500	Aquamarine = 0x7fffd4,	Green = 0x008000,	Olive = 0x808000,
		Aqua #00FFFF	Cyan #00FFFF	DarkOrange #FF8C00	Coral #FF7F50	Azure = 0xf0ffff,	GreenYellow = 0xadff2f,	OliveDrab = 0x6b8e23,
GreenYellow #ADFF2F	LightGreen #90EE90	MediumTurquoise #48D1CC	PaleTurquoise #AFEEEE	Tomato #FF6347	OrangeRed #FF4500	Beige = 0xf5f5dc,	Honeydew = 0xf0ffff0,	Orange = 0xffa500,
PaleGreen #98FB98	YellowGreen #9ACD32	PowderBlue #B0E0E6	LightBlue #ADD8E6	Red #FF0000	Crimson #DC143C	Bisque = 0xffe4c4,	HotPink = 0xff69b4,	OrangeRed = 0xff4500,
Chartreuse #7FFF00	LawnGreen #7CFC00	LightSkyBlue #87CEFA	SkyBlue #87CEEB	Firebrick #B22222	Brown #A52A2A	Black = 0x000000,	IndianRed = 0xcd5c5c,	Orchid = 0xda70d6,
Lime #00FF00	LimeGreen #32CD32	DeepSkyBlue #00BFFF	LightSteelBlue #B0C4DE	DarkRed #800000	Maroon #800000	BlanchedAlmond = 0xffebcd,	Indigo = 0x4b0082,	PaleGoldenrod = 0xeeee8aa,
Green #008000	ForestGreen #228B22	CornflowerBlue #6495ED	DodgerBlue #1E90FF	SaddleBrown #8B4513	Sienna #A0522D	Blue = 0x0000ff,	Ivory = 0xfffff0,	PaleGreen = 0x98fb98,
DarkGreen #006400	MediumSpringGreen #00FA9A	SteelBlue #4682B4	RoyalBlue #4169E1	DarkGoldenrod #B8860B	Chocolate #D2691E	BlueViolet = 0x8a2be2,	Khaki = 0xf0e68c,	PaleTurquoise = 0xafeeee,
SpringGreen #00FF7F	MediumSeaGreen #3CB371	Blue #0000FF	MediumBlue #0000CD	Peru #CD853F	Goldenrod #DAA520	Brown = 0xa52a2a,	Lavender = 0xe6e6fa,	PaleVioletRed = 0xdb7093,
SeaGreen #2E8B57	DarkSeaGreen #8FBC8F	DarkBlue #00008B	MidnightBlue #191970	Gold #FFD700	Yellow #FFFF00	Chocolate = 0xd2691e,	LavenderBlush = 0xffd0f5,	PapayaWhip = 0xffefd5,
OliveDrab #6B8E23	Olive #808000	Navy #000080	LightSlateGray #778899	LightYellow #FFFFE0	LightGoldenYellow #FAFAD2	Coral = 0xff7f50,	LawnGreen = 0x7cfc00,	PeachPuff = 0xffdab9,
DarkOliveGreen #556B2F		DarkSlateGray #2F4F4F	SlateGray #708090	LemonChiffon #FFFACD	PapayaWhip #FFEFD5	CornflowerBlue = 0x6495ed,	LemonChiffon = 0xffffac,	Peru = 0xcd853f,
		DarkSlateBlue #483D8B	MediumSlateBlue #7B68EE	Moccasin #FFE4B5	NavajoWhite #FFDEAD	Cornsilk = 0xffff8c,	LightBlue = 0xad8e6,	Pink = 0xffc0cb,
		SlateBlue #6A5ACD	Indigo #4B0082	Wheat #F5DEB3	BurlyWood #DEB887	Crimson = 0xdc143c,	LightCoral = 0xf08080,	Plum = 0xdda0dd,
FloralWhite #FFFAF0	Ivory #FFFFF0	BlueViolet #8A2BE2	MediumPurple #9370DB	Tan #D2B48C	PaleGoldenrod #EE88AA	Cyan = 0x00ffff,	LightGray = 0xd3d3d3,	PowderBlue = 0xb0e0e6,
