// 숫자값 화면에 띄우는 방법

wstring count = L"" + to\_wstring(m\_Points.size());

TextOut(dc, 100, 100, count.c\_str(), count.size());

// (x,y) 좌표의 픽셀 칠하기

dc.SetPixelV(x, y, RGB(x, y, 0));

\*if문을 이용하여 원하는 픽셀에 색칠

// 선그리는 함수

dc.MoveTo(x1, y1); dc.Lineto(x2, y2);

// 팬 생성 방법

CPen pen; pen.CreatePen(PS\_SOLID, 2, RGB(255, 0, 0));

\*종류 PS\_SOLID, PS\_DASH, PS\_DOT, PS\_DASHDOT, PS\_DASHDOTDOT, PS\_NULL, PS\_INSIDEFRAME

// 솔리드 브러시

CBrush brush1(RGB(255, 0, 0)); dc.SelectObject(&brush1); dc.Rectangle(50, 50, 200, 200);

// 해치 브러시

CBrush brush2(HS\_DIAGCROSS, RGB(255, 0, 0)); dc.SelectObject(&brush2); dc.Ellipse(250, 50, 400, 200);

// 패턴 브러시

CBitmap bitmap; bitmap.LoadBitmap(IDB\_BITMAP1);

CBrush brush3(&bitmap); dc.SelectObject(&brush3); dc.RoundRect(450, 50, 600, 200, 50, 50);

// 폰트 생성

CPaintDC dc(this); CFont font; font.CreatePointFont(400, \_T("Arial"));

dc.SelectObject(&font);. dc.TextOut(100, 100, CString("Hello"));

// 오각형 그리기

CRgn rgn; CPoint ptVertex[5];

rgn.CreatePolygonRgn(ptVertex, 5, ALTERNATE);

dc.SelectObject(&brush); dc.PaintRgn(&rgn);

// 더블 버퍼 & 비트맵 반복 X

CDC memDC;

memDC.CreateCompatibleDC(&dc);

CRect rect;

GetClientRect(rect);

CBitmap \_bitmap;

\_bitmap.LoadBitmap(IDB\_BITMAP1);

BITMAP bmpinfo;

\_bitmap.GetBitmap(&bmpinfo);

CDC dcmem;

dcmem.CreateCompatibleDC(&memDC);

dcmem.SelectObject(&\_bitmap);

CBitmap bitmap;

bitmap.CreateCompatibleBitmap(&dc, rect.Width(), rect.Height());

memDC.SelectObject(&bitmap);

memDC.SelectObject(brush);

memDC.StretchBlt(

(rect.Width() / 2) - (rect.Height() \* 0.5 / 2)

,0,

rect.Height() \* 0.5,

rect.Height(),

&dcmem,

0,

0,bmpinfo.bmWidth,

bmpinfo.bmHeight,

SRCCOPY);

dc.BitBlt(0, 0, rect.Width(), rect.Height(), &memDC, 0, 0, SRCCOPY);

erase 추가

// 클라이언트 영역 밖에서 마우스 위치 확인

SetCapture(); - 누를때

:ReleaseCapture(); 땔때

// 타이머 세팅

Create 함수 생성 후 OnTimer 생성

void SetTimer(int id, int time, void \*fp); 설정

\*VK

0x41 A , 0x57 W, 0x53 S, 0x44 D

// 문자 메시지

#include <afxtempl.h>

CArray <TCHAR,TCHAR> m\_str;

void CChildView::OnPaint() {

CPaintDC dc(this);

CFont font;

font.CreatePointFont(150, \_T("궁서"));

dc.SelectObject(&font);

CRect rect;

GetClientRect(&rect);

dc.DrawText(m\_str.GetData(),m\_str.GetSize(),&rect,DT\_LEFT); }

void CChildView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)

{

if (nChar == \_T('\b')) {

if(m\_str.GetSize() > 0) m\_str.RemoveAt(m\_str.GetSize()-1); }

else{ m\_str.Add(nChar); }

Invalidate(); }

// 키 입력 예시

void CChildView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)

{

switch (nChar)

{

case VK\_LEFT: m\_xPos -= 20; break;

case VK\_RIGHT: m\_xPos += 20; break;

case VK\_UP: m\_yPos -= 20; break;

case VK\_DOWN: m\_yPos += 20; break;

case VK\_SPACE: m\_bFill = !m\_bFill;

}

// 마우스 커스 위치 추적 예제

BOOL m\_bMouseIn;

void CChildView::OnMouseMove(UINT nFlags, CPoint point)

{

if (m\_bMouseIn == FALSE) {

// 마우스 커서 추적을 요청한다.

TRACKMOUSEEVENT tme;

tme.cbSize = sizeof(tme);

tme.dwFlags = TME\_LEAVE;

tme.hwndTrack = this->m\_hWnd;

tme.dwHoverTime = HOVER\_DEFAULT;

::TrackMouseEvent(&tme);

// 메인 윈도우 크기를 300\*150으로 변경한다.

CWnd\* pMainWnd = AfxGetMainWnd();

CRect rect; pMainWnd->GetWindowRect(&rect);

rect.right = rect.left + 300;

rect.bottom = rect.top + 150;

pMainWnd->MoveWindow(&rect);

// 마우스 커서가 클라이언트 영역에 있음을 기억해둔다. m\_bMouseIn = TRUE; }

}

void CChildView::OnMouseLeave() {

// 마우스 커서가 클라이언트 영역 밖에 있음을 기억해둔다. m\_bMouseIn = FALSE;

// 메인 윈도우 크기를 200\*100으로 변경한다.

CWnd\* pMainWnd = AfxGetMainWnd();

CRect rect; pMainWnd->GetWindowRect(&rect);

rect.right = rect.left + 200;

rect.bottom = rect.top + 100;

pMainWnd->MoveWindow(&rect); }