



WINAPI -CHAPTER 6-

SOULSEEK



목차

1. BitmapClass

2. BitmapManager

BITMAPCLASS

```
#include<windows.h>
```

```
class BitMap
```

```
{
```

```
private:
```

```
HDC      MemDC;
```

```
HBITMAP  m_pBitMap;
```

```
HBITMAP  m_pBitOld;
```

```
SIZE     m_size;
```

```
public:
```

```
BitMap();
```

```
void Init(HDC hdc, HINSTANCE hInst, int id);
```

```
void Draw(HDC hdc, int x, int y, int spX = 1, int spY = 1);
```

```
~BitMap();
```

```
};
```

BITMAPCLASS

```
#include "BitMap.h"
```

```
BitMap::BitMap()
```

```
{  
}
```

```
//메모리 DC를 생성하고 그림 그릴 준비를 한다.
```

```
Void BitMap::Init(HDC hdc, HINSTANCE hInst, int id)
```

```
{
```

```
    MemDC = CreateCompatibleDC(hdc);
```

```
    M_pBitMap = (HBITMAP)LoadBitmap(hInst, MAKEINTRESOURCE(id));
```

```
    M_pBitOld = (HBITMAP)SelectObject(MemDC, m_pBitMap);
```

```
    BITMAP BitMap_Info;
```

```
    //준비된 비트맵의 정보를 가져온다. - width, height
```

```
    GetObject(m_pBitMap, sizeof(BitMap_Info), &BitMap_Info);
```

```
    m_size.cx = BitMap_Info.bmWidth;
```

```
    m_size.cy = BitMap_Info.bmHeight;
```

```
}
```

```
void BitMap::Draw(HDC hdc, int x, int y)
```

```
{
```

```
    //그려준다.
```

```
    StretchBlt(hdc, x, y, m_size.cx * spX, m_size.cy * spY, MemDC, 0, 0, m_size.cx, m_size.cy, SRCCOPY);
```

```
}
```

```
BitMap::~~BitMap()
```

```
{
```

```
    //해제코드를 추가한다.
```

```
    SelectObject(MemDC, m_pBitOld);
```

```
    DeleteObject(m_pBitMap);
```

```
    DeleteDC(MemDC);
```

```
}
```

MANAGER CLASS

- **Class**는 재료 or 도구, **Manager** 관리
- **Manager**는 계속 지속적으로 연산을 하거나 갱신되는 데이터를 적용해야 할 객체를 구분, 선별해서 명령을 내려 주는 단위, **Class**는 갱신, 또는 적용되는 데이터를 직접적으로 변경되고 표현되는 단위
 ex) 역할을 하는 배우는 **Class**, “야! 이번 장면은 이렇게 해야 해!” 는 **Manager**
- 게임에서 컨텐츠나 장면 별로 관리하는 **Manager**와 그것에 영향 받는 **Class**로 구성된다.

MANAGER CLASS

```
#pragma once
#include<Windows.h>
#include"Bitmap.h"
class Gamemanager
{
private:
    static Gamemanager* m_pThis;
    Bitmap* m_bt;

public:
    static Gamemanager* GetInstans()
    {
        if (m_pThis == NULL)
        {
            m_pThis = new Gamemanager;
        }
        return m_pThis;
    }

    Gamemanager();

    void Init(HWND hWnd, HINSTANCE hInst);
    void Draw(HDC hdc, int x, int y);
    void Release();

    ~Gamemanager();
};
```

MANAGER CLASS

```
#include "Gamemanager.h"
```

```
Gamemanager* Gamemanager::m_pThis = NULL;
```

```
Gamemanager::Gamemanager()
```

```
{  
}
```

```
void Gamemanager::Init(HWND hWnd, HINSTANCE hInst)
```

```
{
```

```
    HDC hdc = GetDC(hWnd);
```

```
    m_bt = new Bitmap;
```

```
    m_bt->Init(hdc, hInst);
```

```
}
```

```
void Gamemanager::Draw(HDC hdc, int x, int y)
```

```
{
```

```
    m_bt->Draw(hdc, x, y);
```

```
}
```

```
void Gamemanager::Release()
```

```
{
```

```
    delete(m_pThis);
```

```
}
```

```
Gamemanager::~Gamemanager()
```

```
{
```

```
    delete(m_bt);
```

```
}
```

MANAGER CLASS

1. 짝 맞추기 게임을 만들어보자.