

Tsunagi

繋系 Engine

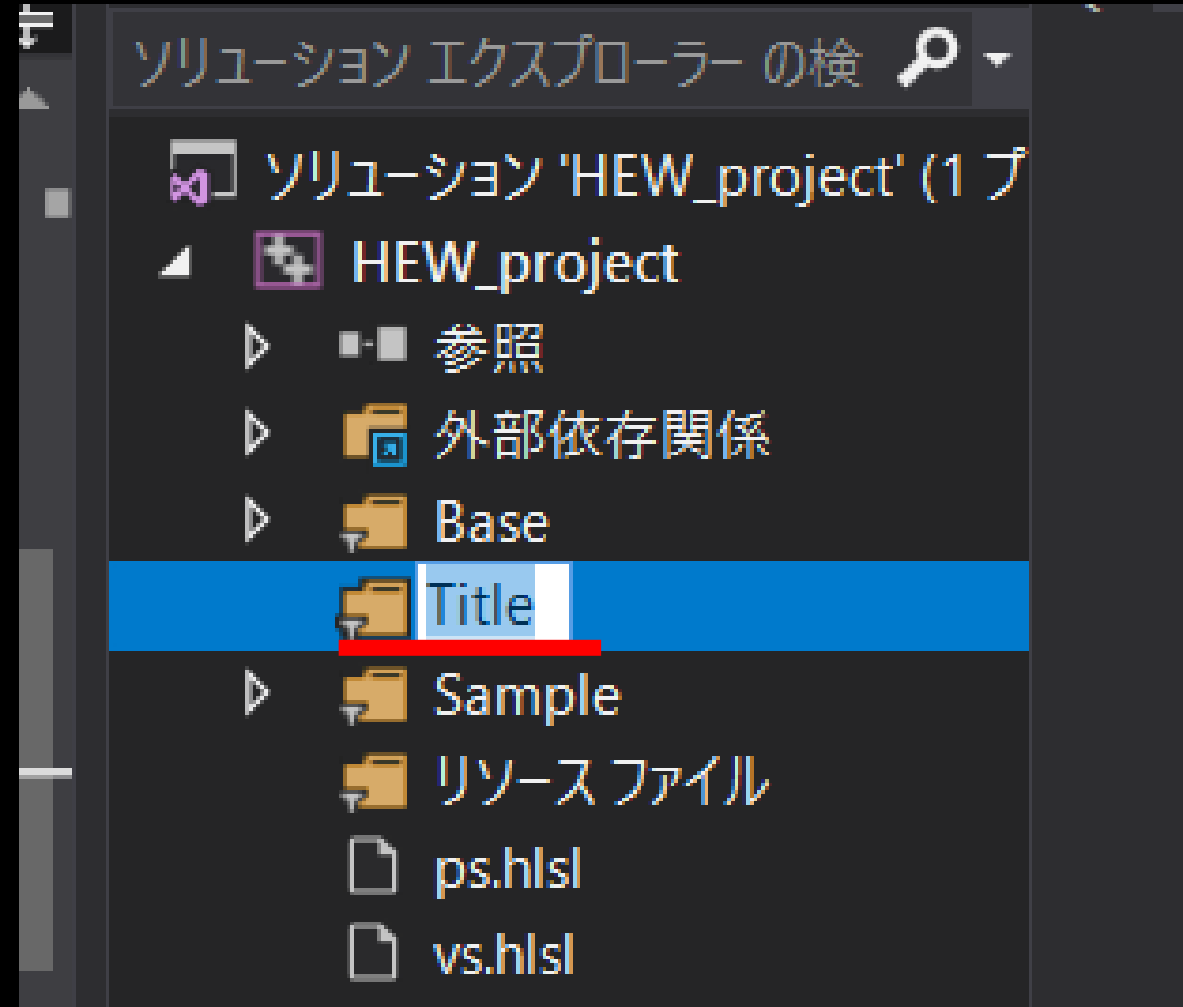