SeaShell #FFF5EE	OldLace #FDF5E6	DarkViolet #9400D3	DarkOrchid #9932CC	Khaki #F0E68C	DarkKhaki #BDB76B	DarkBlue = 0x00008b,	LightPink = 0xffb6c1,	Purple = 0x800080,
Cornsilk #FFF8DC	GhostWhite #F8F8FF	DarkMagenta #800080	Purple #800080			DarkGoldenrod = 0xb8860b,	LightSalmon = 0xffa07a,	Red = 0xff0000,
WhiteSmoke #F5F5F5	MintCream #F5FFFA	MediumOrchid #BA55D3	Magenta #FF00FF			DarkGray = 0xa9a9a9,	LightSeaGreen = 0x20b2aa,	RosyBrown = 0xbc8f8f,
Azure #F0FFFF	Honeydew #F0FFF0	Fuchsia #FF00FF	Orchid #DA70D6			DarkGreen = 0x006400,	LightSkyBlue = 0x87cefa,	RoyalBlue = 0x4169e1,
AliceBlue #F0F8FF	LightCyan #E0FFFF	Thistle #D8BFD8	PaleVioletRed #DB7093			DarkKhaki = 0xbdb76b,	LightSlateGray = 0x778899,	SaddleBrown = 0x8b4513,
Linen #FAFAD2	AntiqueWhite #FAEBD7	MediumVioletRed #C71585	DeepPink #FF1493			DarkMagenta = 0x8b008b,	LightSteelBlue = 0xb0c4de,	Salmon = 0xfa8072,
Beige #F5F5DC	BlanchedAlmond #FFEBCD	HotPink #FF69B4	LightPink #FFB6C1			DarkOrange = 0xff8c00,	LightYellow = 0xffffe0,	SandyBrown = 0xf4a460,
Bisque #FFE4C4	PeachPuff #FFDAB9	Pink #FFC0CB	MistyRose #FEE4E1			DarkRed = 0x8b0000,	Lime = 0x32cd32,	SeaGreen = 0x2e8b57,
LavenderBlush #FFF0F5	Lavender #E6E6FA					DarkSalmon = 0xe9967a,	Linen = 0xfaf0e6,	SeaShell = 0xffffe6,
						DarkSeaGreen = 0x2f4f4f,	Magenta = 0xff00ff,	Sienna = 0xa0522d,
						DarkSlateBlue = 0x483d8b,	Maroon = 0x800000,	Silver = 0xc0c0c0,
						DarkSlateGray = 0x2f4f4f,	MediumAquamarine = 0x66cdaa,	SkyBlue = 0x87ceeb,
						DarkTurquoise = 0x00ced1,	MediumBlue = 0x0000cd,	SlateBlue = 0x6a5acd,
						DarkViolet = 0x9400d3,	MediumOrchid = 0xba55d3,	SlateGray = 0x708090,
						DeepPink = 0xff1493,	MediumPurple = 0x9370db,	Snow = 0xfffff0,
						DeepSkyBlue = 0x00bfff,	MediumSeaGreen = 0x3cb371,	SpringGreen = 0x00ff7f,
						DimGray = 0x696969,	MediumSlateBlue = 0x7b68ee,	SteelBlue = 0x4682b4,
						DodgerBlue = 0x1e90ff,	MediumSpringGreen = 0x00fa9a,	Tan = 0xd2b48c,
						Firebrick = 0xb22222,	MediumTurquoise = 0x48d1cc,	Teal = 0x008080,
						FloralWhite = 0xfffaf0,	MediumVioletRed = 0xc71585,	Thistle = 0xd8bfd8,
						ForestGreen = 0x228b22,	MidnightBlue = 0x191970,	Tomato = 0xff6347,
						Fuchsia = 0xff00ff,	MintCream = 0xf5fffa,	Turquoise = 0x40e0d0,
						Gainsboro = 0xdcdcdc,	MistyRose = 0xffe4e1,	Violet = 0xee82ee,
							Moccasin = 0xffe4b5,	Wheat = 0xf5deb3,
								White = 0xffffffff,
								WhiteSmoke = 0xf5f5f5,
								Yellow = 0xffff00,
								YellowGreen = 0x9acd32,

