

```

#include "RankingScene.h"
#include "DxLib.h"
#include "InputControl.h"
#include "SceneManager.h"

/*****
* マクロ定義
*****/
#define RANKING_FILE      ("dat/rankingdata.csv")
#define RANKING_MAX      (10)
#define RANKING_NAME_LEN (11)
/*****
* 型定義
*****/
typedef struct
{
    int rank;                // ランク
    char name[RANKING_NAME_LEN]; // 名前
    int score;              // スコア
}T_RANKING;

typedef struct
{
    int x;
    int y;
}T_CURSOR;

/*****
* グローバル変数宣言
*****/
T_RANKING Ranking_Data[RANKING_MAX];           // ランキングデータ
T_RANKING New_Score;                           // 新しいスコアデータ
int DispMode;                                 // 表示モード

T_CURSOR Cursor;                             // カーソル用変数
int name_num;

/*****
* プロトタイプ宣言
*****/
void file_read(void);                        // ファイル読み込み
void file_write(void);                      // ファイル書き込み
void ranking_sort(void);                    // ランキングソート処理
void ranking_input_name(void);              // 名前入力処理
void ranking_input_name_draw(void);         // 名前入力描画処理

/*****
* ランキング画面：初期化处理

```

```

* 引 数 : なし
* 戻り値 : エラー情報(-1:異常有, -1以外:正常終了)
*****/
int RankingScene_Initialize(void)
{
    int ret = 0;

    file_read();

    switch (DispMode)
    {
        case RANKING_INPUT_MODE:
            Cursor.x = 0;
            Cursor.y = 0;
            name_num = 0;
            break;
        case RANKING_DISP_MODE:
        default:

            break;
    }

    return ret;
}

/*****
* ランキング画面 : 更新処理
* 引 数 : なし
* 戻り値 : なし
*****/
void RankingScene_Update(void)
{
    switch (DispMode)
    {
        case RANKING_INPUT_MODE:
            ranking_input_name();
            break;
        case RANKING_DISP_MODE:
        default:
            if (GetButtonDown(XINPUT_BUTTON_B))
            {
                Change_Scene(E_TITLE);
            }
            break;
    }
}

```

```

/*****
* ランキング画面：描画処理
* 引 数：なし
* 戻り値：なし
*****/
void RankingScene_Draw(void)
{
    int i;

    switch (DispMode)
    {
        case RANKING_INPUT_MODE:
            ranking_input_name_draw();
            break;
        case RANKING_DISP_MODE:
        default:
            for (i = 0; i < RANKING_MAX; i++)
            {
                DrawFormatString(20, 10 + (i * 25), GetColor(255, 255, 255), "%2d,
%10s, %10d", Ranking_Data[i].rank, Ranking_Data[i].name, Ranking_Data[i].score);
            }
            break;
    }
}

/*****
* ランキング画面：画面変更処理
* 引 数：なし
* 戻り値：なし
*****/
void Set_RankingMode(int mode)
{
    DispMode = mode;
}

/*****
* ランキング画面：スコア取得処理
* 引 数：なし
* 戻り値：なし
*****/
void Set_RankingScore(int score)
{
    New_Score.score = score;
}

/*****

```

```

* ランキング画面：ファイル読み込み処理
* 引 数：なし
* 戻り値：なし
*****/
void file_read(void)
{
    FILE* fp = NULL;
    int i;

    OutputDebugString("ファイルを読み込みます");
    fopen_s(&fp, RANKING_FILE, "r");

    if (fp == NULL)
    {
        OutputDebugString("ファイルが読み込めません");
        OutputDebugString("ファイルを生成します");
        file_write();
    }
    else
    {
        for (i = 0; i < RANKING_MAX; i++)
        {
            fscanf_s(fp, "%2d, %[^,], %10d\n", &Ranking_Data[i].rank,
Ranking_Data[i].name, RANKING_NAME_LEN, &Ranking_Data[i].score);
        }

        fclose(fp);
    }
}

/*****
* ランキング画面：ファイル書き込み処理
* 引 数：なし
* 戻り値：なし
*****/
void file_write(void)
{
    FILE* fp = NULL;
    int i;

    OutputDebugString("ファイルを書き込みます");
    fopen_s(&fp, RANKING_FILE, "w");

    if (fp == NULL)
    {
        OutputDebugString("ファイルが書き込めません");
    }
}

