

Window Programming

Visual C++ MFC Programming

Lecture 05

김예진

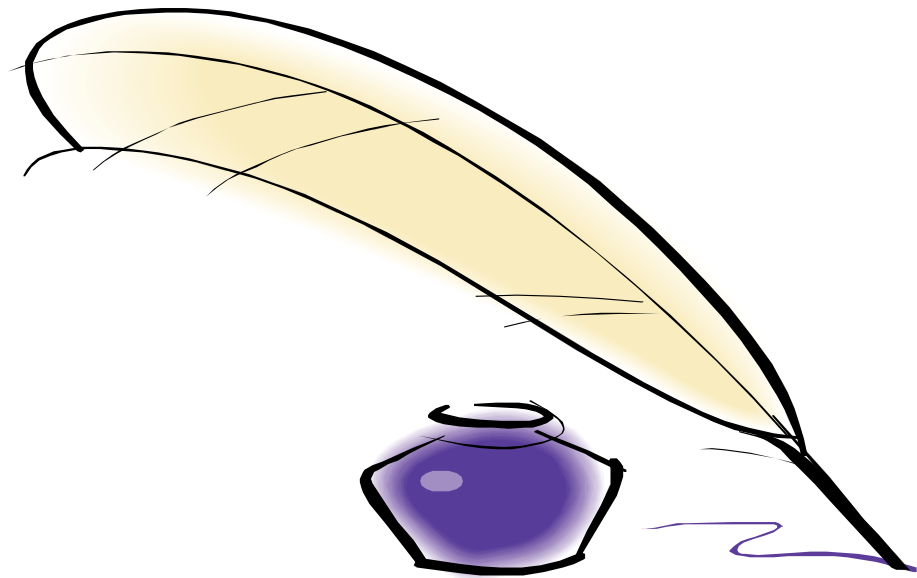
Dept. of Game Software

Notices

- 03/20: HW 1 (Due: 03/26)

Plan

- MFC로 그림 그리기
 - Point
 - 연습 1: Color 점 찍기
 - 연습 2: Color 점 다수 찍기
 - Rectangle & ellipse
 - 연습 3: CRect 사용
 - Line
 - 연습 4: Line 그리기
 - Text
 - 연습 5: Text 출력
 - Mapping mode
 - 연습 6: 좌표 변환



Review: 다양한 디바이스 컨텍스트 클래스

클래스 이름	용도
CPaintDC	클라이언트 영역에 출력할 때 (WM_PAINT 메시지 핸들러에서만 사용)
CClientDC	클라이언트 영역에 출력할 때 (WM_PAINT 메시지 핸들러를 제외한 다른 모든 곳에서 사용)
CWindowDC	윈도우의 전체 영역(클라이언트 영역 + 비 클라이언트 영역)에 출력할 때
CMetaFileDC	메타 파일(Metafile)에 출력할 때

CMetaFileDC 클래스

- 메타 파일(Metafile)
 - GDI 명령어를 저장할 수 있는 파일
 - 여러 개의 저장한 GDI 명령어를 한꺼번에 수행하여 그림을 그릴 수 있음
- CMetaFileDC 사용 방법
 - CMetaFileDC 객체를 만든 후 CMetaFileDC::Create() 호출
 - 메타 파일 객체를 일반적인 디바이스 컨텍스트 객체로 간주하고 출력 관련 멤버 함수를 호출
 - CMetaFileDC::Close()를 호출하면 윈도우 운영체제가 내부적으로 메타 파일을 만든 후 메타 파일 핸들(HMETAFILE 타입)을 리턴
 - CDC::PlayMetaFile()로 메타 파일을 실행

그리기 함수 (1/3)

- 점 찍기

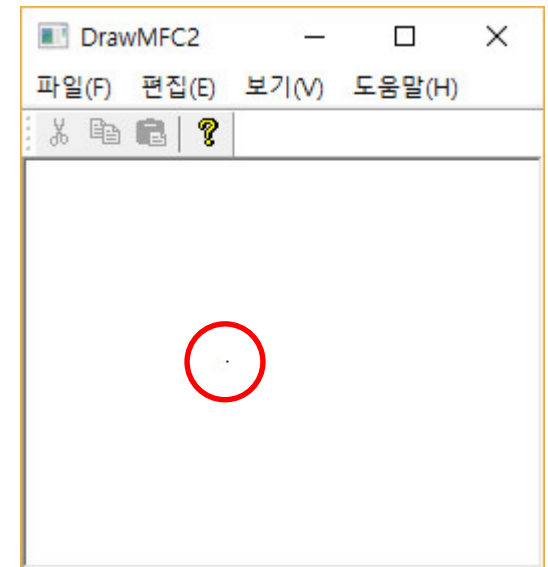
이름	기능
GetPixel	화면의 특정 위치에 해당하는 점의 색을 얻는다.
SetPixel	화면의 특정 위치에 원하는 색의 점을 찍으며 원래의 점의 색을 리턴한다.
SetPixelV	SetPixel과 기능은 동일하지만 SetPixel 함수와 달리 원래의 점의 색을 리턴하지 않으므로 속도가 더 빠르다.

```
COLORREF color = dc.GetPixel (x,y);  
dc.SetPixelV(x, y, RGB(r, g, b));
```

