19. 파일 경로 매니저 클래스

CPathManager.h

```h
#pragma once

#include "../Game.h"

class CPathManager
{
private:
	unordered_map<string, wstring> m_mapPath;

public:
	bool Init();
	bool CreatePath( const string& strKey, const wchar_t* pPath, const string& strBaseKey = ROOT_PATH );
	const wchar_t* FindPath( const string& strKey );

	DECLARE_SINGLE( CPathManager )
};
```

경로키값 / 실제경로값

Game.h

```h
#pragma once

#include <Windows.h>
#include <list>
#include <vector>
#include <unordered_map>
#include <crtdbg.h>

using namespace std;

#include "resource.h"
#include "Macro.h"
#include "Types.h"
#include "Flag.h"

#define PI 3.141592f

// °æ·Î Å°
#define ROOT_PATH		"RootPath"
#define TEXTURE_PATH	"TexturePath"
```

CPathManager.cpp

```cpp
#include "PathManager.h"

DEFINITION_SINGLE( CPathManager )

CPathManager::CPathManager()
{
}

CPathManager::~CPathManager()
{
}

bool CPathManager::Init()
{
	wchar_t strPath[MAX_PATH] = {};

	GetModuleFileName( NULL, strPath, MAX_PATH );

	for( int i = lstrlen( strPath ) - 1; i >= 0; --i )
	{
		if( strPath[i] == '/' || strPath[i] == '\\' )
		{
			memset( strPath + (i + 1), 0, sizeof( wchar_t ) * (MAX_PATH - (i + 1)) );
			break;
		}
	}

	m_mapPath.insert( make_pair( ROOT_PATH, strPath ) );

	// Texture 폴더 경로 설정
	if( !CreatePath( TEXTURE_PATH, L"Texture\\" ) )
	{
		return false;
	}

	return true;
}

bool CPathManager::CreatePath( const string& strKey, const wchar_t* pPath, const string& strBaseKey )
{
	const wchar_t* pBasePath = FindPath( strBaseKey );

	wstring strPath;

	if( pBasePath )
		strPath = pBasePath;

	strPath += pPath; // 최종 경로 만들어주기

	m_mapPath.insert( make_pair( strKey, strPath ) );

	return true;
}
```

Win32에서 미리 정의된 숫자

GetModuleFileName 함수는 exe 실행 파일의 이름을 포함한 경로를 리턴하기 때문에, 실행 파일이 있는 폴더의 '경로'만을 얻기위해 맨 뒤의 실행 파일 이름 부분을 지워준다.

```cpp
const wchar_t* CPathManager::FindPath( const string& strKey )
{
	unordered_map<string, wstring>::iterator iter = m_mapPath.find( strKey );
	if( iter == m_mapPath.end() )
	{
		return nullptr;
	}
	return iter->second.c_str();
}
```

Core.cpp

```cpp
#include "Core.h"
#include "Scene\SceneManager.h"
#include "Core/Timer.h"
#include "Core/PathManager.h"

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

CCore::CCore()
{
    _CrtSetDbgFlag( _CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF );
    //_CrtSetBreakAlloc(234);
}

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

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();

    // 화면 DC 만들어주기
    m_hDC = GetDC( m_hWnd );

    // 타이머 초기화
    if( !GET_SINGLE( CTimer )->Init() )
    {
        return false;
    }

    // 경로 관리자 초기화
    if( !GET_SINGLE( CPathManager )->Init() )
    {
        return false;
    }
```

Texture 클래스

```cpp
#pragma once

#include "../Ref.h"

class CTexture : public CRef
{
private:
        friend class CResourcesManager;

private:
        CTexture();
        ~CTexture();
};
```

리소스 매니저를 통해서만 접근을 허용하도록 함.

ResourcesManager 클래스

```cpp
#pragma once

#include "../Game.h"

class CResourcesManager
{
private:
        unordered_map<string, class CTexture*> m_mapTexture;

public:
        class CTexture* LoadTexture( const string& strKey, const wchar_t* pFileName, const string&

        DECLARE_SINGLE( CResourcesManager )
};
```

```cpp
#include "ResourcesManager.h"
#include "Texture.h"

DEFINITION_SINGLE( CResourcesManager )

CResourcesManager::CResourcesManager()
{
}

CResourcesManager::~CResourcesManager()
{
        Safe_Release_Map( m_mapTexture );
}

CTexture* CResourcesManager::LoadTexture( const string& strKey, const wchar_t* pFileName, const string& s
{
        return nullptr;
}
```