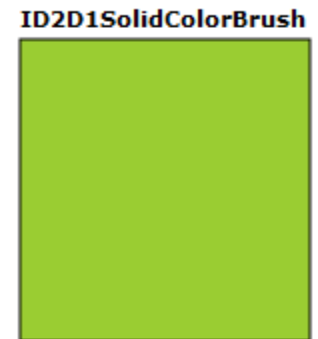
# 단색 붓 사용하기

01.BrushTypesExample

- 단색 붓 객체
  - ID2D1SolidColorBrush
- 단색 붓 객체 생성하기
  - ID2D1RenderTarget::CreateSolidColorBrush 함수 호출
- 참고: 단색 붓: 상대적으로 생성 비용이 저렴함
  - 필요할 때마다 매번 생성해도 성능에 큰 영향이 없음
    - 계조 붓이나 비트맵 붓은 비용이 큼: 재활용이 필요함
- 예

```
ID2D1SolidColorBrush* pGridBrush = NULL;
ID2D1SolidColorBrush* m_pBlackBrush = NULL;
ID2D1SolidColorBrush* m_pYellowGreenBrush = NULL;
pCompatibleRenderTarget->CreateSolidColorBrush(
    D2D1::ColorF(0.93f, 0.94f, 0.96f, 1.0f), &pGridBrush );
m_pRenderTarget->CreateSolidColorBrush(
    D2D1::ColorF(D2D1::ColorF::Black, 1.0f), &m_pBlackBrush );
m_pRenderTarget->CreateSolidColorBrush(
    D2D1::ColorF(0x9ACD32, 1.0f), &m_pYellowGreenBrush );
```

```
m_pRenderTarget->FillRectangle(&rcBrushRect, m_pYellowGreenBrush); //내부 채움
m_pRenderTarget->DrawRectangle(&rcBrushRect, m_pBlackBrush, 1, NULL); //검정색으로 외곽선
```





# 선형 계조 붓 사용하기

01.BrushTypesExample

- 선형 계조 붓 객체
  - ID2D1LinearGradientBrush
    - 선형 계조(한 계조축을 따라가며 정의됨)로 영역을 칠함
- 선형 계조 붓 객체 생성하기
  - ID2D1RenderTarget::CreateLinearGradientBrush 함수 호출
  - 명세: ID2D1GradientStopCollection 객체를 사용하여 계조축과 컬러를 명세함
    - 계조축 상의 지점의 위치와 컬러
- 예
  - 노란색과 녹색의 선형 계조로 그리기

ID2D1LinearGradientBrush



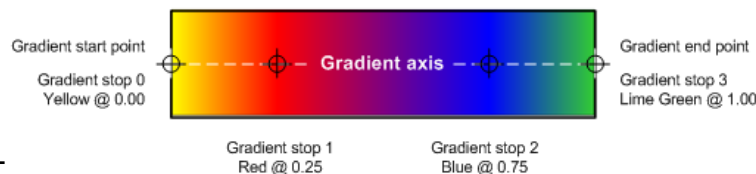
# 선형 계조 붓 사용하기'

- 단계 1: **D2D1\_GRADIENT\_STOP** 구조체 배열 만들기
  - 각 구조체는 각 "계조 명세 지점"(gradient stop)을 (컬러,위치)로 명시함
  - 계조 명세 지점(gradient stop)
    - 계조축 상의 한 지점에서의 위치와 컬러의 명시
    - 위치는 0.0(계조의 시작 위치)에서 1.0(계조의 종료 위치) 사이
  - 예: 첫 번째/두 번째 stop은 위치 0/1에 노란색/녹색을 명시

```
D2D1_GRADIENT_STOP gradientStops[2];  
gradientStops[0].color = D2D1::ColorF(D2D1::ColorF::Yellow, 1);  
gradientStops[0].position = 0.0f;  
gradientStops[1].color = D2D1::ColorF(D2D1::ColorF::ForestGreen, 1);  
gradientStops[1].position = 1.0f;
```