연습 1: Color 점 찍기

- (100, 100) 위치에 blue 점 찍기
 - SetPixel() 또는 SetPixelV() 사용
 - blue 값(COLORREF 타입) = 0x000000FF 사용

```
// (100,100)위치에 파란색을 칠함  
dc.SetPixel(100, 100, 0x000000FF);  
// or  
dc.SetPixelV(100, 100, 0x000000FF);  
  
// (100,100)위치의 색을 가져온다.  
COLORREF color = dc.GetPixel(100, 100);
```



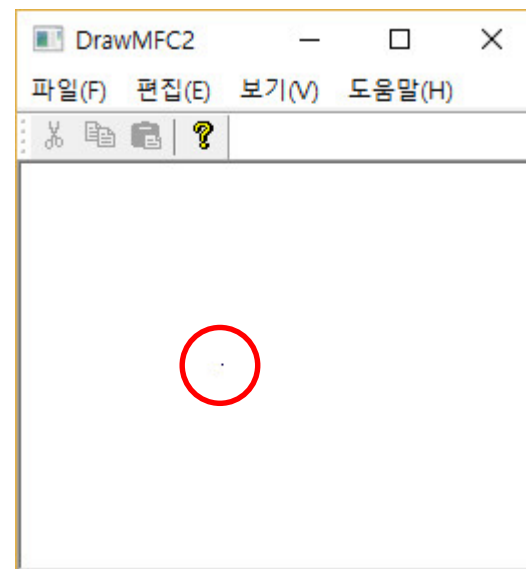
COLORREF

- 색을 저장하는 변수
- 32 bit: 0x00rrggbb 형식
- 여러가지 편한 매크로와 같이 사용:

```
// RGB 매크로를 사용하여 COLORREF color 값으로 변환  
COLORREF color = RGB(r, g, b);
```

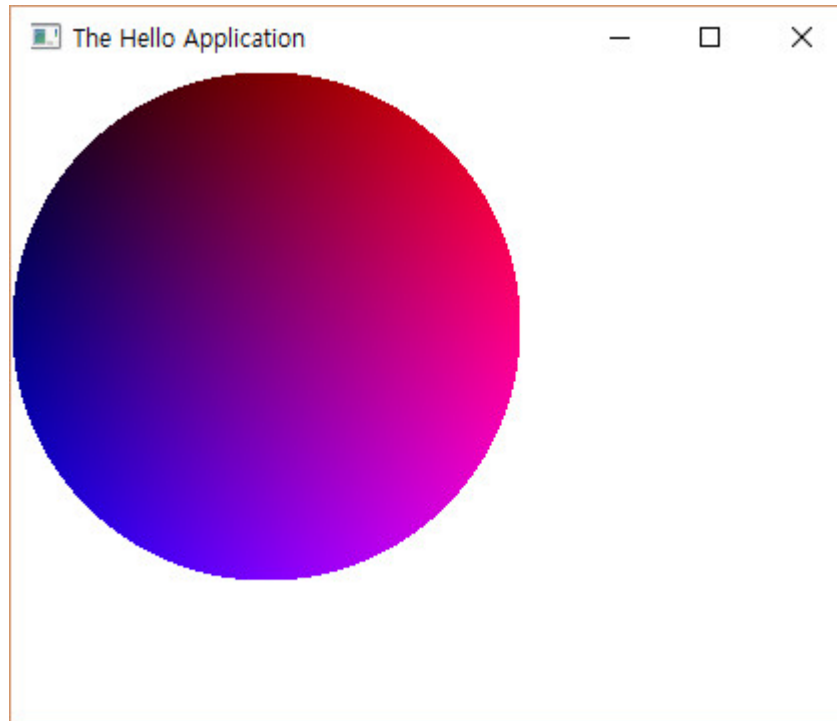
```
// 변수 r, g, b에 가져온 색의 R값, G값, B값을 저장  
int r = GetRValue(color);  
int g = GetGValue(color);  
int b = GetBValue(color);
```

```
// (100,100)위치에 파란색을 칠함  
dc.SetPixel(100, 100, RGB(0, 0, 100));
```



