[ CGameManager.h ]

#pragma once

class CSceneState;

#define g\_pGameManager CGameManager::GetInstance()

enum SceneType : int

{

eMainScene, eGameScene, ePauseScene, eEndScene

};

class CGameManager

{

private:

CSceneState\* m\_nowScene;

CGameManager() { }

public:

static CGameManager\* GetInstance();

virtual ~CGameManager();

void CheckSceneChange(SceneType input);

void CheckNullScene();

void Destroy();

};

[ CGameManager.cpp ]

#include "GameManager.h"

#include "SceneState.h"

#include "State.h"

CGameManager \* CGameManager::GetInstance()

{

static CGameManager instance;

return &instance;

}

CGameManager::~CGameManager() { }

void CGameManager::CheckSceneChange(SceneType input)

{

// >> 1

CheckNullScene();

m\_nowScene = CSceneState::SetScene(input);

m\_nowScene->PrintState();

// // >> 2

// switch (input)

// {

// case eMainScene:

// {

// CheckNullScene();

// m\_nowScene = new CMainState;

// m\_nowScene->SetNowScene();

// }

// break;

//

// case eGameScene:

// {

// CheckNullScene();

// m\_nowScene = new CGameState;

// m\_nowScene->SetNowScene();

// }

// break;

//

// case ePauseScene:

// {

// CheckNullScene();

// m\_nowScene = new CPauseState;

// m\_nowScene->SetNowScene();

// }

// break;

//

// case eEndScene:

// {

// CheckNullScene();

// m\_nowScene = new CEndState;

// m\_nowScene->SetNowScene();

// }

// break;

// }

}

void CGameManager::CheckNullScene()

{

if (m\_nowScene != NULL)

delete m\_nowScene;

}

void CGameManager::Destroy()

{

if (m\_nowScene)

delete m\_nowScene;

}

[ SceneState.h ]

#pragma once

#include "GameManager.h"

class CSceneState

{

protected:

CSceneState() { }

public:

static CSceneState\* GetInstance();

static CSceneState\* SetScene(SceneType type);

virtual ~CSceneState() { }

virtual void SetNowScene() { }

virtual void PrintState() { }};

[ SceneState.cpp ]

#include "SceneState.h"

#include "State.h"

CSceneState \* CSceneState::GetInstance()

{

static CSceneState instance;

return &instance;

}

CSceneState\* CSceneState::SetScene(SceneType type)

{

switch (type)

{

case eMainScene:

return new CMainState();

break;

case eGameScene:

return new CGameState();

break;

case ePauseScene:

return new CPauseState();

break;

case eEndScene:

return new CEndState();

break;

}

}

[ State.h ]

#pragma once

#include <iostream>

#include "SceneState.h"

using namespace std;

class CMainState : public CSceneState

{

public:

CMainState() { }

~CMainState() { }

void SetNowScene() override { PrintState(); }

void PrintState() override { cout << "Main Scene" << endl; }

};

class CGameState : public CSceneState

{

public:

CGameState() { }

~CGameState() { }

void SetNowScene() override { PrintState(); }

void PrintState() override { cout << "Game Scene" << endl; }

};

class CPauseState : public CSceneState

{

public:

CPauseState() { }

~CPauseState() { }

void SetNowScene() override { PrintState(); }

void PrintState() override { cout << "Pause Scene" << endl; }

};

class CEndState : public CSceneState

{

public:

CEndState() { }

~CEndState() { }

void SetNowScene() override { PrintState(); }

void PrintState() override { cout << "End Scene" << endl; }

};

[ main.cpp ]

#include <iostream>

#include <string>

#include <stdlib.h>

#include <crtdbg.h>

#include "GameManager.h"

using namespace std;

int main()

{

\_CrtSetDbgFlag(\_CRTDBG\_ALLOC\_MEM\_DF || \_CRTDBG\_LEAK\_CHECK\_DF);

// \_CrtSetBreakAlloc(147); // string

char input;

while (true)

{

cout << "1 : Main / 2 : Game / 3 : Pause / 4 : End" << endl;

cin >> input;

if (input == '1')

g\_pGameManager->CheckSceneChange(eMainScene);

if (input == '2')

g\_pGameManager->CheckSceneChange(eGameScene);

if (input == '3')

g\_pGameManager->CheckSceneChange(ePauseScene);

if (input == '4')

{

g\_pGameManager->CheckSceneChange(eEndScene);

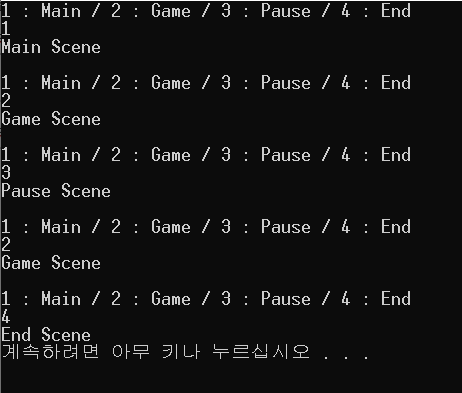
break;

}

cout << endl;

[ 실행화면 ]

}

 g\_pGameManager->Destroy();

\_CrtDumpMemoryLeaks();

return 0;

}