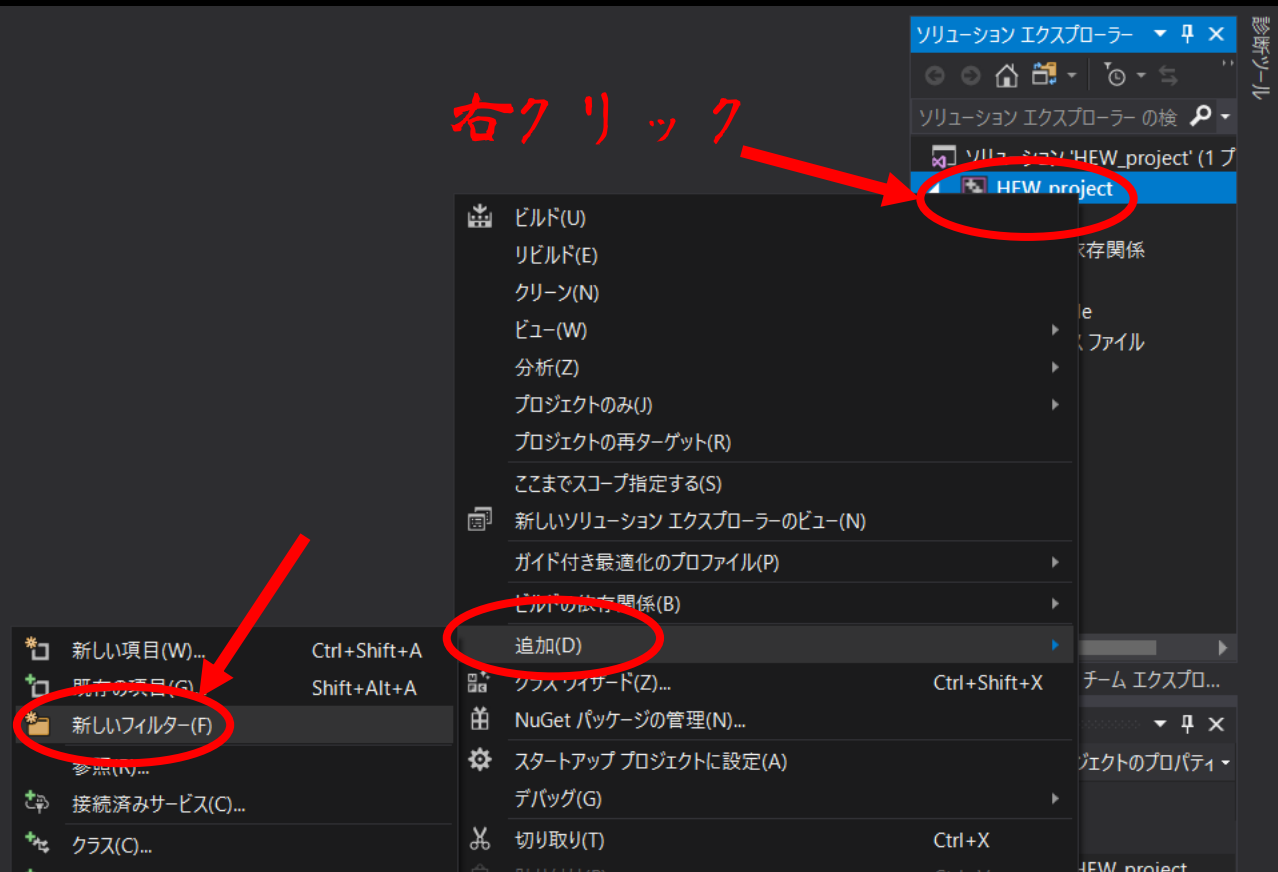
使用ガイド