연습 2: Color 점 다수 찍기

- 서서히 색이 변하는 원 그리기
 - (x, y) 가 원의 내부 좌표일 때만 색을 칠함
 - 원의 좌표: $(x - r)(x - r) + (y - r)(y - r) < r * r$
 - $r = 255 / 2 = 127$



연습 2: Color 점 다수 찍기

```
// 서서히 색이 변하는 원 그리기
for (int x = 0 ; x < 255; x++) {
    for (int y = 0; y < 255; y++) {
        //(x,y)가 원의 내부 좌표일 때만 색을 칠한다
        if ((x - 127) * (x - 127) + (y - 127) * (y - 127) < 127 * 127)
            dc.SetPixelV(x, y, RGB(x,0,y));
    }
}
```

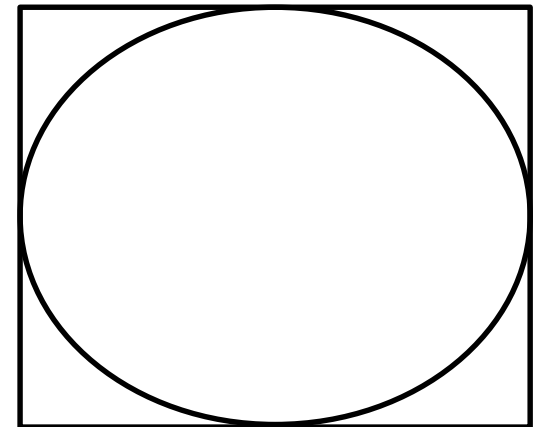
DC의 그리기 관련 멤버 함수

- 도형 그리기

이름	기능
Rectangle()	사각형을 그린다.
Ellipse()	사각형에 내접하는 타원을 그린다.

```
dc.Rectangle (x1, y1, x2, y2);  
dc.Ellipse (x1, y1, x2, y2);
```

(x1, y1)

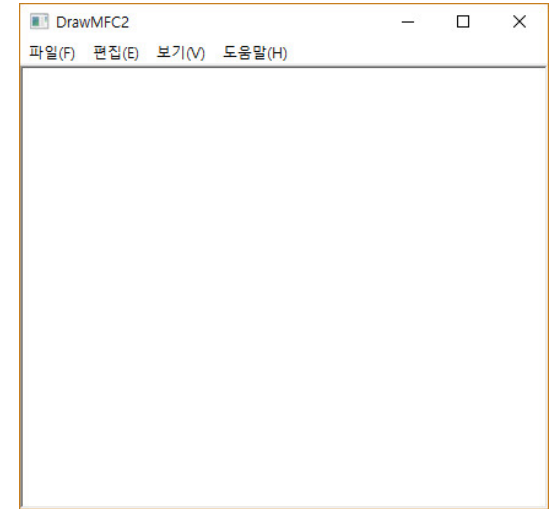


(x2, y2)
11

void GetClientRect(CRect)

- 창의 사이즈를 알고 싶다?

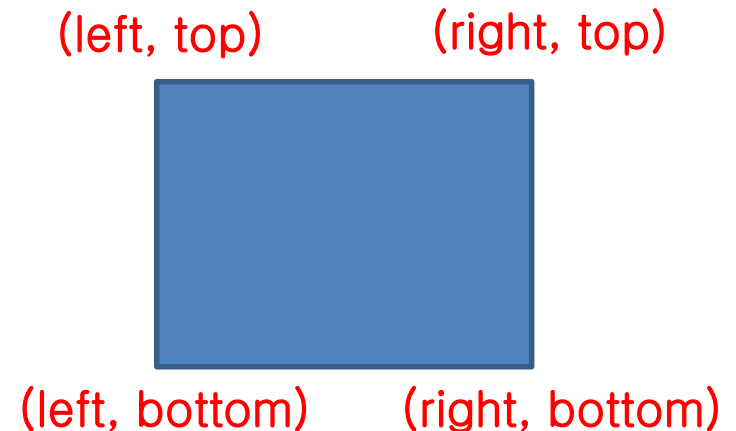
```
CRect rect;  
GetClientRect(rect);
```



- CRect : 사각형 정보를 저장하는 클래스

– member variables:

- bottom
- top
- right
- left



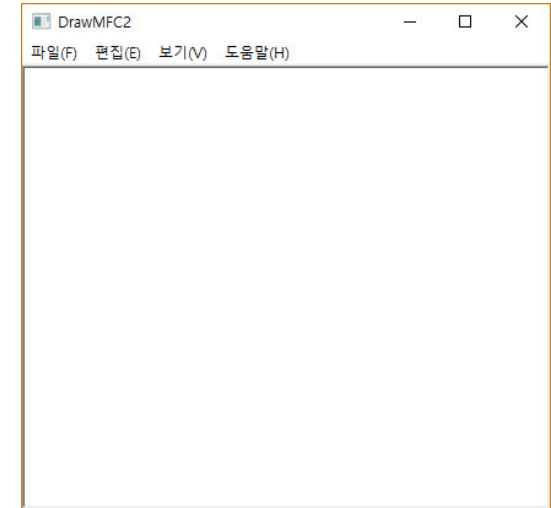
교재 page 78~83 읽기

[http://msdn2.microsoft.com/ko-kr/library/zzs00fs6\(VS.80\).aspx](http://msdn2.microsoft.com/ko-kr/library/zzs00fs6(VS.80).aspx)

연습 3: CRect 사용

- 창의 사이즈를 알고 싶다?