## – 그림 예

- 계조축: 흰색 점선
- 계조 명세 지점: 원 마크
  - 1<sup>st</sup>/2<sup>nd</sup>/3<sup>rd</sup>/4<sup>th</sup> 지점: 위치 0.0/0.25/0.75/1.0에서 노란색/붉은색/파랑색/녹색



# 선형 계조 붓 사용하기"

- 단계 2: ID2D1GradientStopCollection 객체 생성하기

- CreateGradientStopCollection 함수를 호출

- 인자1,인자2: D2D1\_GRADIENT\_STOP 객체들의 배열, 객체 개수,

```
ID2D1GradientStopCollection* pGradientStops = NULL;  
m_pRenderTarget->CreateGradientStopCollection( gradientStops, 2,  
                                                &pGradientStops );
```

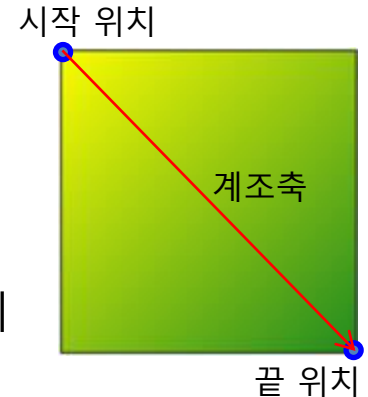
- 참고: CreateGradientStopCollection 함수의 확장 버전 사용

- 인자3: 컬러 보간이 수행되는 감마 공간을 선택: 1.0 또는 2.2
      - D2D1\_GAMMA\_2\_2 (디폴트) 또는 D2D1\_GAMMA\_1\_0
    - 인자4: 정규화된 [0,1] 영역 밖에서 계조하는 방법
      - D2D1\_EXTEND\_MODE\_XXX; XXX=CLAMP(디폴트) 또는 WRAP 또는 MIRROR
      - 구체적인 내용은: 뒤에서 설명하는 "확장 모드"를 참조
    - 위의 함수는 아래와 동일함

```
ID2D1GradientStopCollection* pGradientStops = NULL;  
m_pRenderTarget->CreateGradientStopCollection( gradientStops, 2,  
                                                D2D1_GAMMA_2_2, D2D1_EXTEND_MODE_CLAMP,  
                                                &pGradientStops );
```

# 선형 계조 붓 사용하기"

- 단계 3: `ID2D1LinearGradientBrush`를 생성
  - `CreateLinearGradientBrush` 함수를 호출
    - 인자1: 계조축을 명시
      - 구체적인 내용은: 다음쪽 참조
    - 인자2: 단계2에서 만든 `ID2D1GradientStopCollection` 객체



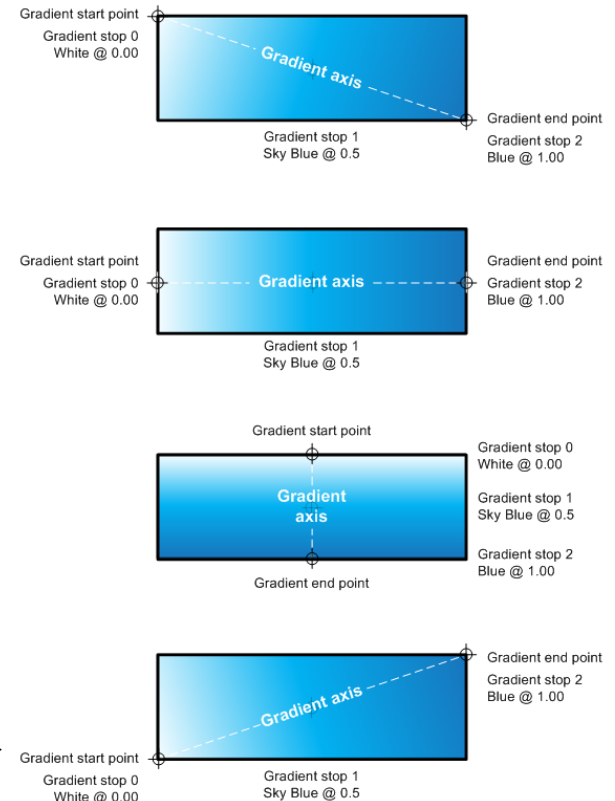
```
m_pRenderTarget->CreateLinearGradientBrush(  
    D2D1::LinearGradientBrushProperties( D2D1::Point2F(0, 0), D2D1::Point2F(150, 150) ),  
    pGradientStops, &m_pLinearGradientBrush );
```

