

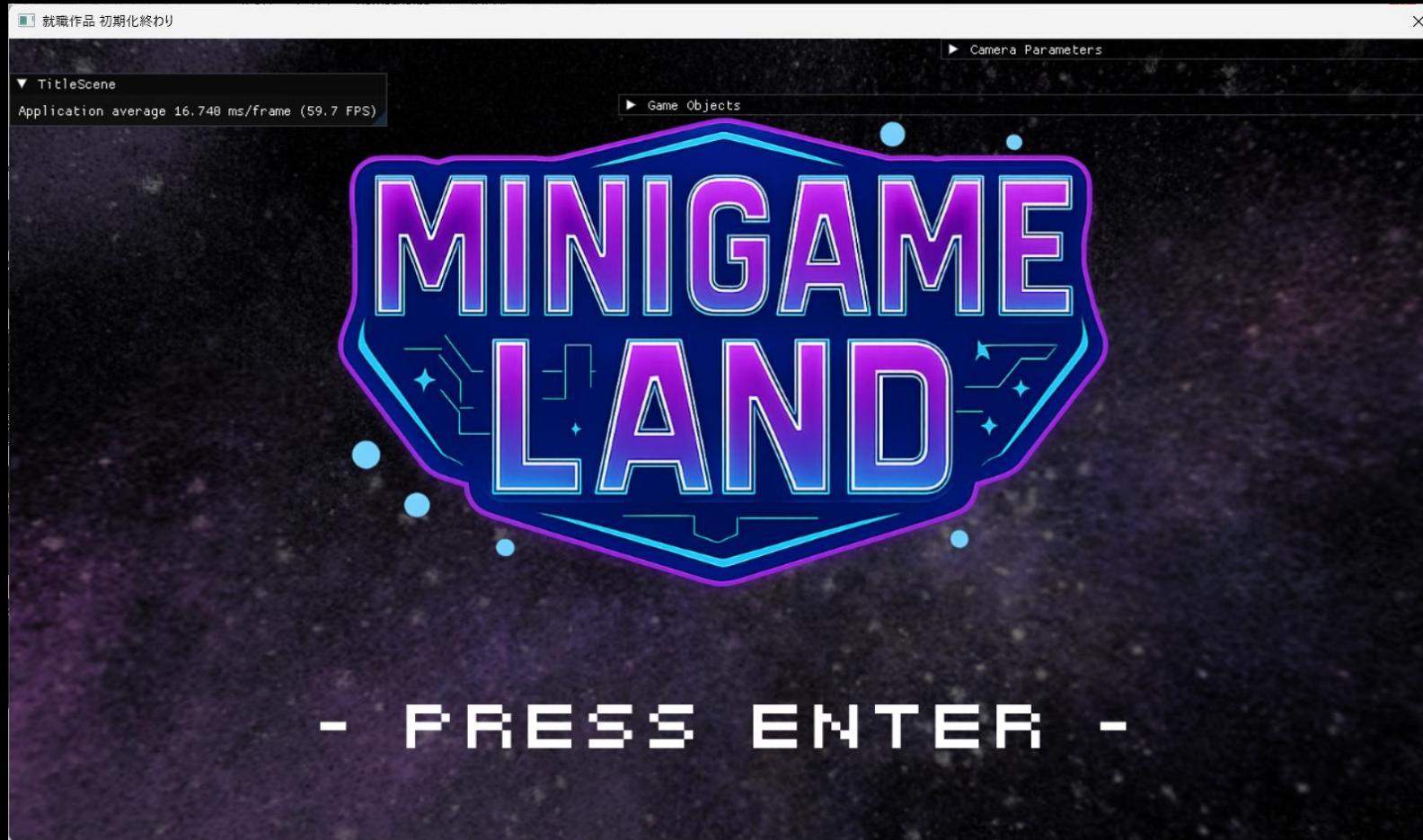
途中進捗
11/19

ゲーム学科3年 西口 煌大
生年月日:2004 11/19(21歳)

