

```
1 #define _CRT_SECURE_NO_WARNINGS
2 #include <stdio.h>
3 #include "DxLib.h"
4 #include "Rank.h"
5 #include "SceneMgr.h"
6 #include "Input.h"
7 #include "Game.h"
8
9
10 static int mImageHandle;    //画像ハンドル格納用変数
11 static int WaitTime;        //ランキング表示時間
12
13 struct RankingData    Ranking[RANKCOUNT];    // ランキングデータ変数宣言
14
15
16 //ランキング画面 初期化
17 void Rank_Initialize() {
18     mImageHandle = LoadGraph("images/Scene_Rank.png"); //画像のロード
19     WaitTime = 0;
20
21     Rank_Read();
22 }
23
24 //ランキング画面 終了処理
25 void Rank_Finalize() {
26     DeleteGraph(mImageHandle); //画像の解放
27 }
28
29 //ランキング画面 更新
30 void Rank_Update() {
31     if (iKeyFlg == PAD_INPUT_B || CheckHitKey(KEY_INPUT_ESCAPE) != 0) { //Bボタンが押され
        ていたら
32         SceneMgr_ChangeScene(eScene_Menu); //シーンをメニューに変更
33     }
34 }
35
36 //ランキングデータ描画
37 void Rank_Draw() {
38     //ランキング画像表示
39     DrawGraph(0, 0, mImageHandle, FALSE);
40
41     // ランキング一覧を表示
42     SetFontSize(24);
43     for (int i = 0; i < RANKCOUNT; i++) {
44         DrawFormatString(100, 200 + i * 25, GetColor(0, 0, 0), "%2d %10s %10d",
45             Ranking[i].no, Ranking[i].name, Ranking[i].score);
46     }
47
48     // 文字の表示(点減)
49     if (++WaitTime < 50) {
50         SetFontSize(16);
51         DrawString(20, 440, "EscキーまたはBボタンを押すとメニュー画面に戻ります。",
52             GetColor(136, 136, 255));
53     }
54     else if (++WaitTime > 100) {
55         WaitTime = 0;
56     }
57 }
58 //ランキング入力処理
```

```
59 void Rank_Input()
60 {
61     //ゲームスコアとランキングスコアの比較
62     if (Game_GetScore() < Ranking[(RANKCOUNT-1)].score) {
63         SceneMgr_ChangeScene(eScene_Menu); //シーンをメニューに変更
64     }
65     else { //ランキング入力画面
66         //ランキング画像表示
67         DrawGraph(0, 0, mImageHandle, FALSE);
68
69         // フォントサイズの設定
70         SetFontSize(20);
71
72         // 名前入力指示文字列の描画
73         DrawString(150, 240, "ランキングに登録します", GetColor(0, 0, 0));
74         DrawString(150, 270, "名前を英字(10文字以内)で入力してください", GetColor(0, 0, 0));
75
76         // 名前の入力
77         DrawString(150, 310, "> ", GetColor(0, 0, 0));
78         DrawBox(160, 305, 300, 335, GetColor(0, 0, 0), TRUE);
79         KeyInputSingleCharString(170, 310, 10, Ranking[(RANKCOUNT - 1)].name, FALSE);
80
81         Ranking[(RANKCOUNT - 1)].score = Game_GetScore(); //ランキングデータの最後にスコアを登録
82         Ranking[(RANKCOUNT - 1)].no = RANKCOUNT; //ランキングNo. は最後にしておく
83
84         Rank_Sort(); // ランキング並べ替え
85         Rank_Save(); // ランキングデータの保存
86
87         SceneMgr_ChangeScene(eScene_Rank); //ランキング表示画面へ移行
88     }
89 }
90
91 //ランキング並べ替え
92 void Rank_Sort()
93 {
94     int i, j;
95     RankingData work;
96
97     // 選択法ソート
98     for (i = 0; i < RANKCOUNT; i++) {
99         for (j = i + 1; j < RANKCOUNT; j++) {
100             if (Ranking[i].score <= Ranking[j].score) {
101                 work = Ranking[i];
102                 Ranking[i] = Ranking[j];
103                 Ranking[j] = work;
104             }
105         }
106     }
107
108     // 順位付け
109     for (i = 0; i < RANKCOUNT; i++) {
110         Ranking[i].no = 1;
111     }
112     // 得点と同じ場合は、同じ順位とする
113     // 同順位があった場合の次の順位はデータ個数が加算された順位とする
114     for (i = 0; i < RANKCOUNT; i++) {
115         for (j = i + 1; j < RANKCOUNT; j++) {
116             if (Ranking[i].score > Ranking[j].score) {
```

```
117         Ranking[j].no++;
118     }
119 }
120 }
121 }
122
123 //ランキングデータの保存
124 void Rank_Save()
125 {
126     FILE* fp;
127
128     // ファイルオープン
129     fopen_s(&fp, RANKING_FILE, "w");
130     if (fp != NULL) {
131         // ランキングデータ配列データを書き込む
132         for (int i = 0; i < RANKCOUNT; i++) {
133             fprintf(fp, "%s %d %d\n", Ranking[i].name, Ranking[i].no, Ranking[i].score);
134         }
135
136         //ファイルクローズ
137         fclose(fp);
138     }
139     else {
140         //エラー処理（必要に応じて作成）
141         // printf("Ranking Data Error\n");
142     }
143 }
144
145 //ランキングデータ読み込み
146 void Rank_Read()
147 {
148     FILE* fp;
149     int i;
150
151     //ファイルオープン
152     fopen_s(&fp, RANKING_FILE, "r");
153     if (fp != NULL) {
154         //ランキングデータ配列データを読み込む
155         for (int i = 0; i < RANKCOUNT; i++) {
156             fscanf_s(fp, "%s %d %d", Ranking[i].name, 10, &Ranking[i].no, &Ranking[i].score);
157         }
158         fclose(fp);
159     }
160     else { //ファイルがオープンできないor読み込めない
161         //エラー処理（必要に応じて作成）
162         // printf("Ranking Data Error\n");
163
164         // ランキングデータの初期化
165         for (i = 0; i < RANKCOUNT; i++) {
166             Ranking[i].no = (i+1);
167             sprintf_s(Ranking[i].name, "...");
168             Ranking[i].score = 0;
169         }
170     }
171 }
172
173 //ランキング画面へ移行するかチェック
174 void Rank_InputUpdate() {
175     int nowScore = Game_GetScore();
```

```
176     int rankScore = Ranking[(RANKCOUNT - 1)].score;
177
178     if (nowScore < rankScore) {          //ゲームスコアとランキングの一番低いスコアとの比較
179         //スコアがランキング内にも入らない→メニューへ
180         SceneMgr_ChangeScene(eScene_Menu); //メニュー画面へ移行
181     }
182 }
```