

```
1 // *** +2lines
2 #define _CRT_SECURE_NO_WARNINGS
3 #include <string.h>
4 #include "DxLib.h"
5 #include "SceneMgr.h"
6 #include "Input.h"
7 #include "Game.h"
8 #include "Player.h"
9 #include "Enemy.h"
10 #include "stage.h"
11 #include "Effect.h"
12 #include "Rank.h"
13
14
15 /*****
16 ** Game.cpp内変数 **
17 *****/
18 static int mImageHandle; //画像ハンドル格納用変数
19 static int mImageBackground; //画像ハンドル格納用変数
20 static int mImageHp; //画像ハンドル格納用変数
21
22 static int gameScore; //ゲームスコア
23 static int mGameTime; //ゲーム経過時間
24 static int mgameStatus; //ゲーム状態
25
26 static int mScroll1; //背景スクロール用Y座標1
27 static int mScroll2; //背景スクロール用Y座標2
28
29
30 /*****
31 * ゲーム 初期化
32 * 引 数 : なし
33 * 戻り値 : なし
34 *****/
35 void Game_Initialize() {
36     mImageHandle = LoadGraph("images/Scene_Game.png"); //画像のロード
37     mImageBackground = LoadGraph("images/stage1.png"); //画像のロード
38     // *** +1line
39     mImageHp = LoadGraph("images/select.png"); //画像のロード
40
41     // ゲーム中BGM再生
42     PlayMusic("sounds/yukinagori.mp3", DX_PLAYTYPE_LOOP);
43
44     // 背景座標設定
45     mScroll1 = 0;
46     mScroll2 = -960;
47
48     Player_Initialize();
49     Enemy_Initialize();
50     Effect_Initialize();
51
52     mGameTime = 0;
53     mgameStatus = true;
54 }
55
56
57 /*****
58 * ゲーム 終了処理
59 * 引 数 : なし
60 *****/
```

```
61  * 戻り値: なし
62  *****/
63 void Game_Finalize() {
64     DeleteGraph(mImageHandle);    //画像の解放
65
66     // ゲーム中BGM停止
67     StopMusic();
68
69     Player_Finalize();
70     Enemy_Finalize();
71     Effect_Finalize();
72
73 }
74
75
76 // *** 1-function change
77 /*****/
78 * ゲーム 更新
79 * 引 数: なし
80 * 戻り値: なし
81 *****/
82 void Game_Update() {
83     //YボタンまたはESCボタンでメニューに戻る
84     if (iKeyFlg == PAD_INPUT_4 || (CheckHitKey(KEY_INPUT_ESCAPE) != 0)) {
85         SceneMgr_ChangeScene(eScene_Menu); //シーンをメニューに変更
86     }
87
88     // ゲームプレイ中なら各処理をアップデート
89     if (mPlayer.hp > 0) {
90         Player_Update();
91         Enemy_Update();
92         Effect_Update();
93
94         mGameTime++;
95
96         Stage1();
97     }
98 }
99
100
101 // *** 1-function change
102 /*****/
103 * ゲーム 描画処理
104 * 引 数: なし
105 * 戻り値: なし
106 *****/
107 void Game_Draw() {
108
109     BackScroll();
110
111     if (mPlayer.hp > 0) {
112         Player_Draw();
113         Effect_Draw();
114         Enemy_Draw();
115     }
116     // ゲーム画面 (フレーム描画)
117     DrawGraph(0, 0, mImageHandle, TRUE);
118
119     if (mgameStatus == FALSE) {
120         GameClear();
121     }
```

```

121     }
122
123     if (mPlayer.hp <= 0) {
124         GameOver();
125     }
126     else {
127         ScoreDraw();
128     }
129 }
130
131
132 // *** 1-function change
133 /*****
134  * ゲームクリア処理
135  * 引 数：なし
136  * 戻り値：なし
137  *****/
138 void GameClear(void)
139 {
140     const char* str1 = "---- Xキーを押してランキング入力画面へ ----";
141     const char* str2 = "---- Xキーを押してタイトルへ戻る ----";
142     char str3[48];
143
144     // プレイのスコア設定
145     gameScore = mPlayer.score;
146
147     //ランキング1位との比較 ランキング内ならランキング入力画面へ誘導
148     if (Ranking[0].score < gameScore) {
149         if (iKeyFlg == PAD_INPUT_2) { //Xキーが押されていたら