- 단계 4: 그리기에 사용
  - 생성된 `ID2D1LinearGradientBrush`를 이용

```
m_pRenderTarget->FillRectangle(&rcBrushRect, m_pLinearGradientBrush);
```

# 선형 계조 붓 사용하기'''

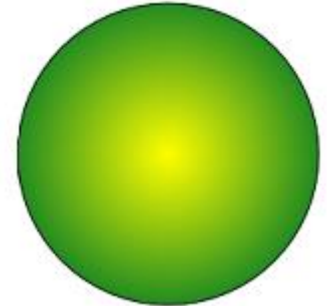
- 단계 3': (더 자세히)
  - 계조축의 명시
    - 위치좌표는 : 렌더타겟에서의 픽셀단위의 절대 위치에 매핑됨.
    - D2D1\_LINEAR\_GRADIENT\_BRUSH\_PROPERTIES 구조체
      - 필드: startPoint , endPoint
    - 구조체를 편리하기 만드는 도움함수: LinearGradientBrushProperties
      - 인자1,인자2: [시작지점,끝지점]
        - » 예시: 시작 위치 (0,0), 끝 위치 (150,150),
      - 추가적인 예시(오른쪽 그림): [(0,0), (150, 50)], [(0,25), (150,25)], [(75,0), (75,50)], [(0,50),(150,0)]
  - 참고: 붓을 생성한 후에도 계조축을 바꿀 수 있음
    - 붓의 함수 호출: SetStartPoint and SetEndPoint 함수



# 방사형 계조 붓 사용하기

## 01.BrushTypesExample

ID2D1RadialGradientBrush



- 방사형 계조 붓 객체
  - ID2D1RadialGradientBrush
    - 두 컬러를 계조 원점에서 타원 밖으로 향하면서 혼합함
      - 오른쪽 그림: 계조 원점이 타원의 중심인 경우
    - 계조 명세 지점을 다루는 방법은 이전과 동일함
- 방사형 계조 붓 객체 생성하기
  - ID2D1RenderTarget::CreateRadialGradientBrush 함수 호출
  - 인자1: D2D1\_RADIAL\_GRADIENT\_BRUSH\_PROPERTIES 구조체
- 예: 방사형 계조로 칠한 원
  - 두 계조 명세 지점
    - 위치 0.0에서 노란색, 위치 (1.0)에서 녹색.
  - 타원의 중심 (75,75), 계조 원점 오프셋 (0, 0), x/y-반지름 (75,75)

# 방사형 계조 붓 사용하기'

---

- 단계 1: D2D1\_GRADIENT\_STOP 구조체 배열 만들기 (이전과 동일)
  - 계조명세지점 D2D1\_GRADIENT\_STOP 구조체 배열 만들기

```
D2D1_GRADIENT_STOP gradientStops[2];  
gradientStops[0].color = D2D1::ColorF(D2D1::ColorF::Yellow, 1);  
gradientStops[0].position = 0.0f;  
gradientStops[1].color = D2D1::ColorF(D2D1::ColorF::ForestGreen, 1);  
gradientStops[1].position = 1.0f;
```

- 단계 2: ID2D1GradientStopCollection 객체 생성하기(이전과 동일)
  - CreateGradientStopCollection 호출

```
ID2D1GradientStopCollection* pGradientStops = NULL;  
m_pRenderTarget->CreateGradientStopCollection( gradientStops, 2,  
    D2D1_GAMMA_2_2, D2D1_EXTEND_MODE_CLAMP,  
    &pGradientStops );
```