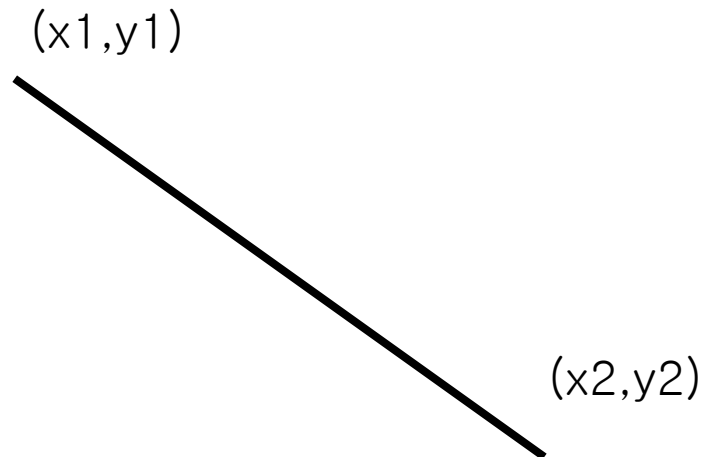
```
CRect rect;  
GetClientRect(rect);  
dc.Ellipse(rect);
```



그리기 함수 (3/3)

이름	기능
MoveTo()	현재 위치를 옮긴다.
LineTo()	현재 위치로부터 특정 위치까지 선을 그린 후 현재 위치를 갱신한다.

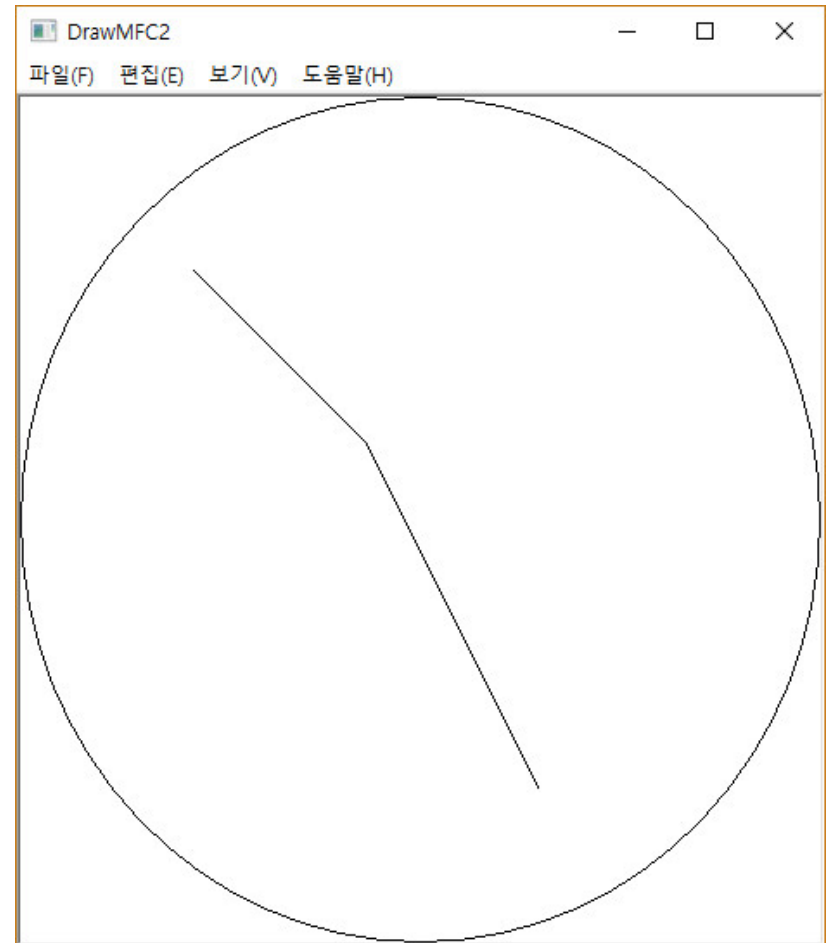
```
dc.MoveTo(x1,y1);  
dc.LineTo(x2,y2);
```



