

12月審査会



# つたえる ポート イン フォリオ HAL

4年制ゲーム学科3年  
西口 榮大

## ● どんなプログラマ？

遊んだ時の**感触**を意識する

**体験志向**プログラマ

使用言語 : C++20 / DirectX11

制作期間 : 2ヶ月(個人制作)

ゲーム

メイドインクリオの再現

モクヒョウ

違和感のないゲーム

イシキ

遊ぶ人の視点

## ● 進捗(シンチョク)

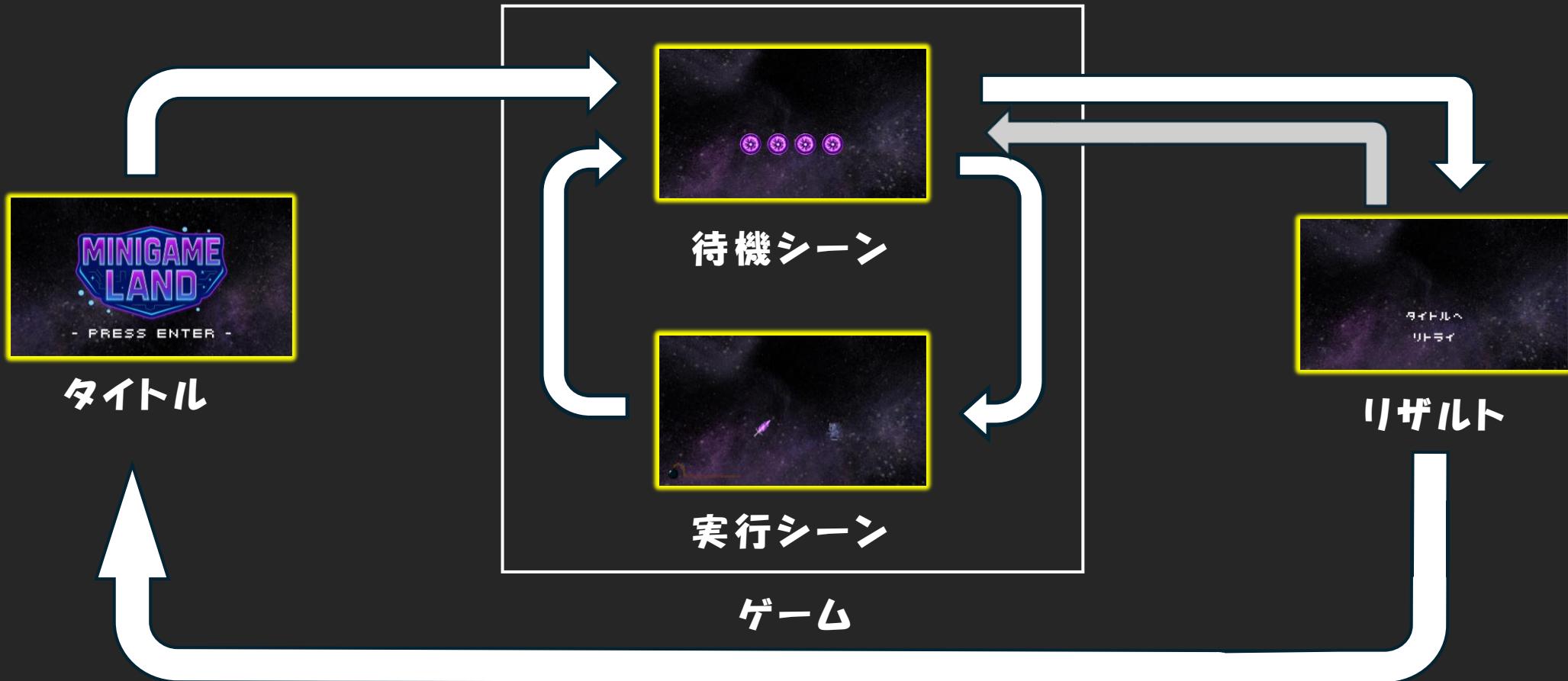
ゲームループ完成

リズム感の実装

**最重要要件**

**没入感 UP  
違和感 DOWN**

# ●ゲームフロー



## ●違和感を無くすために

### リズム感

- ① オブジェクトに乗せる
- ② 待機→実行の遷移は3小節
- ③ 早くクリアした場合の処理



# リズムをとるために

```
1  struct RhythmBeatConst
2  {
3      // 入力値
4      float bpm = 120.0f;           // BPM (Beats Per Minute)
5      int beatUnit = 4;            // 拍子の分母 (4 = 4分音符, 8 = 8分音符, etc.)
6      int ticksPerBeat = 16;       // 1拍を何分割するか
7
8      // 計算結果
9      float secondsPerBeat = 0.0f; // 1拍の長さ (秒)
10     float secondsPerBar = 0.0f; // 1小節の長さ (秒)
11     float secondsPerTick = 0.0f; // 1Tickの長さ (秒)
12     float ticksPerSecond = 0.0f; // 1秒あたりに進むTick数
13
14     // Setup (セットアップ) :
15     // 設定をまとめて反映する関数名によく使われる
16     void Setup(float bpm_, int beatsPerBar_ = 4, int ticksPerBeat_ = 16)
17     {
18         bpm = bpm_;
19         beatUnit = beatsPerBar_;
20         ticksPerBeat = ticksPerBeat_;
21
22         // 1拍の長さ (秒) = 60 / BPM
23         secondsPerBeat = 60.0f / bpm;
24
25         // 1小節の長さ (秒) = 1拍 * 拍数
26         secondsPerBar = secondsPerBeat * static_cast<float>(beatUnit);
27
28         // 1Tickの長さ (秒) = 1拍 / Tick数
29         secondsPerTick = secondsPerBeat / static_cast<float>(ticksPerBeat);
30
31         // 1秒あたりのTick数 = 1 / secondsPerTick
32         ticksPerSecond = 1.0f / secondsPerTick;
33     }
34 }
35 
```

FPS、BPM、拍、1拍の分割  
4つの変数から拍の間隔の値を算出

```
37  class RhythmBeat
38  {
39      private:
40          RhythmBeatConst m_Beat{};
41          float           m_TickCounter = 0.0f;           // 経過時間の蓄積
42          int             m_TickIndex   = 0;             // 現在のTick数
43          int             m_Advance     = 0;             // 現在の拍子インデックス
44
45      public:
46          RhythmBeat() = default;
47          // 初期化
48          void Initialize(const RhythmBeatConst& config);
49          // 更新
50          int Update(float deltaTime);
51
52          // 現在のTickを取得
53          int GetTotalTick() const
54          {
55              return m_TickIndex;
56          }
57
58          // 現在の拍子インデックスを取得
59          // 今が何拍目かを返す
60          int GetBeatIndex() const
61          {
62              return m_TickIndex / m_Beat.ticksPerBeat;
63          }
64
65          // 現在の拍子内のTick位置を取得
66          // 今の拍子の中で何Tick目かを返す
67          int GetTickInBeat() const
68          {
69              return m_TickIndex % m_Beat.ticksPerBeat;
70          }
71
72          const RhythmBeatConst& GetBeatConst() const
73          {
74              return m_Beat;
75          }
76
77          int GetAdvance() const
78          {
79              return m_Advance;
80          }
81
82 }
```

算出された値を元に  
現在の拍数を監視

# ● 今後の展望

## 1 未実装項目

情報

リザルトのスコア・オブジェクトの画像表示

演出

ライフ消失のパーティクル描画・スピードアップ

インゲーム

敵の挙動制御

## 2 優先順位

情報

→

演出

→

インゲーム

ご覧いただきありがとうございました。