

<プログラム解説>

1.制作期間・担当箇所

プログラムの制作期間は 2024 年 6 月下旬～2024 年 7 月上旬までの約3週間で
す。

以下のプログラム全体の設計と基礎を担当しました。Stage1 と Stage3 は友人が担
当しました。

ソースファイル/Main.cpp

ソースファイル/MyLib

ソースファイル/Scene

ソースファイル/Game/Stage2

2.プログラムのアピールポイント

・ポリモーフィズムを利用したメモリ管理

Goforbroke で反省点としてあげたメモリ管理を改善しました。

前は SceneManager.h にて全てのシーンをインスタンス化しました。ゲーム開
始時に必要のないシーンも初期化されメモリを使ってしまう問題を cpp 内でメモ
リ確保と破棄を行うことで解決しました。

最初に SceneBase.h(右図)を作りこれを継承してそ
れぞれゲームシーンクラスを作成しました。書き忘
れ防止のため純粋仮想関数で定義し、それぞれの
シーンクラスは Loop と Draw 関数を呼び出せば動
く仕組みにしました。

場所 ソースファイル/Scene/SceneBase.h

```
16  
17  
18 public:  
19     virtual int Loop() = 0;  
20     virtual void Draw() = 0;  
21  
22     void Init();           //初期化  
23     void Load();          //ロード  
24     void Step();          //ステップ  
25     void Exit();          //破棄  
26     void Next();          //次へ  
27
```

シーンを管理する SceneManager.h にてポインタで宣言します。switch 文でシーンを分け、それぞれ切り替わるときに *cScene に delete または new して必要なクラスだけ一時的にメモリを確保するように改善しました。(右図)

Scene/SceneManager.cpp

```
39 switch (m_eAllSceneID)
40 {
41     case SCENE_TITLE:
42         if (cScene->Loop() == 1)
43         {
44             delete cScene;
45             cScene = new CPlay();
46             m_eAllSceneID = SCENE_PLAY;
47         }
48         break;
```

•Stage 内のみ変更すれば動く仕組みづくり

チーム制作を意識して他の人がシーン切り替えやシステムを変更しなくて済むように組みました。ベース(Game/GameBase)を作りステージ全てで継承しています。m_base にクラスを確保し、全ステージに作成されている Init~Exit までを m_base から呼び出します。

こうすることで、ほかのメンバーは担当箇所のステージ以外を触らなくて済み、触ってエラーが起きることを防ぎました。

Scene/Play/Play.cpp

```
91 // 古いシーン削除
92 if (m_base != nullptr)
93 {
94     if (m_base->GetResult() == 1)
95     {
96         delete m_base;
97         m_next += 1;
98     }
99     else
100         delete m_base;
101 }
102
103 // 新しいシーン作成
104 switch (m_next)
105 {
106     case STAGE_1:
107         m_base = new CStage1;
108         break;
109     case STAGE_2:
110         m_base = new CStage2;
111         break;
112     case STAGE_3:
113         m_base = new CStage3;
114         break;
115 }
```

長くなりましたが、最後まで読んでいただきありがとうございました。