연습 4: 라인 그리기

- (100, 100), (200, 200), (300, 400) 포인트를 경유하는 라인을 그려라.

```
dc.MoveTo(100, 100);  
dc.LineTo(200, 200);  
dc.LineTo(300, 400);
```



텍스트 함수

- 텍스트 출력 함수

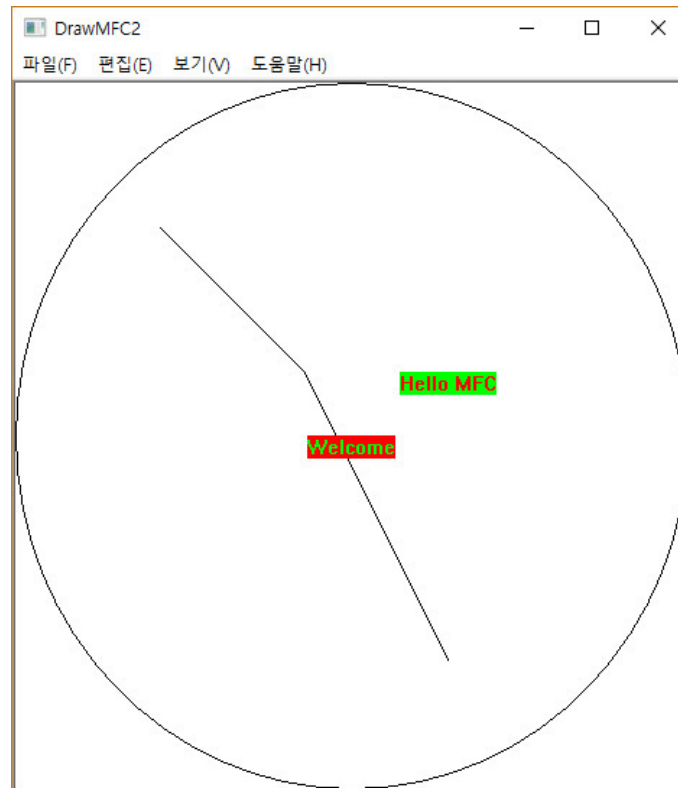
이름	기능
TextOut()	특정 위치에 문자열을 출력한다.
DrawText()	사각형을 기준으로 문자열을 출력한다.
SetTextColor()	문자의 색을 바꾼다.
SetBkColor()	문자의 배경색을 바꾼다.
SetTextAlign()	기준 위치에 대한 문자열의 정렬 방식을 정한다.

```
dc.SetTextColor(RGB(255, 0, 0));  
dc.SetBkColor(RGB(0, 255, 0));  
dc.SetTextAlign(TA_CENTER);  
dc.TextOut(300, 200, _T("Hello MFC"));
```

[http://msdn2.microsoft.com/ko-kr/library/e37h9k5s\(VS.80\).aspx](http://msdn2.microsoft.com/ko-kr/library/e37h9k5s(VS.80).aspx)

연습 5: 텍스트 출력

- 클라이언트 영역 중간에 다음과 같은 텍스트 출력
 - CRect 사용



연습 4: 텍스트 출력

```
CRect rect;  
GetClientRect(rect);  
dc.SetTextColor(RED);  
dc.SetBkColor(RED);  
  
// dc.SetBkMode(TRANSPARENT);  
dc.SetTextAlign(TA_CENTER);  
dc.TextOut(rect.Width()/2, rect.Height()/2, _T("Welcome"));  
  
// or  
dc.DrawText(_T("Welcome"), &rect, DT_CENTER | DT_VCENTER | DT_SINGLELINE);
```