1. Objectを作ろう

フィルタを作ってかけよう

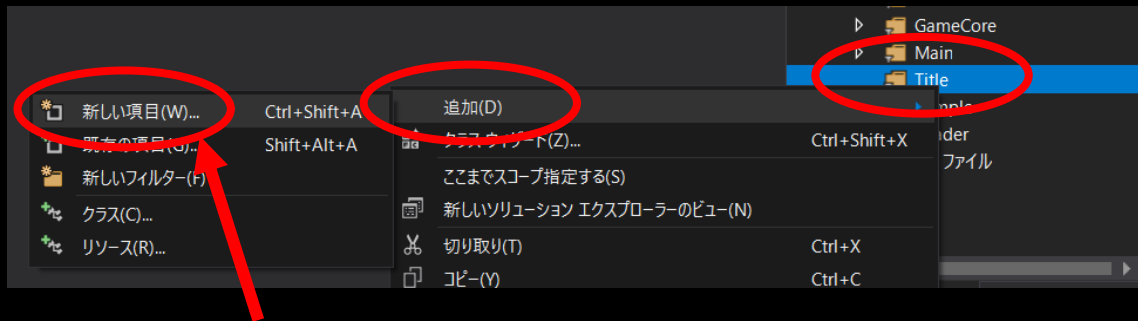
わかりやすい名前にておく

右クリック

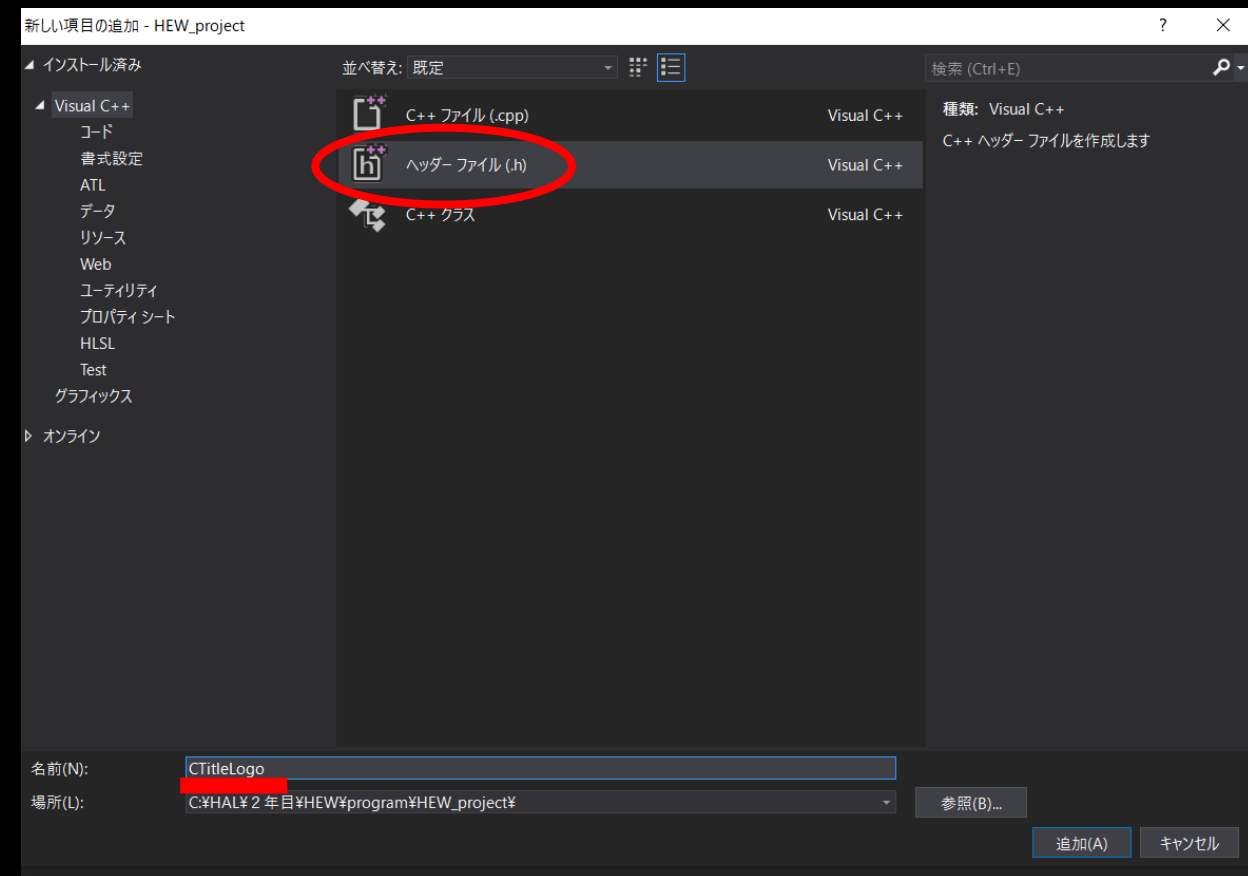


1. Objectを作ろう

新しい項目を選択



ヘッダを作成



1. Objectを作ろう

ヘッダを以下のようにプログラミングする

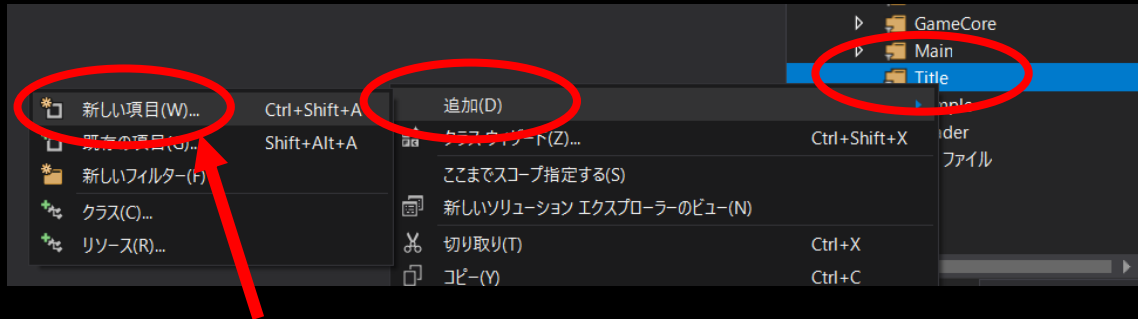
```
#pragma once
#include "CObject.h" //オブジェクトクラスを読み込む

class CTitleLogo : public CObject //オブジェクトクラスを継承
{
private:
