

<プログラム解説>

1.制作期間

プログラムの制作期間は2023年11月～2024年4月までの約5か月です。
11月上旬までにモック制作、企画書作成を終わらせ、11月中旬から本格的に作り始めました。11月は必要なモデルを制作したり、2D画像を制作したりしました。
12月からステージ、プレイヤー、天球、敵、様々な必要なものを描画しました。
1月～3月はゲーム内容を面白くすることを中心にプログラムの変更を行いました。
4月は必要ない変数の削除や関数の省略、他の人でも読みやすいコードにすることを意識してコードを整理しました。

2.プログラムのアピールポイント

・シングルトンクラス

苦手意識のあるシングルトンに挑戦しました。
プログラム内にインスタンスが一つだけあればいい「Fade クラス」に練習として取り入れました。その他にも「FPS クラス」や「Input クラス」など使える場面があることに気が付きましたが、今回は時間が足りず、実装できませんでした。

場所:ソースファイル/MyLib/Fade

```
3
4     CFade* CFade::m_instance = nullptr;
5
6     //インスタンス取得
7     CFade* CFade::GetInstance()
8     {
9         Alloc();
10        return m_instance;
11    }
12    //クラスを生成してメモリ確保
13    void CFade::Alloc()
14    {
15        if (m_instance == nullptr)
16        {
17            m_instance = new CFade;
18        }
19    }
20    //破棄
21    void CFade::DeleteInstance()
22    {
23        if (m_instance != nullptr)
24        {
25            delete m_instance;
26            m_instance = nullptr;
27        }
28    }
```

・自作ライブラリの作成

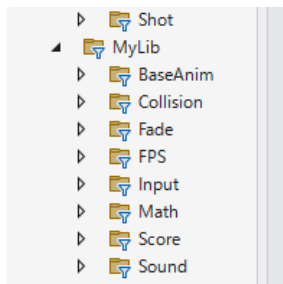
ゲームを作る際に必須なFPSや当たり判定、キー入力など毎回はじめから作らなくてもいいように「MyLib」ファイルを作成しました。
プログラム起動時から終了時まで使い続ける、かつプログラム全体で使用するような機能のクラスはstaticクラスとして作成し、別の場所から引数に値を渡してメソッドを呼ぶだけで問題なく動くように作りました。

「BaseAnim」に関しては、プレイヤーや敵に継承されることを前提に考え後から

上書きできるように仮想関数を使用しました。

場所:ソースファイル/MyLib/(FPS/Input/Collision)

ソースファイル/MyLib/BaseAnim



BaseAnim.cpp

```
37 public:
38     virtual void Init();           //初期化
39     virtual void Draw();          //描画処理
40     virtual void Exit();          //終了処理
41     virtual void UpData();        //更新処理
```

・コードの可読性を高めることを意識

読む人に「この処理は何のためにあるのか?」という疑問を持たせないように意識して作りました。複雑な処理はコメントを残し、読み間違いがないように気を付けました。無駄な変数や関数を省き、短くできるコードは短くしました。例えば、初期化→ループ→破棄を switch 文で場面を分けて管理していたところ、関数ポインタを使いそれぞれの処理関数を呼ぶ別の書き方で短くまとめました。

場所:ソースファイル/Scene/PlayScene/SceneBase

```
5 //毎ループ
6 void CBaseScene::Loop()
7 {
8     //関数ポインタを宣言し関数を配列に入れる
9     void (CBaseScene::* scene[]) () = {
10         &CBaseScene::Init, &CBaseScene::Load, &CBaseScene::AmStep, &CBaseScene::PmStep, &CBaseScene::Exit, &CBaseScene::Next
11     };
12     //idに合わせて処理を実行
13     (this->*scene[m_eSceneID]) ();
14 }
15
```

3. プログラムの反省点・課題

・メモリ管理

シーン別に New と delete をうまく使えていないことが反省点です。ゲームが完全に終了するまで無駄なメモリが確保されたままです。タイトルシーンが終わればタイトルクラスを使用することはないので無駄なメモリを削除するべきだと思います。

した。ほかにもコンストラクタで初期化を行っているため最初に使わないクリアやゲームオーバークラスのメモリがすでに確保されていることが問題だと思いました。

•プログラム内にデータを直接書き込んでいる

できるだけ外部データから読み込むことがベストだと教わりました。例えば MyLib の中の lput.cpp、push_back を使って設定していますが、ここもバイナリファイルを読み込むようにしたり、場合によっては CSV ファイルを使って外部から読み込むことが後から変更するときに簡単でほかの人も修正しやすいと思いました。

•Swich 文や、else if に頼りすぎている

これらを多用しすぎると無駄に長いコードになってしまったり、読みづらくなることを学びました。そこから早期 return を意識したり上記で記載した関数ポインタを使用したりしてスクロールしなくても読みやすいコードを意識しました。ですがまだ様々な場面で頼っている部分が多いことを自覚しています。今後は使わないようにするのではなくコードの読みやすさを一番に、使い分けができればもう少し成長ができると考えています。

長文になりましたが、最後まで読んでいただきありがとうございました。