출력할 문자열

글자의 출력 맞춤 모양

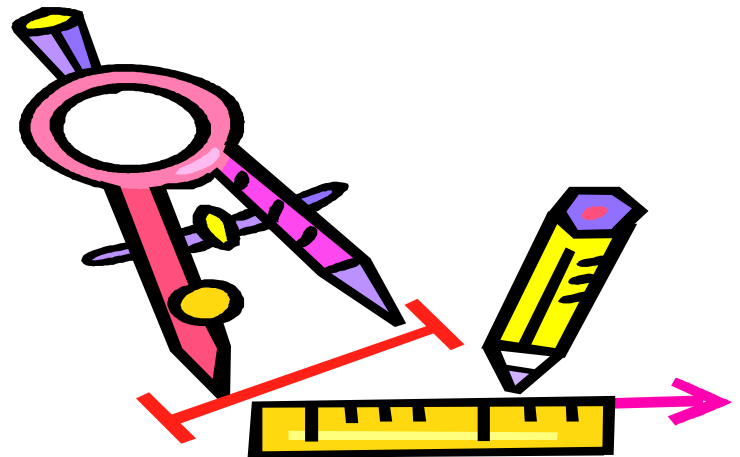
문장을 출력할 범위를 의미하는 사각형

속성 함수

속성	초기값	속성을 얻는 함수	속성을 변경하는 함수
텍스트 색상	검정색	GetTextColor()	SetTextColor()
배경 색상	흰색	GetBkColor()	SetBkColor()
배경 모드	OPAQUE	GetBkMode()	SetBkMode()
매핑 모드	MM_TEXT	GetMapMode()	SetMapMode()
그리기 모드	R2_COPYPEN	GetROP2()	SetROP2()
현재 위치	(0, 0)	GetCurrentPosition()	MoveTo()
펜	BLACK_PEN	SelectObject()	SelectObject()
브러시	WHITE_BRUSH	SelectObject()	SelectObject()
폰트	SYSTEM_FONT	SelectObject()	SelectObject()
비트맵	없음	SelectObject()	SelectObject()
팔레트	없음	SelectPalette()	SelectPalette()
영역	없음	SelectObject()	SelectObject()

2가지 좌표계

- 화면상에 5cm 길이의 라인을 긋고 싶다면?
- Windows의 2가지 길이 표시 방법:
 - Device Unit: Pixel
 - Logical Unit: Real Metric (mm, cm, inch, ...)



매핑 모드 변환

CDC::SetMapMode (MappingMode);

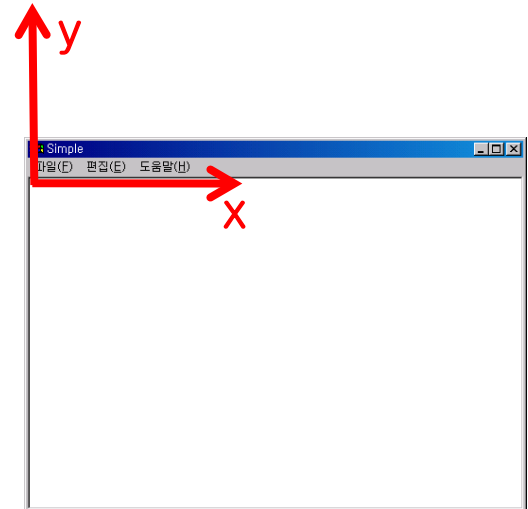
매핑 모드	단위	x축	y축
MM_TEXT	1 픽셀	→ +x	↓ +y
MM_LOMETRIC	0.1 mm	→ +x	↓ -y
MM_HIMETRIC	0.01 mm	→ +x	↓ -y
MM_LOENGLISH	0.01 인치	→ +x	↓ -y
MM_HIENGLISH	0.001 인치	→ +x	↓ -y
MM_TWIPS	1/1440 인치	→ +x	↓ -y
MM_ISOTROPIC	사용자 정의(가로, 세로 길이 동일)	사용자 정의	사용자 정의
MM_ANISOTROPIC	사용자 정의	사용자 정의	사용자 정의

Mapping Mode

- 5cmX5cm 크기의 사각형을 그리자!