タイトル：MINIGAME LAND

画像：Debugビルド タイトルシーン

ジャンル：2Dミニゲーム集
C++20 / DirectX11



概要

- ・フルスクラッチしたフレームワークで
「メイドインワリオの再現」を制作して
ます。
 - なぜフルスクラッチなのか...
 - 低レベルな仕組みから理解して制作したいから。
 - 私自身が分かりやすく直感的に扱いたいと思ったから

11/8(土)時点の進捗

フレームワークの作成と
シーン遷移（演出なし）ができる程度です。

目標：違和感のないゲームにする！

本題：現在進捗

- INゲームの待機シーン・稼働シーンの作成
 - シーン間の必要データ受け渡し
 - 3ステージのプロト作成（要改善）
 - 待機シーンでのライフ作成
 - 稼働シーン突入のシークエンス作成
- タイトルシーンUI改善

現在、速度重視で動いています。。。

詳細は次のページから

シーン間の必要データ受け渡し

```
// Sceneクラス間の受け渡しデータ  
// isClear      : ミニゲームをクリアしたかのフラグ  
// requestRetry : ゲームをもう一度するかのフラグ  
// stageCount   : 何ステージクリアしているか  
// previousScene : 現在のシーン  
// oldScene     : 一つ前のシーン  
// nextScen    : 次のシーン  
  
class SceneRelationData  
{  
public:  
    bool isClear = false;  
    bool requestRetry = false;  
    int stageCount = 0;  
    SCENE_NO previousScene = SCENE_NO::NONE;  
    SCENE_NO oldScene     = SCENE_NO::NONE;  
    SCENE_NO nextScen    = SCENE_NO::NONE;  
};
```

最初は動的にしていましたが
バグ処理に時間がかかる...
例:nextScene->Set(oldScene->Get());
※関数名は仮です

staticに...

```
static SceneRelationData m_RelationData;
```

必要なデータだけを持ってくる。

3ステージのプロト作成（要改善）

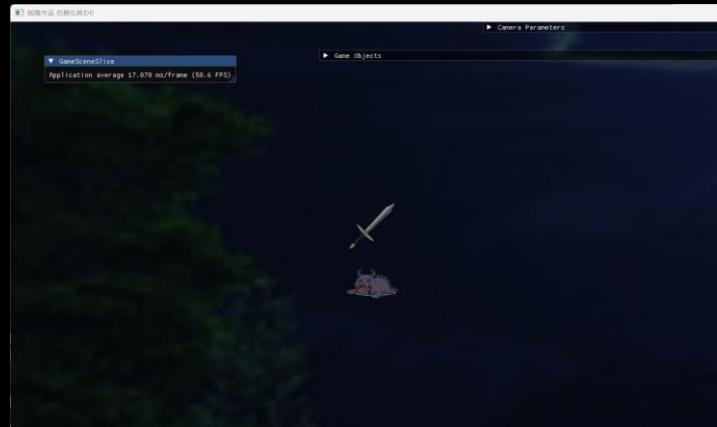
- ・3ステージとも基盤の作成を行いました。

できていること

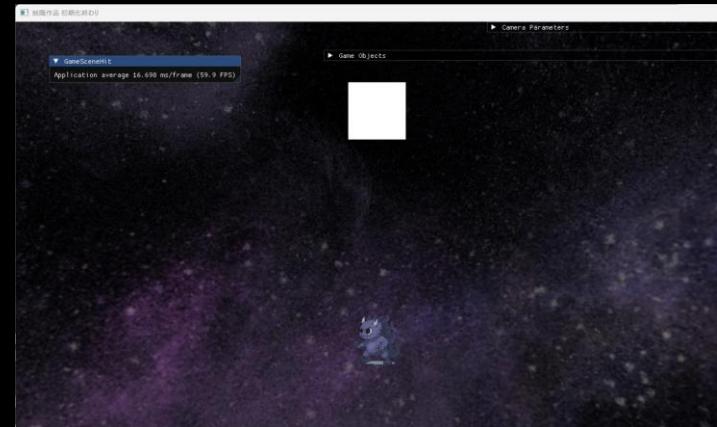
- ・汎用的なクリア条件の処理
- ・敵を倒す処理
- ・スピードアップ考慮の調整