150             SceneMgr_ChangeScene(eScene_RankInput); //ランキング画面入力へ
151         }
152
153         strcpy(str3, str1);
154     }
155     //ランキング外
156     else {
157         if (iKeyFlg == PAD_INPUT_2) { //Xキーが押されていたら
158             SceneMgr_ChangeScene(eScene_Menu); //シーンをゲーム画面に変更
159         }
160
161         strcpy(str3, str2);
162     }
163     mgameStatus = true;
164
165     // フォントサイズの設定
166     SetFontSize(40);
167     DrawString(20, 220, "GAME CLEAR", 0xffffffff, 0);
168
169     // フォントサイズの設定
170     SetFontSize(20);
171     DrawString(20, 440, str1, 0xffffffff, 0);
172 }
173
174
175 // *** 1-function change
176 /*****
177  * ゲームオーバー処理
178  * 引 数：なし
179  * 戻り値：なし
180  *****/

```

```

181 void GameOver(void)
182 {
183     const char* str1 = "---- Xキーを押してランキング入力画面へ ----";
184     const char* str2 = "---- Xキーを押してタイトルへ戻る ----";
185     char str3[48];
186
187     // プレイのスコア設定
188     gameScore = mPlayer.score;
189
190     if (Ranking[0].score < gameScore) {
191         if (iKeyFlg == PAD_INPUT_2) { //Xキーが押されていたら
192             SceneMgr_ChangeScene(eScene_RankInput); //ランキング画面入力へ
193         }
194
195         strcpy( str3, str1 );
196     }
197     else {
198         if (iKeyFlg == PAD_INPUT_2) { //Xキーが押されていたら
199             SceneMgr_ChangeScene(eScene_Menu); //シーンをゲーム画面に変更
200         }
201
202         strcpy( str3, str2 );
203     }
204     mgameStatus = true;
205
206     // フォントサイズの設定
207     SetFontSize(40);
208     DrawString(35, 220, "GAME OVER", 0xffffffff, 0);
209
210     // フォントサイズの設定
211     SetFontSize(20);
212     DrawString(35, 440, str3, 0xffffffff, 0);
213 }
214
215
216 /*****
217  * 背景スクロール等
218  * 引 数: なし
219  * 戻り値: なし
220  *****/
221 void BackScroll()
222 {
223     //背景表示
224     mScroll1 = mScroll1 + 5;
225     if (mScroll1 >= 960) { mScroll1 = -960; }
226     mScroll2 = mScroll2 + 5;
227     if (mScroll2 >= 960) { mScroll2 = -960; }
228
229     DrawGraph(0, mScroll1, mImageBackground, TRUE);
230     DrawGraph(0, mScroll2, mImageBackground, TRUE);
231 }
232
233
234 /*****
235  * スコアの表示
236  * 引 数: なし
237  * 戻り値: なし
238  *****/
239 void ScoreDraw()
240 {

```

```
241 // フォントサイズの設定
242 SetFontSize(20);
243
244 // スコア表示
245 DrawFormatString(496, 45, GetColor(0, 0, 0), "%010d", Ranking[0].score);
246 DrawFormatString(494, 43, GetColor(255, 255, 255), "%010d", Ranking[0].score);
247 DrawFormatString(496, 98, GetColor(0, 0, 0), "%010d", mPlayer.score);
248 DrawFormatString(494, 96, GetColor(255, 255, 255), "%010d", mPlayer.score);
249
250 //残機表示
251 for (int i = 0; i < mPlayer.hp; i++) {
252     DrawGraph(455 + i * 25, 255, mImageHp, TRUE);
253 }
254 }
255
256
257
258
259 /*****
260 * ゲームスコア取得
261 * 引 数: なし
262 * 戻り値: gaemScore
263 *****/
264 int Game_GetScore() {
265     return gameScore;
266 }
267
268
269 /*****
270 * ゲーム時間取得
271 * 引 数: なし
272 * 戻り値: mGameTime
273 *****/
274 int Game_GetmGameTime() {
275     return mGameTime;
276 }
277
278
279 /*****
280 * ゲーム状況セット
281 * 引 数: flg
282 * 戻り値: mGameTime
283 *****/
284 void Game_SetmgameStatus( int flg ) {
285     mgameStatus = flg;
286 }
287
```