# 방사형 계조 붓 사용하기"

---

- 단계 3: `ID2D1RadialGradientBrush` 생성
  - `ID2D1RenderTarget::CreateRadialGradientBrush` 함수 호출
  - 인자 1: `D2D1_RADIAL_GRADIENT_BRUSH_PROPERTIES`
    - 타원의 중심, 타원중심에 상대적인 계조 원점의 오프셋, 타원의 x/y-반지름
    - 편리함수: `RadialGradientBrushProperties`

```
m_pRenderTarget->CreateRadialGradientBrush(  
    D2D1::RadialGradientBrushProperties(  
        D2D1::Point2F(75, 75), D2D1::Point2F(0, 0), 75, 75 ),  
    pGradientStops, &m_pRadialGradientBrush );
```

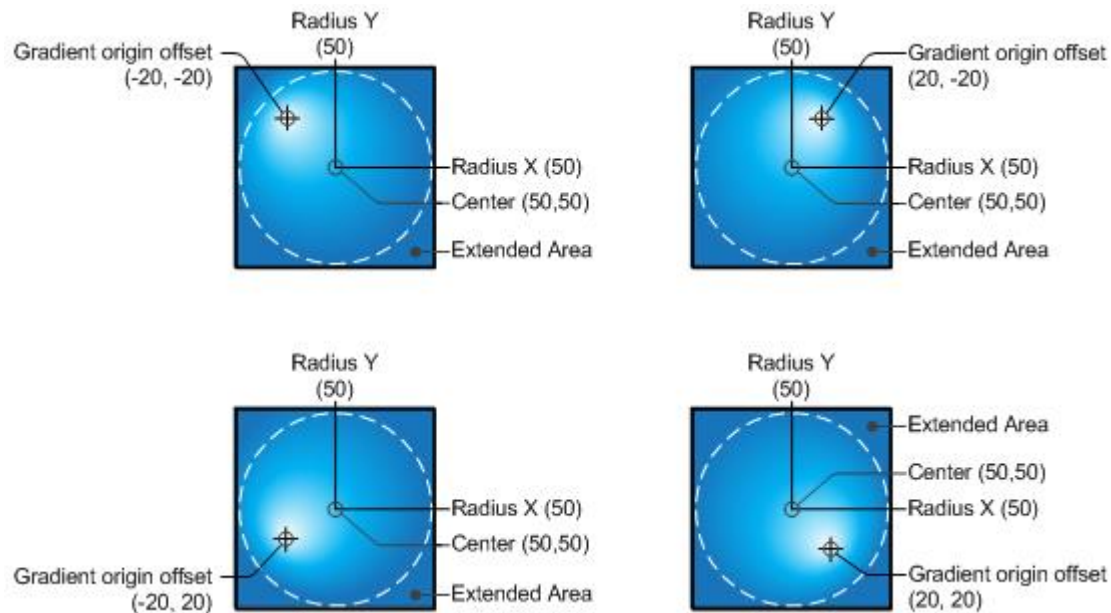
- 단계 4: 붓을 사용하여 색칠

```
m_pRenderTarget->FillEllipse(ellipse, m_pRadialGradientBrush);  
m_pRenderTarget->DrawEllipse(ellipse, m_pBlackBrush, 1, NULL);
```



# 방사형 계조 붓 사용하기"

- 단계 2: (더 자세히)
  - RadialGradientBrushProperties의 인자
    - 인자1: center: 타원의 중심
    - 인자2: gradientOriginOffset: 타원중심에 상대적인 계조 원점의 오프셋
    - 인자3,4: radiusX/radiusY: 타원의 x/y-반지름



# 비트맵 붓 사용하기

- 비트맵 붓

- ID2D1BitmapBrush

- 영역을 한 비트맵으로 칠함
    - 비트맵은 ID2D1Bitmap 객체로 표현됨

- 예: 식물 비트맵으로 칠한 정사각형



- 단계 1: 비트맵 객체 ID2D1Bitmap 를 준비

- 도움 함수 LoadBitmapFromResource 호출

- 이 함수는 차후의 "비트맵"에서 자세히 설명함

```
LoadBitmapFromResource( m_pRenderTarget, m_pWICFactory, L"FERN", L"Image", &m_pBitmap );
```