    const Vector3 titleSize{1.25f, 1.25f*0.25f}; //タイトルのサイズ
    const Vector3 titlePosition{0, 0.3f, 0}; //タイトルの座標
    const int titleLayer = LayerNum::UI; //タイトルのレイヤー番号
public:
    CTitleLogo();
    ~CTitleLogo();

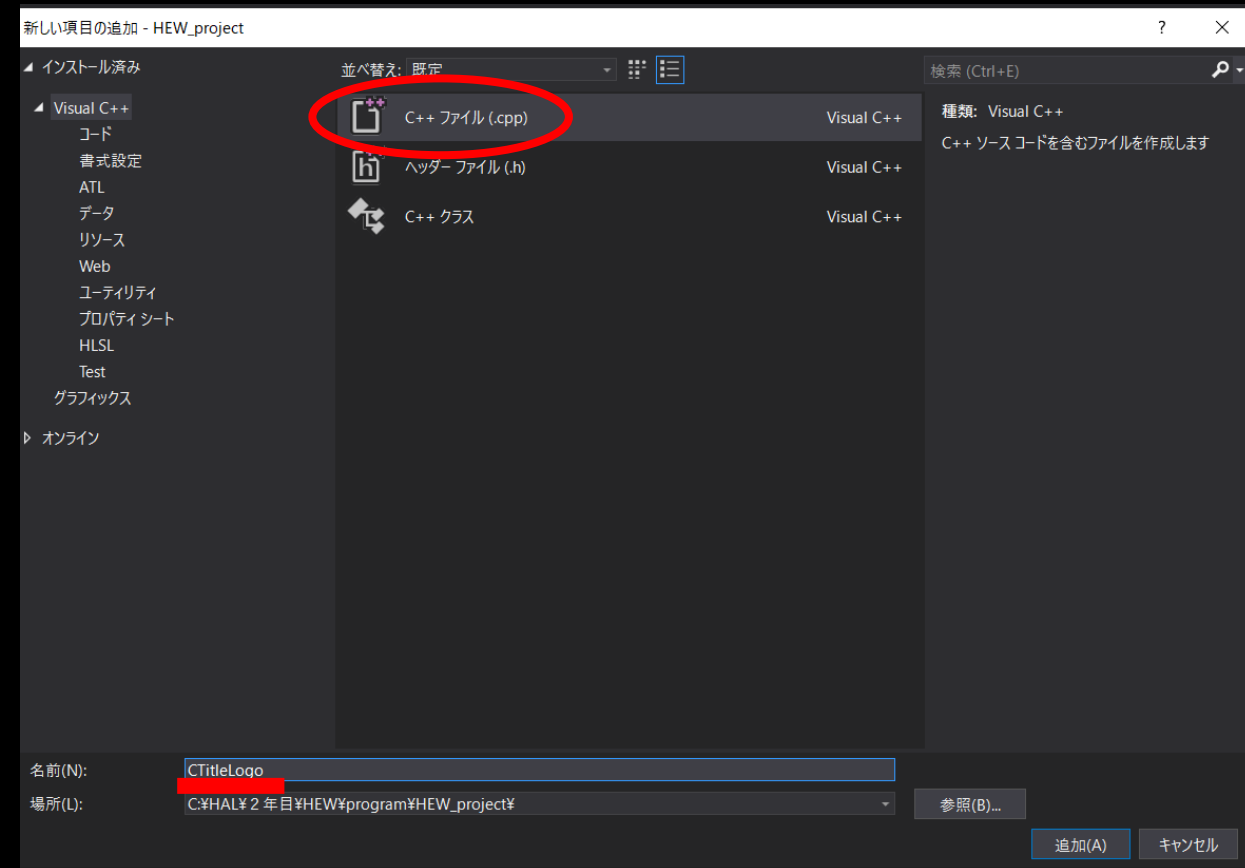
    static ID3D11ShaderResourceView* sTexture; //テクスチャを置いておく静的変数
    static wchar_t cTexturePath[PATH_LENGTH]; //タイトルテクスチャのパス
};
```