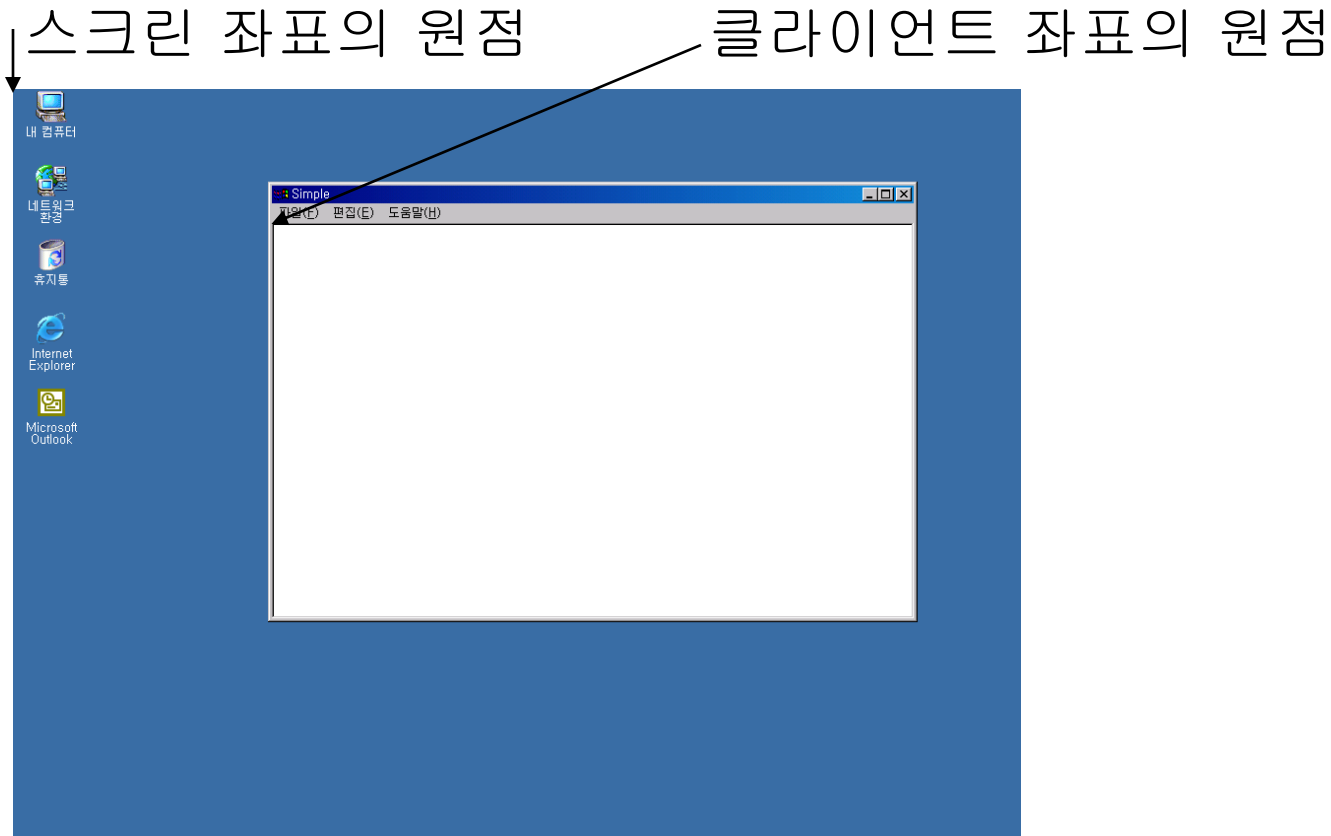
매핑 모드	단위	x축	y축
MM_LOMETRIC	0.1 mm	→ +x	↓ -y

```
dc.SetMapMode(MM_LOMETRIC);  
dc.Rectangle(0, 0, 500, -500);
```



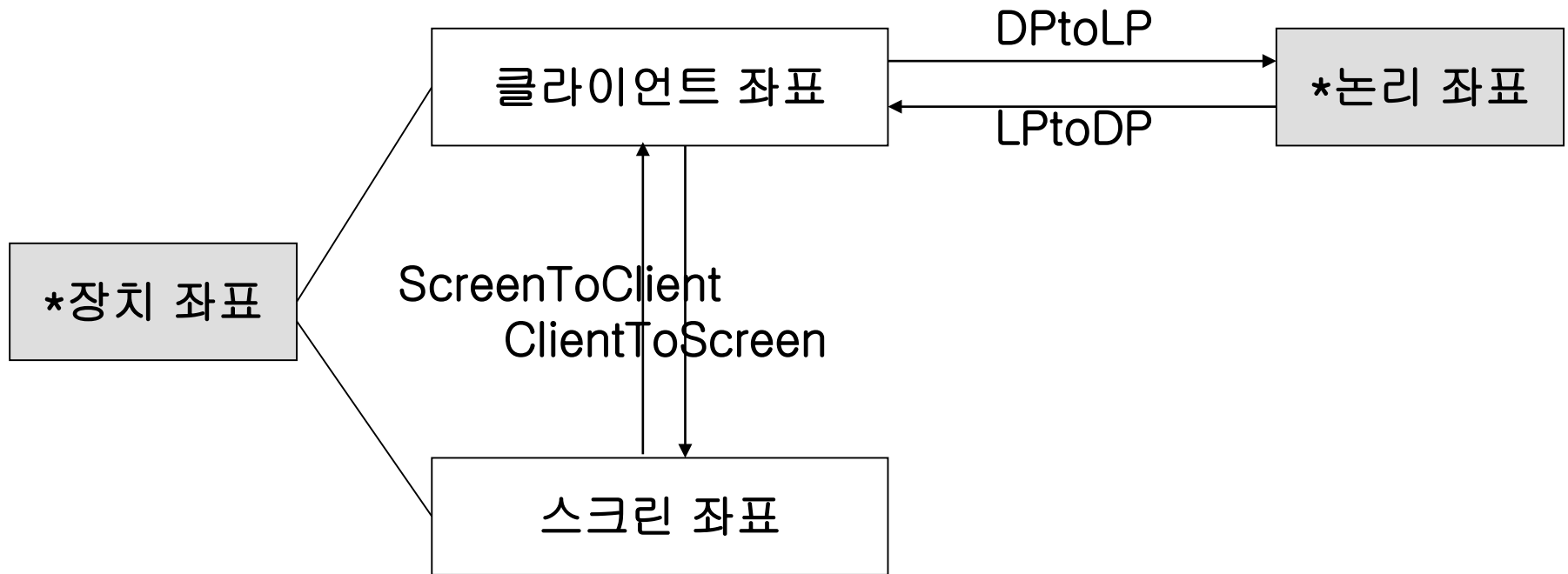
좌표 변환 (1/2)