できていないこと

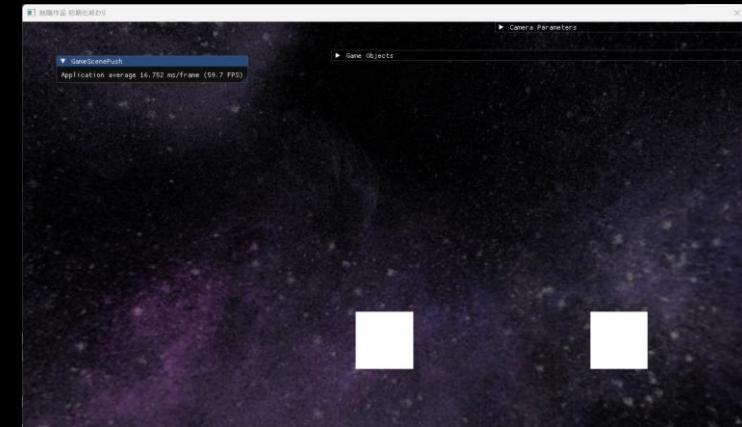
- UI（お題提示やパーティクル
- 難易度に応じた敵の配置・処理
- 難易度・スピードアップ処理 etc..



敵を切るゲーム
必要操作
上下左右移動



敵をたたくゲーム
必要操作
左右移動 Enter



車をよけるゲーム
必要操作
Enter

待機シーンでのライフ作成

- 作成のみできています
ステージ失敗 → ライフが減るは未実装



稼働シーン突入のシーケンス作成

- 連続でおなじステージに入らないように作成

超脳筋で作成しています

```
// 全ステージから選ぶ時の乱数を取得する関数

SCENE_NO GameSceneWait::StageSelectAllRandom()

{
    SCENE_NO stageKinds[3] = {
        SCENE_NO::GAME_SLICE,
        SCENE_NO::GAME_HIT,
        SCENE_NO::GAME_PUSH
    };

    // 最初の一回だけ作成する乱数は3パターンの中から選択
    std::uniform_int_distribution<int> dist(0, 2);
    int selectedIdx = dist(m_RandomEngine);
    return stageKinds[selectedIdx];
}
```

```
// oldを見てから2パターンの選択
else
{
    switch (m_RelationData.oldScene)
    {
        case SCENE_NO::GAME_SLICE:
            _NextScene = RandomChoose<SCENE_NO>(SCENE_NO::GAME_PUSH,
                                                    SCENE_NO::GAME_HIT);
            break;
        case SCENE_NO::GAME_PUSH:
            _NextScene = RandomChoose<SCENE_NO>(SCENE_NO::GAME_SLICE,
                                                    SCENE_NO::GAME_HIT);
            break;
        case SCENE_NO::GAME_HIT:
            _NextScene = RandomChoose<SCENE_NO>(SCENE_NO::GAME_PUSH,
                                                    SCENE_NO::GAME_SLICE);
            break;
        default:
            _NextScene = StageSelectAllRandom();
            break;
    }
}
```

タイトルシーンUI改善



- ① タイトルロゴがふわふわ
- ② Press Enterがチカチカ
- ③ スカイドームがくるくる

以上です。

タイトルシーンにおいては
UI面の制作が進みました。

今後の展望

目標の「違和感のないゲーム」を作るために...

- ① リザルトシーンのパラメータ作成
- ② ゲームループ完了

----- INゲーム最低ライン -----

- ③ リズム感の導入（BGM・ジングルに合わせたBPMやUI作成）
- ④ スピード・難易度アップのパラメータ処理
- ⑤ 難易度対応の敵拳動作成

----- UI超最低ライン -----

12月審査会までに最低でも④まで
作成して見せれる形に

補足：フレームワークについて

10月の審査会向けに作成した資料もいれてます。
ご興味やお時間あるとき御覧いただけたら幸いです。

本ドライブやリンクから見れます。

実装重視で励んでいます。

ご覧いただきありがとうございました。