1. Objectを作ろう

新しい項目を選択



ヘッダと同名のcppを作成



1. Objectを作ろう

cppを以下のようにプログラミングする

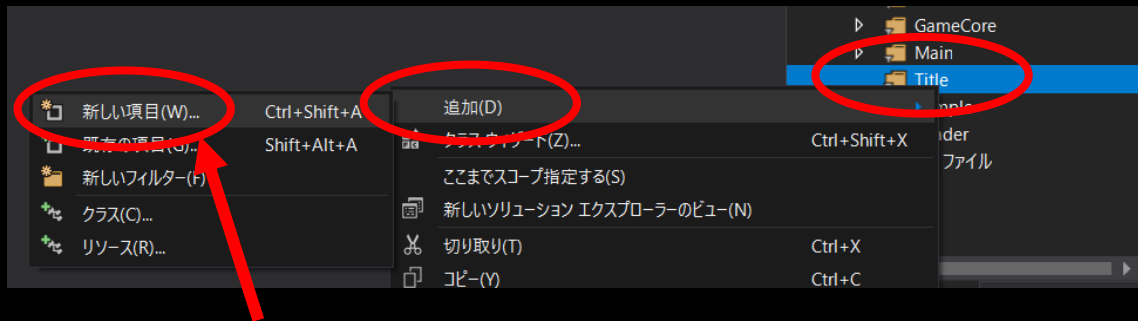
```
#include "CTitleLogo.h"

ID3D11ShaderResourceView* CTitleLogo::sTexture; //テクスチャ用の静的変数
wchar_t CTitleLogo::cTexturePath[PATH_LENGTH] = L"Assets/TitleLogo.png"; //テクスチャのパス

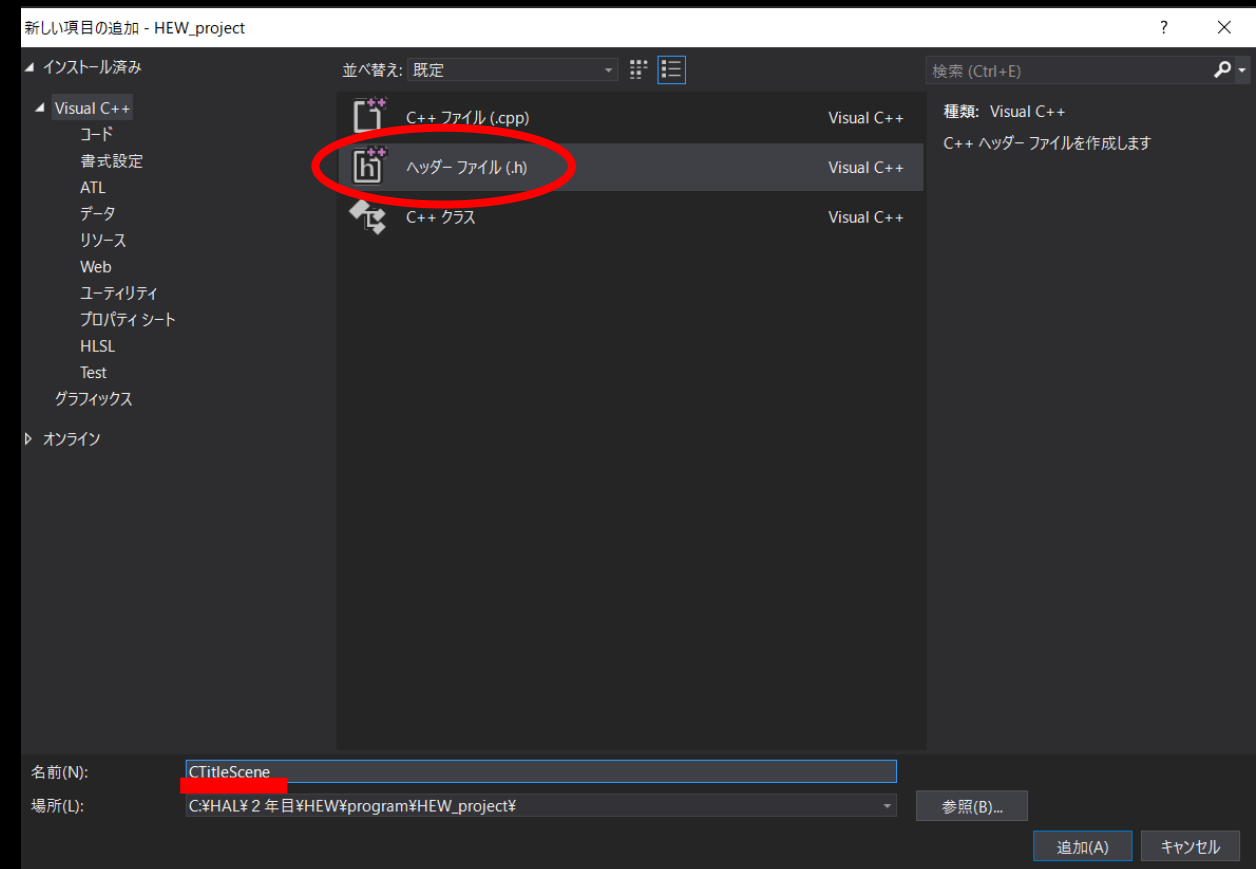
CTitleLogo::CTitleLogo() //コンストラクタ
{
    SetLayerNum(titleLayer); //レイヤー番号をLayerNum::UI;
    SetScale(titleSize); //サイズを変更
    SetPosition(titlePosition); //座標を変更
    SetTexture(sTexture); //テクスチャに静的変数を代入
}
```