- 단계 2: 비트맵 붓 객체 ID2D1BitmapBrush 생성

- ID2D1RenderTarget::CreateBitmapBrush 호출

```
ID2D1BitmapBrush* m_pBitmapBrush;  
m_pRenderTarget->CreateBitmapBrush( m_pBitmap, &m_pBitmapBrush );
```

- 단계 3: 붓을 사용하여 그리기

```
m_pRenderTarget->FillRectangle(&rcBrushRect, m_pBitmapBrush);
```

# CreateBitmapBrush 인자

---

- CreateBitmapBrush 함수
- 함수1: CreateBitmapBrush(bitmap, bitmapBrush)
  - 인자1: ID2D1Bitmap\* bitmap,
  - 디폴트값:
    - 확장모드(extend mode)는 수평/수직 모두 CLAMP
    - 보간 모드(interpolation mode)는 LINEAR
    - 불투명도(opacity)는 1.0f
    - 변환(transform)은 항등행렬(identity matrix)
- 함수2: CreateBitmapBrush(bitmap, bitmapBrushProperties, bitmapBrush)
  - 인자1: 상동
  - 인자2: D2D1\_BITMAP\_BRUSH\_PROPERTIES
    - 인자1,2: 수평/수직방향 확장모드: D2D1\_EXTEND\_MODE\_XXX
    - 인자3: 스케일링 또는 회전 시에 비트맵 변환에 적용할 보간 모드: D2D1\_BITMAP\_INTERPOLATION\_MODE\_XXX, XXX=NEAREST\_NEIGHBOR, LINEAR
    - 편리함수: BitmapBrushProperties

# 확장 모드

---

- 확장 모드(extend mode)란?
  - 계조나 비트맵이 영역을 완전하게 칠하지 못할 경우
    - 명시된 확장 모드에 따라서 남은 부분을 채움
  - 정규화된 영역인  $[0,1]$ 의 외부의 영역을 채우는 방법을 명시함
- 확장 모드의 명시 방법
  - 비트맵의 경우
    - 붓에 수평/수직 확장 모드를 명시함
    - `ID2D1BitmapBrush::SetExtendModeX, SetExtendModeY`
  - 계조의 경우
    - 계조붓을 생성하기 위해서 `CreateGradientStopCollection` 함수를 호출할 때에 확장모드를 명시
    - 수평 및 수직 구분하지 않음

# 확장 모드'

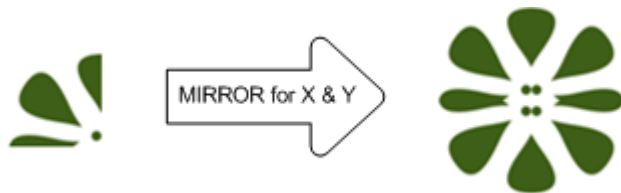
- 확장 모드의 종류

- D2D1\_EXTEND\_MODE\_XXX;  
XXX=CLAMP,WRAP,MIRROR
  - CLAMP: 붓 내용의 에지 픽셀들을 반복함
  - WRAP: 붓 내용을 반복함
  - MIRROR: WRAP과 동일함.  
단 붓 내용을 반복할 때에 뒤집기 함.
- 비트맵의 경우 예시: 오른쪽 그림

