

```
#include "SceneManager.h"
#include "../Utility/InputControl.h"
#include "DxLib.h"
#include "TitleScene.h"
#include "GameMainScene.h"
#include "ResultScene.h"
#include "HelpScene.h"
#include "RankingDispScene.h"
#include "RankingInputScene.h"

SceneManager::SceneManager() : current_scene(nullptr)
{

}

SceneManager::~SceneManager()
{

}

// シーンマネージャー機能：初期化处理
void SceneManager::Initialize()
{
    // ウィンドウのタイトルを設定
    SetMainWindowText("Drive&Avoid");

    // ウィンドウモードで起動
    if (ChangeWindowMode(TRUE) != DX_CHANGESCREEN_OK)
    {
        throw ("ウィンドウモードで起動できませんでした\n");
    }

    // DXライブラリの初期化
    if (DxLib_Init() == -1)
    {
        throw ("Dxライブラリが初期化できませんでした\n");
    }

    // 描画先指定処理
```

```

    if (SetDrawScreen(DX_SCREEN_BACK) == -1)
    {
        throw ("描画先の指定ができませんでした\n");
    }

    // タイトルシーンから始める
    ChangeScene(eSceneType::E_TITLE);
}

// シーンマネージャー機能：更新処理
void SceneManager::Update()
{
    // フレーム開始時間（マイクロ秒）を取得
    LONGLONG start_time = GetNowHiPerformanceCount();

    // メインループ
    while (ProcessMessage() != -1)
    {
        // 現在時間を取得
        LONGLONG now_time = GetNowHiPerformanceCount();

        // 1フレーム当たりの時間に到達したら、更新および描画処理を行う
        if ((now_time - start_time) >= DELTA_SECOND)
        {
            // フレーム開始時間を更新する
            start_time = now_time;

            // 入力機能：更新処理
            InputControl::Update();

            // 更新処理(戻り値は次のシーン情報)
            eSceneType next = current_scene->Update();

            // 描画処理
            Draw();

            // エンドが選択されていたら、ゲームを終了する
            if (next == eSceneType::E_END)
            {

```

```

        break;
    }

    // 現在のシーンと次のシーンが違っていたら、切り替え処理を行う
    if (next != current_scene->GetNowScene())
    {
        ChangeScene(next);
    }
}

// ESCAPEキーが押されたら、ゲームを終了する
if (CheckHitKey(KEY_INPUT_ESCAPE) ||
InputControl::GetButtonUp(XINPUT_BUTTON_BACK))
{
    break;
}
}
}

```

// シーンマネージャー機能：終了時処理

```

void SceneManager::Finalize()
{
    // シーン情報の削除
    if (current_scene != nullptr)
    {
        current_scene->Finalize();
        delete current_scene;
        current_scene = nullptr;
    }

    // DXライブラリの使用を終了する
    DxLib_End();
}

```

// シーンマネージャー機能：描画処理

```

void SceneManager::Draw() const
{
    // 画面の初期化

```

```

ClearDrawScreen();

// シーンの描画
current_scene->Draw();

// 裏画面の内容を表画面に反映
ScreenFlip();
}

// シーン切り替え処理
void SceneManager::ChangeScene(eSceneType scene_type)
{
    // シーンを生成する (SceneBaseが継承されているか?)
    SceneBase* new_scene = dynamic_cast<SceneBase*>(CreateScene(scene_type));

    // エラーチェック
    if (new_scene == nullptr)
    {
        throw ("シーンが生成できませんでした。 \n");
    }

    // 前回シーンの終了時処理を行う
    if (current_scene != nullptr)
    {
        current_scene->Finalize();
        delete current_scene;
    }

    // 新しく生成したシーンの初期化を行う
    new_scene->Initialize();

    // 現在シーンの更新
    current_scene = new_scene;
}

// シーン生成処理
SceneBase* SceneManager::CreateScene(eSceneType scene_type)
{
    // 引数(シーンタイプ)によって、生成するシーンを決定する

```

```
switch (scene_type)
{
    case eSceneType::E_TITLE:
        return new TitleScene;
    case eSceneType::E_MAIN:
        return new GameMainScene;
    case eSceneType::E_RESULT:
        return new ResultScene;
    case eSceneType::E_HELP:
        return new HelpScene;
    case eSceneType::E_RANKING_DISP:
        return new RankingDispScene;
    case eSceneType::E_RANKING_INPUT:
        return new RankingInputScene;
    default:
        return nullptr;
}
}
```