CImage 사용법

한국산업기술대학교
게임공학부
김도율

CImage란?

- ATL에서 추가된 이미지 관리 클래스
  - ATL과 MFC에 대한 자세한 설명은 https://msdn.microsoft.com/ko-kr/library/hh967573.aspx 참고
- Visual Studio Community 이상 버전에서만 지원
  - Visual Studio Express for Windows 미지원

- MSDN 설명
  - CImage는 JPEG, GIF, BMP 및 PNG (Portable Network Graphics) 형식으로 이미지를 로드하고 저장할 수 있는 기능을 포함하여 **향상된 비트맵을 지원**합니다.

- 자세한 설명은
  - https://docs.microsoft.com/en-us/cpp/atl-mfc-shared/reference/cimage-class (영문)
  - https://msdn.microsoft.com/ko-kr/library/bwea7by5(v=vs.120).aspx (한글)

CImage 개요

1. 일종의 비트맵 관리 클래스
   - 비트맵 정보를 내부에서 보유
2. 소멸자에서 비트맵 메모리 해제
3. 파일 시스템에서 비트맵 로드 가능
   - 리소스에서도 로드 가능
4. 여러 종류의 이미지 포맷 지원
   - JPEG
   - GIF
   - BMP
   - PNG

CImage의 사용방법

- #include <atlImage.h> 사용
- Load 멤버 함수를 사용해 로드 가능
- GDI의 그리기 함수 지원
  - BitBlt
  - StretchBlt
  - TransparentBlt
  - AlphaBlend
- 알아서 잘 그리게 하려면 Draw 함수 사용
- Destroy() 함수를 사용하여 명시적 해제 가능

```
#include <atlimage.h>   // CImage 클래스가 정의된 헤더
void OnDraw(HWND hWnd)
{
    CImage img;
    img.Load(TEXT("Bitmap01.png"));
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    {
        HDC memDC = CreateCompatibleDC(hdc);
        HBITMAP memBit = CreateCompatibleBitmap(hdc, CLIENT_WIDTH, CLIENT_HEIGHT);
        SelectObject(memDC, memBit);
        // 알파 값(투명 값)을 지원하는 이미지의 경우, Draw를 사용하여 적용 가능
        // HDC, x, y, width, height
        img.Draw(memDC, 100, 100, 200, 200);

        BitBlt(hdc, 0, 0, CLIENT_WIDTH, CLIENT_HEIGHT, memDC, 0, 0, SRCCOPY);
        DeleteObject(memBit);
        DeleteDC(memDC);
    }
    EndPaint(hWnd, &ps);
    img.Destroy();  // img 객체를 명시적으로 파괴할 때 사용. 생략해도 무관.
}
```

CImage의 강점 ~ 그리기

- Draw 함수를 사용하여 PNG 등의 투명 값을 적용하여 그리기 가능
- 그리기 함수의 매개인자를 사용하기 편하도록 매우 많이 오버로드
  - x Dest, y Dest, Dest Width, Dest Height, x Src, y Src, Src Width, Src Height
  - rc Dest, rc Src
  - rc Dest
  - x Dest, y Dest, Dest Width, Dest Height
  - x Dest, y Dest
  - etc.

- 더 자세히 알고 싶으면 atlImage.h 헤더에 함수 정의가 있으므로 참고

CImage의 주의사항

- Load()의 인자로 준 위치에 이미지 파일이 없거나 읽을 수 없으면 E_FAIL 반환
  - **런타임 에러를 발생시키지 않음 (= 죽지 않음)**
- IsNull() 멤버 함수를 사용하여 비트맵이 들어있는지 확인 가능
  - **Load 실패 시 IsNull()의 반환 값은 true**

- 하나의 CImage 객체를 재사용하기 위해서는 Destroy()로 내부의 Bitmap 해제 필요

- 이미지를 불러들이고 해제하는 것은 매우 비싼(시간이 많이 드는) 연산이므로 Create 때 수행

CImage 잘 써 보기(심화)

- CImage로 더블버퍼링 하기

```
#include <atlimage.h>   // CImage 클래스가 정의된 헤더
void OnDraw(HWND hWnd)
{
    CImage img;
    // BPP(Bits Per Pixel. 24 : RGB, 32 : RGBA
    img.Create(CLIENT_WIDTH, CLIENT_HEIGHT, 24);
    PAINTSTRUCT ps;
    HDC hdc = BeginPaint(hWnd, &ps);
    {
        HDC memDC = img.GetDC();

        // TODO: 그리기

        img.Draw(hdc, 0, 0);
        img.ReleaseDC();
    }
    EndPaint(hWnd, &ps);
}
```

- 리소스의 이미지 읽기
- MAKEINTRESOURCE 생략해도 무관
- png 이미지는 불러오기 문제 있음

```
CImage img;
// 둘 다 OK
img.LoadFromResource(hInst, MAKEINTRESOURCE(IDB_BITMAP1));
img.LoadFromResource(hInst, IDB_BITMAP1);
```