

```

//-----
// インベーターゲーム
//-----
#include <DxLib.h>
#include <time.h>
#include "main.h"
#include "Player.h"
#include "PlayerShot.h"
#include "Enemy.h"
#include "EnemyShot.h"
#include "SceneTitle.h"
#include "SceneGame.h"
#include "SceneGameOver.h"

//-----
// 変数定義
//-----
eSceneKind nowScene;           // 現在のシーンの種別を格納する変数
// キーの状態
int nowSpaceKey;               // スペースキーの現在の状態
int prevSpaceKey;              // スペースキーの直前の状態

//-----
// WinMain関数
//-----
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstanced, LPSTR lpCmdLine, int nCmdShow)
{
    if (SystemInit() == false) {
        return -1;           // エラー終了
    }
    GameInit();
    //-----
    // ゲームループ
    //-----
    while (ProcessMessage() == 0 && CheckHitKey(KEY_INPUT_ESCAPE) == 0) {
        // スペースキーの状態を更新
        prevSpaceKey = nowSpaceKey;           // 現在のキー状態を直前のキー状態にコピー保存する
        nowSpaceKey = CheckHitKey(KEY_INPUT_SPACE); // 現在のキーの状態を更新する
        //-----
        // テスト用処理
        //-----
        prevWKeyPress = nowWKeyPress;
        nowWKeyPress = CheckHitKey(KEY_INPUT_W);
        prevLKeyPress = nowLKeyPress;
        nowLKeyPress = CheckHitKey(KEY_INPUT_L);
        prevOKeyPress = nowOKeyPress;
        nowOKeyPress = CheckHitKey(KEY_INPUT_O);
        //-----
        prevF1KeyPress = nowF1KeyPress;
        nowF1KeyPress = CheckHitKey(KEY_INPUT_F1);
        prevF2KeyPress = nowF2KeyPress;
        nowF2KeyPress = CheckHitKey(KEY_INPUT_F2);

        switch (nowScene) {
            case SCENE_KIND_TITLE:
                SceneTitleMainProc();
                break;
            case SCENE_KIND_GAME:
                SceneGameMainProc();
                break;
            case SCENE_KIND_GAMEOVER:
                SceneGameOverMainProc();
                break;
            default:
                break;
        }
    }
    //-----
    // システムの終了処理
    //-----
    if (Release() == false) return -1;
    return 0;           // ゲームの正常終了
}

```

```

/**
*-----
* 起動時のみの初期化処理
* Input:
*  無し
* Output:
*   true = 正常終了 / false = エラー終了
*-----
*/
bool SystemInit(void)
{
    //-----
    // システム設定
    //-----
    SetWindowText("000000 谷口知士郎"); // ウィンドウのタイトルの設定
    SetGraphMode(WINDOW_SIZE_WID, WINDOW_SIZE_HIG, 32); // ウィンドウのサイズと色モードの設定
    ChangeWindowMode(true); // ウィンドウの表示モード(true=ウィンドウモード/false=フルスクリーン)

    int err = DxLib_Init(); // DXライブラリの初期化
    if (err == -1) return false;

    SRand((int)time(NULL));

    nowScene = SCENE_KIND_TITLE;

    //-----
    // グラフィックの登録
    //-----
    if (PlayerSysInitProc() == false) return false;
    if (PlayerBulletSysInitProc() == false) return false;

    // 敵の画像の読み込み
    if (EnemySysInitProc() == false) return false;
    // 敵弾の初期化
    if (EnemyShotSysInit() == false) return false;
    // タイトル画面の初期化
    if (SceneTitleInit() == false) return false;
    // ゲームオーバー画面の初期化
    if (SceneGameOverInit() == false) return false;

    return true;
}

/**
*-----
* ゲーム開始時の初期化処理
* Input:
*  無し
* Output:
*  無し
*-----
*/
void GameInit(void)
{
    //-----
    // 変数の初期化
    //-----
    nowSpaceKey = prevSpaceKey = 0;

    //-----
    // テスト用処理
    //-----
    prevWKeyPress = nowWKeyPress = 0;
    prevLKeyPress = nowLKeyPress = 0;
    prevOKeyPress = nowOKeyPress = 0;
    //-----
    prevF1KeyPress = nowF1KeyPress = 0;
    prevF2KeyPress = nowF2KeyPress = 0;

    // プレイヤー
    PlayerInitProc();
    // プレイヤー弾
    PlayerBulletInitProc();
    // 敵キャラ
    EnemyInitProc();
}

```

```

/**
 *-----
 * ゲーム終了時の解放処理
 * Input:
 *   無し
 * Output:
 *   無し
 *-----
 */
bool Release(void)
{
    // ゲームオーバー画面の解放処理
    if (SceneGameOverRelease() == false) return false;
    // タイトル画面の解放処理
    if (SceneTitleRelease() == false) return false;
    // ゲーム画面の解放処理
    if (SceneGameReleaseProc() == false) return false;
    // プレイヤーの解放処理
    if (PlayerReleaseProc() == false) return false;
    // プレイヤー弾の解放処理
    if (PlayerBulletReleaseProc() == false) return false;
    // 敵キャラの解放処理
    if (EnemyReleaseProc() == false) return false;
    // 敵弾の解放処理
    if (EnemyBulletReleaseProc() == false) return false;

    if (DxLib_End() == -1) return false; // DXライブラリの終了処理

    return true;
}

```