2. Sceneを作ろう

新しい項目を選択



ヘッダを作成



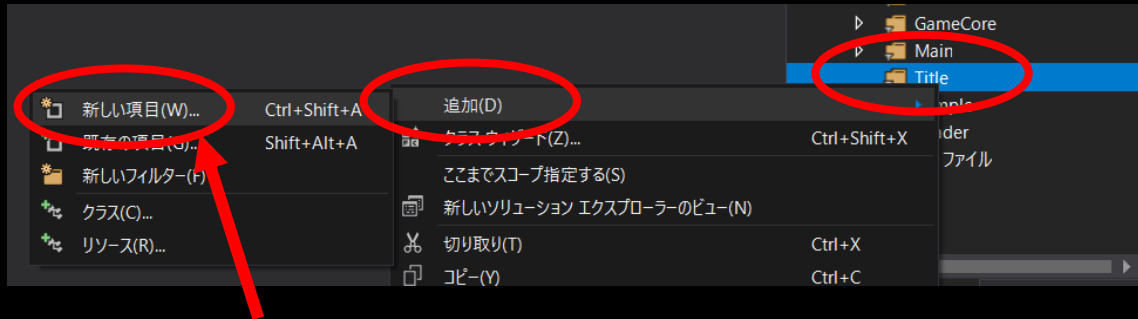
2. Sceneを作ろう

ヘッダを以下のようにプログラミングする

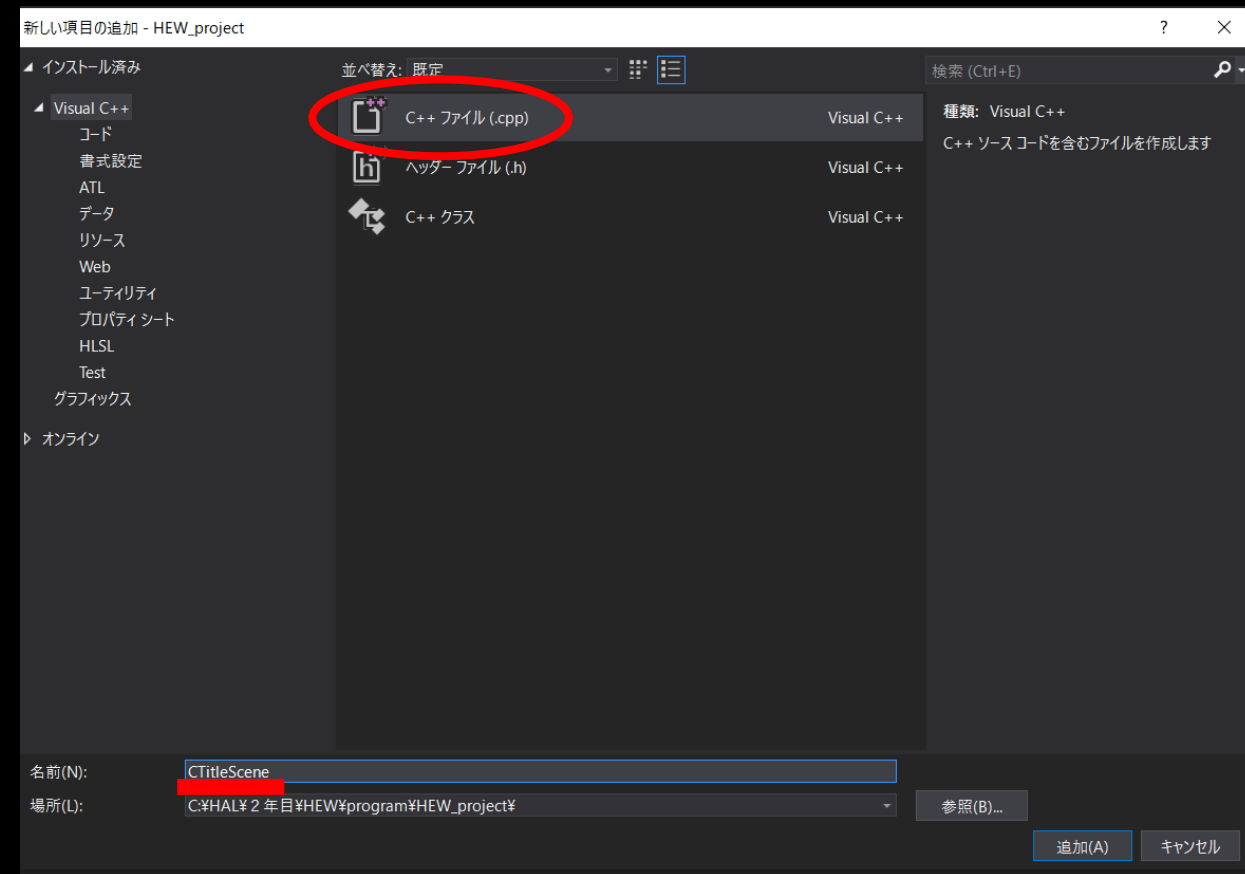
```
project (グローバル スコープ)
1  #pragma once
2  #include "CScene.h" // シーンクラスを読み込む
3  #include "CTitleLogo.h" // 作ったオブジェクトを読み込む
4
5  class CTitleScene : public CScene // シーンクラスを継承
6  {
7  public:
8      CTitleScene();
9      ~CTitleScene() override;
10     void Update() override; // 更新処理をオーバーライド
11 private:
12     bool change = false; // シーン切り替え用フラグ
13
14     /*シーンに使うオブジェクトをここに列挙*/
15     CTitleLogo* tl; // オブジェクト生成用ポインタ
16 }
```

2. Sceneを作ろう

新しい項目を選択



ヘッダと同名のcppを作成



2. Sceneを作ろう

cppを以下のようにプログラミングする

コンストラクタで生成

```
1  #include "CSceneManager.h" → //シーンマネージャを読み込む
2  #include "CTitleScene.h" → //ヘッダを読み込む
3
4  CTitleScene::CTitleScene() → //コンストラクタ
5  {
6      /*クラスの静的変数にテクスチャをロード*/
7      LoadTexture(CTitleLogo::cTexturePath, &CTitleLogo::sTexture);
8
9      tl = new CTitleLogo; → //オブジェクトの生成
10     ObjectRegister(tl); → //オブジェクト配列に登録
11 }
12
13 CTitleScene::~CTitleScene() → //デストラクタ
14 {
15     /*読み込んだテクスチャの解放*/
16     COM_SAFE_RELEASE(CTitleLogo::sTexture);
17 }
18
```