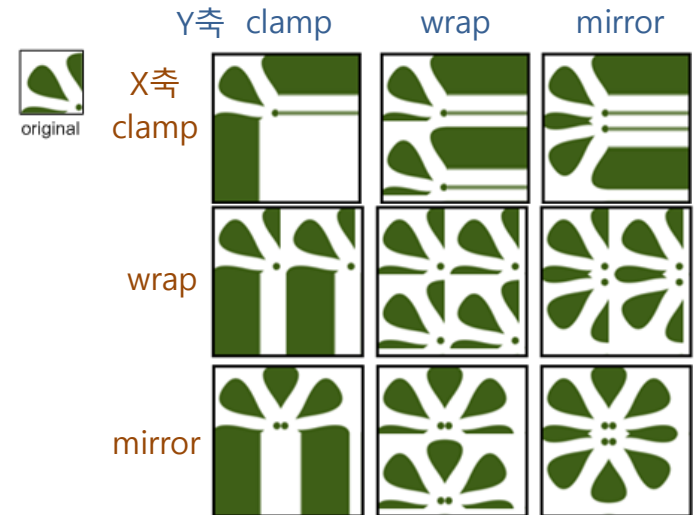
- 예

```
m_pBitmapBrush->SetExtendModeX(D2D1_EXTEND_MODE_MIRROR);  
m_pBitmapBrush->SetExtendModeY(D2D1_EXTEND_MODE_MIRROR);
```

```
m_pRenderTarget->FillRectangle(exampleRectangle, m_pBitmapBrush);
```



## 02.DrawExtendMode



# 붓 변환하기

---

- 붓으로 색칠할 때
  - 붓은 그릴 객체에 상대적으로 그리지 못함!!
  - 붓은 **렌더타겟의 원점 (0,0)에 상대적으로** 그림!!
- 계조의 이동
  - 선형 계조: LinearGradientBrushProperties 호출 시에, 시작 지점과 끝 지점을 지정
  - 방사형 계조: RadialGradientBrushProperties 호출 시에, 타원의 중심과 반지름을 지정
  - 비트맵의 이동: 붓의 SetTransform 함수 호출
    - 붓에만 영향을 미침

# 붓 변환하기'

- 예: 사각형[(100,100),(200,200)]에 비트맵 붓으로 그리는 경우
  - 붓을 변환하지 않는 경우: 비트맵을 렌더타겟의 원점에 맞추어 그림: 따라서 비트맵의 일부만 제대로 나타남.
  - 붓에 (50,50) 이동하는 변환을 적용한 경우: 비트맵이 이동되어 사각형이 모두 채워짐

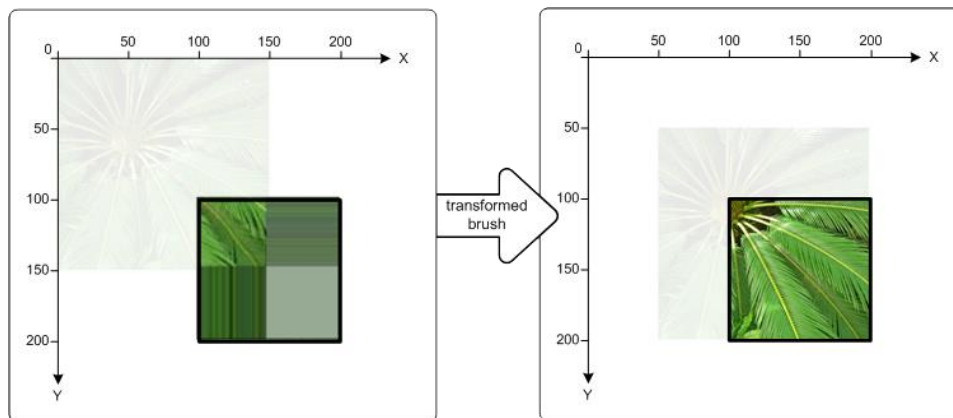
```
LoadBitmapFromResource ( m_pRenderTarget, m_pWICFactory, L"FERN", L"Image", &m_pBitmap );  
m_pRenderTarget->CreateBitmapBrush( m_pBitmap, &m_pBitmapBrush );
```

```
D2D1_RECT_F rcTransformedBrushRect = D2D1::RectF(100, 100, 200, 200);
```

```
m_pBitmapBrush->SetTransform( D2D1::Matrix3x2F::Translation(D2D1::SizeF(50,50)) );
```

```
m_pRenderTarget->FillRectangle( &rcTransformedBrushRect, m_pBitmapBrush );
```

```
m_pRenderTarget->DrawRectangle( &rcTransformedBrushRect, m_pBlackBrush, 1, NULL);
```



# 붓 변환하기"

03.TransformBrushesExample

- 예제
  - ID2D1Brush::SetTransform 를 사용하여 붓을 변환하는 예

