

11. 싱글톤 매크로

Macro.h

```
#pragma once

#define SAFE_DELETE(p) if(p) { delete p; p = NULL; }
#define SAFE_DELETE_ARRAY(p) if(p) { delete[] p; p = NULL; }

#define DECLARE_SINGLE(Type) \
private:\
    static Type* m_pInst;\
public:\
    static Type* GetInst()\
    {\
        if(!m_pInst)\
            m_pInst = new Type;\
        return m_pInst;\
    }\
    static void DestroyInst()\
    {\
        SAFE_DELETE(m_pInst);\
    }\
private:\
    Type();\
    ~Type();

#define DEFINITION_SINGLE(Type) Type* Type::m_pInst = nullptr;
#define GET_SINGLE(Type) Type::GetInst()
#define DESTROY_SINGLE(Type) Type::DestroyInst()
```

이렇게 아래 줄을 붙일수 있음.

SceneManager

```
#pragma once

#include "../Game.h"

class CSceneManager
{
public:
    bool Init();

    DECLARE_SINGLE( CSceneManager )
};
```

파일 경로 주의

특약

```
#include "SceneManager.h"

DEFINITION_SINGLE(CSceneManager)

CSceneManager::CSceneManager()
{
}

CSceneManager::~CSceneManager()
{
}

bool CSceneManager::Init()
{
    return true;
}
```

Core.cpp

```
#include "Core.h"
#include "Scene\SceneManager.h"

CCore* CCore::m_pInst;
bool CCore::m_bLoop = true;

CCore::CCore()
{
}

CCore::~CCore()
{
    DESTROY_SINGLE( CSceneManager );
}

bool CCore::Init( HINSTANCE hInst )
{
    m_hInst = hInst;

    // 윈도우 클래스 등록
    MyRegisterClass();

    // 해상도 설정
    m_tRs.iW = 1280;
    m_tRs.iH = 720;

    // 윈도우 창 생성
    Create();

    // 장면 관리자 초기화
    if( !GET_SINGLE( CSceneManager )->Init() )
    {
        return false;
    }

    return true;
}
```

core는 메인관리자.
서브관리자 해체

각종 서브관리자
초기화

소스 파일 경로 구성

..		
Scene	11. Singleton	28 minutes ago
APIFramework.rc	10. Basic Framework	20 hours ago
APIFramework.vcxproj	11. Singleton	28 minutes ago
APIFramework.vcxproj.filters	11. Singleton	28 minutes ago
Core.cpp	11. Singleton	28 minutes ago
Core.h	10. Basic Framework	20 hours ago
Game.h	10. Basic Framework	20 hours ago
Macro.h	11. Singleton	28 minutes ago
icon1.ico	10. Basic Framework	20 hours ago
main.cpp	10. Basic Framework	20 hours ago
resource.h	10. Basic Framework	20 hours ago
types.h	10. Basic Framework	20 hours ago

Core가 메인 관리자 역할을 하며, SceneManager는 장면을 관리하는 서브 관리자 역할을 한다.
관리가 편하도록 각 서브 관리자가 관리하는 클래스와 관리자 클래스들을 따로 담는 폴더를 만들어 관리한다.