```

```

    }
    else
    {
        for (i = 0; i < RANKING_MAX; i++)
        {
            fprintf(fp, "%2d,%[^,],%10d\n", Ranking_Data[i].rank,
Ranking_Data[i].name, Ranking_Data[i].score);
        }

        fclose(fp);
    }
}

```

```

/*****
* ランキング画面：ランキングソート処理
* 引 数：なし
* 戻り値：なし
*****/

```

```

void ranking_sort(void)
{
    int i, j;          // ループカウンタ
    T_RANKING tmp;     // 退避領域

    // 一番下のスコアを更新する
    Ranking_Data[RANKING_MAX - 1] = New_Score;

    // データのソートを行う
    for (i = 0; i < RANKING_MAX; i++)
    {
        for (j = i + 1; j < RANKING_MAX; j++)
        {
            if (Ranking_Data[i].score < Ranking_Data[j].score)
            {
                tmp = Ranking_Data[i];
                Ranking_Data[i] = Ranking_Data[j];
                Ranking_Data[j] = tmp;
            }
        }
    }

    // 順位を上からふっていく
    for (i = 0; i < RANKING_MAX; i++)
    {
        Ranking_Data[i].rank = i + 1;
    }
}

```

```

    // ファイルに書き込みを行う
    file_write();
}

/*****
* ランキング画面：名前入力処理
* 引 数：なし
* 戻り値：なし
*****/
void ranking_input_name(void)
{
    int c;

    // カーソル操作処理
    if (GetButtonDown(XINPUT_BUTTON_DPAD_LEFT) == TRUE)
    {
        if (Cursor.x > 0)
        {
            Cursor.x--;
        }
    }
    if (GetButtonDown(XINPUT_BUTTON_DPAD_RIGHT) == TRUE)
    {
        if (Cursor.x < 12)
        {
            Cursor.x++;
        }
    }
    if (GetButtonDown(XINPUT_BUTTON_DPAD_UP) == TRUE)
    {
        if (Cursor.y > 0)
        {
            Cursor.y--;
        }
    }
    if (GetButtonDown(XINPUT_BUTTON_DPAD_DOWN) == TRUE)
    {
        if (Cursor.y < 4)
        {
            Cursor.y++;
        }
    }

    // 文字を選択する
    if (GetButtonDown(XINPUT_BUTTON_B) == TRUE)
    {
        if (Cursor.y < 2)

```

```

    {
        c = 'a' + Cursor.x + (Cursor.y * 13);
        New_Score.name[name_num++] = c;
    }
    else if (Cursor.y < 4)
    {
        c = 'A' + Cursor.x + ((Cursor.y - 2) * 13);
        New_Score.name[name_num++] = c;
    }
    else
    {
        if (Cursor.x < 10)
        {
            c = '0' + Cursor.x;
            New_Score.name[name_num++] = c;
        }
        else if (Cursor.x == 10)
        {
            name_num--;
            New_Score.name[name_num] = '\\0';
        }
        else
        {
            DispMode = RANKING_DISP_MODE;
            ranking_sort();
        }
    }
}

}

/*****
* ランキング画面：名前入力描画処理
* 引 数：なし
* 戻り値：なし
*****/
void ranking_input_name_draw(void)
{
    int i;

    SetFontSize(40);
    DrawFormatString(300, 150, GetColor(255, 255, 255), "名前を入力してください");

    // 選択用文字を描画
    for (i = 0; i < 26; i++)
    {

```

```
        DrawFormatString((i % 13 * 50) + 300, (i / 13 * 50) + 330, GetColor(255, 255,
255), "%-3c", 'a' + i);
        DrawFormatString((i % 13 * 50) + 300, (i / 13 * 50) + 430, GetColor(255, 255,
255), "%-3c", 'A' + i);
    }
    for (i = 0; i < 10; i++)
    {
        DrawFormatString((i % 13 * 50) + 300, (i / 13 * 50) + 530, GetColor(255, 255,
255), "%-3c", '0' + i);
    }

    DrawFormatString(300, 220, GetColor(255, 255, 255), ">%s", New_Score.name);

    SetFontSize(20);

    // 選択している文字をフォーカスしている
    DrawBox((Cursor.x * 50) + 290, (Cursor.y * 50) + 330,
        (Cursor.x * 50) + 330, (Cursor.y * 50) + 370,
        GetColor(255, 255, 255), FALSE);

}
```