- 스크린 좌표와 클라이언트 좌표



좌표 변환 (2/2)

- 좌표 변환 함수



*논리 좌표: 프로그래머가 사용하는 단위

*장치 좌표: 운영체제 내부적으로 사용하는 단위 (픽셀)

연습 6: 좌표 변환

// 1번 사각형

```
dc.Rectangle(0, 0, 250, 250);
```

// 2번 사각형

```
dc.SetMapMode(MM_LOMETRIC);
```

```
dc.Rectangle(0, 0, 500, -500);
```

// 3번 사각형

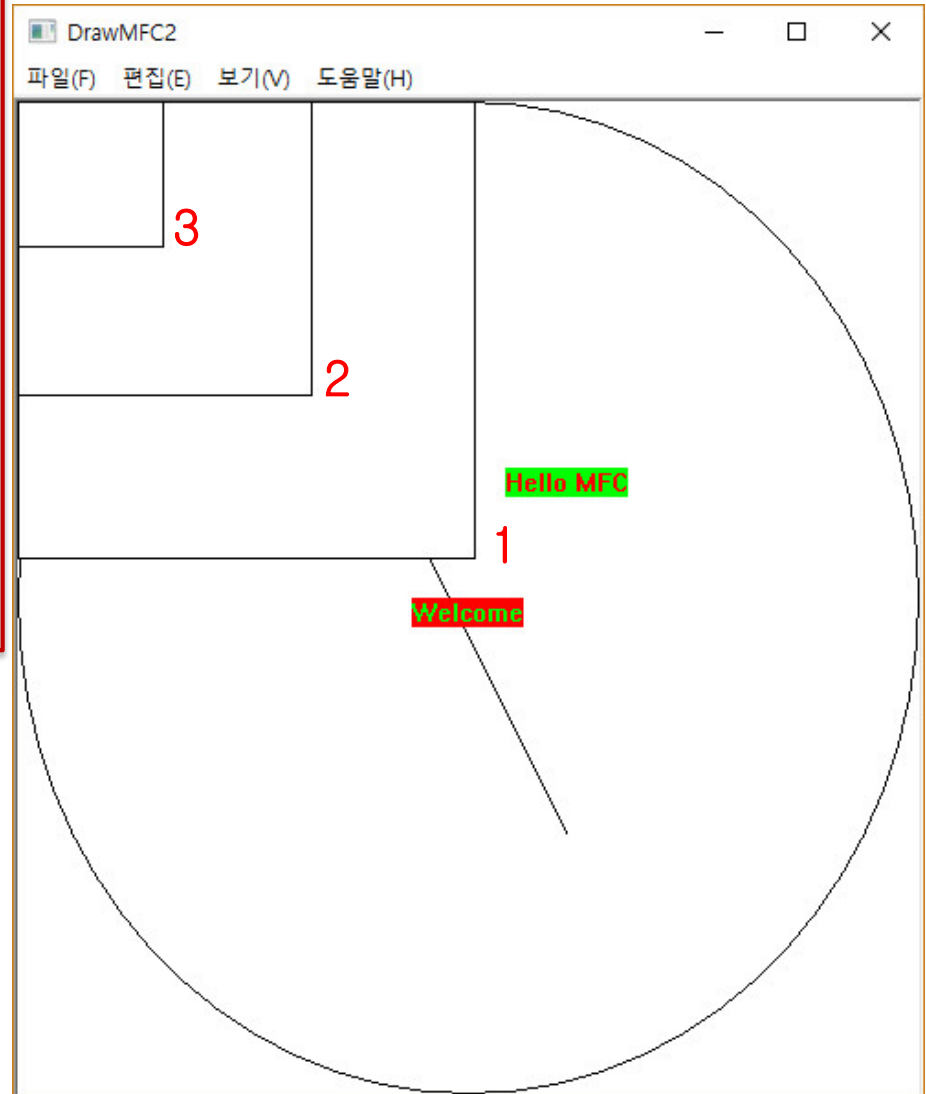
```
POINT p;
```

```
p.x = 500; p.y = -500;
```

```
dc.LPtoDP(&p);
```

```
dc.SetMapMode(MM_TEXT);
```

```
dc.Rectangle(0, 0, p.x/2, p.y/2);
```



Q & A