更新処理でScene切り替え

```
void CTitleScene::Update() → //オーバーライドした更新処理
{
    //シーンの更新処理を走らせる
    CScene::Update();

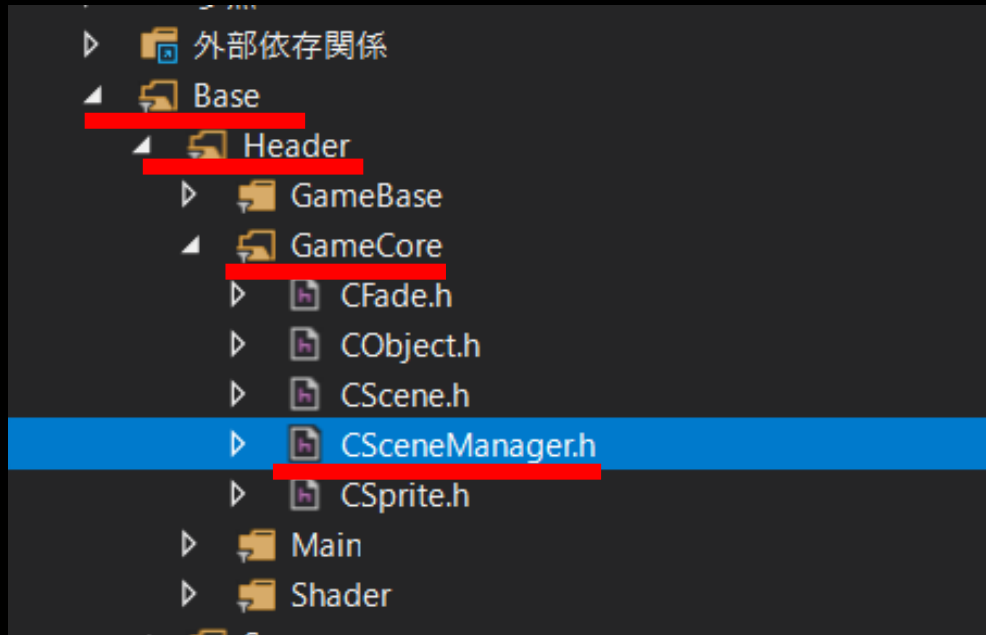
    //スペースキーが押されたらフラグをtrueにする
    if (Input_GetKeyDown(VK_SPACE))
    {
        change = true;
    }

    //フラグがtrueなら
    if (change)
    {
        //フェードオブジェクトにフェードアウトの指示
        fade->FadeOut();

        //フェードのアルファ値が1になったら
        if (fade->alphaOne)
        {
            //シーンを変更
            sceneManager->ChangeScene(SceneNum::STAGE_ONE);
        }
    }
}
```

3. SceneManagerに登録

GameCoreに入ってる
CSceneManager.hを開く



ヘッダのenumにTITLEを追加する

```
#pragma once
#include "CScene.h"

#define FIXED_RATE 60
#define SAFE_DELETE(ptr) delete ptr; ptr = nullptr;

/*シーンの番号*/
/*シーンクラスを作成する度に追加*/
enum SceneNum {
    LOGO_SPLASH, //チームロゴとか出すやつ
    TITLE, //タイトル画面
};

class CSceneManager
{
private:
    CScene* pNowScene = nullptr; //シーンを生成するためのポインタ
    int nSceneNum = LOGO_SPLASH; //シーンの番号を格納しておく変数
    bool sceneChange = false;
    float fixedCount = 0;

public:
    CSceneManager();

    void Play(); //シーンのメイン処理

    void ChangeScene(int); //シーンを変える時に呼ぶ関数
    void SceneFactory(int); //シーンを生成する関数
};
```

3. SceneManagerに登録

SceneManager.cppを開く



作ったSceneをincludeする

```
1 #include "CSceneManager.h"
2 #include "CLogoSplashScene.h"
3 #include "CTitleScene.h" //作ったシーンを追加
4
5 //コンストラクタ
6 CSceneManager::CSceneManager()
7 {
8     SceneFactory(nSceneNum);
```

SceneFactory関数に
作ったSceneを追加

```
44 //シーンを生成するための関数
45 //引数にシーン番号を送ると生成される
46 void CSceneManager::SceneFactory(int _sceneNum)
47 {
48     if (pNowScene != nullptr)
49     {
50         delete pNowScene;
51         pNowScene = nullptr;
52     }
53
54     switch (_sceneNum)
55     {
56     case SceneNum::LOGO_SPLASH:
57         pNowScene = new CLogoSplashScene();
58         break;
59     case SceneNum::TITLE:
60         pNowScene = new CTitleScene();
61         break;
62     }
63
64     //現在のシーンクラスに自分のポインタを譲渡
65     pNowScene->SetManager(this);
66 }
```