

```
1 #include "DxLib.h"
2 #include "SceneMgr.h"
3 #include "Input.h"
4 #include "Game.h"
5 #include "Player.h"
6 #include "Enemy.h"
7 #include "stage.h"
8 #include "Effect.h"
9 // *** +1line
10 #include "Rank.h"
11
12
13 /*****
14 ** Game.cpp内変数 **
15 *****/
16 static int mImageHandle; //画像ハンドル格納用変数
17 static int mImageBackground; //画像ハンドル格納用変数
18 // *** +1line
19 static int mImageHp; //画像ハンドル格納用変数
20
21 static int gameScore; //ゲームスコア
22 static int mGameTime; //ゲーム経過時間
23 static int mgameStatus; //ゲーム状態
24
25 static int mScroll1; //背景スクロール用Y座標1
26 static int mScroll2; //背景スクロール用Y座標2
27
28
29 /*****
30 * ゲーム 初期化
31 * 引 数: なし
32 * 戻り値: なし
33 *****/
34 void Game_Initialize() {
35     mImageHandle = LoadGraph("images/Scene_Game.png"); //画像のロード
36     mImageBackground = LoadGraph("images/stage1.png"); //画像のロード
37     // *** +1line
38     mImageHp = LoadGraph("images/select.png"); //画像のロード
39
40     // ゲーム中BGM再生
41     PlayMusic("sounds/yukinagori.mp3", DX_PLAYTYPE_LOOP);
42
43     // 背景座標設定
44     mScroll1 = 0;
45     mScroll2 = -960;
46
47     Player_Initialize();
48     Enemy_Initialize();
49     Effect_Initialize();
50
51     mGameTime = 0;
52     mgameStatus = true;
53 }
54
55
56
57 /*****
58 * ゲーム 終了処理
59 * 引 数: なし
60 * 戻り値: なし
61 *****/
```

```
61  *****/
62 void Game_Finalize() {
63     DeleteGraph(mImageHandle);    //画像の解放
64
65     // ゲーム中BGM停止
66     StopMusic();
67
68     Player_Finalize();
69     Enemy_Finalize();
70     Effect_Finalize();
71 }
72 }
73
74
75 /*****/
76 * ゲーム 更新
77 * 引 数: なし
78 * 戻り値: なし
79 *****/
80 void Game_Update() {
81     //YボタンまたはESCボタンでメニューに戻る
82     if (iKeyFlg == PAD_INPUT_4 || (CheckHitKey(KEY_INPUT_ESCAPE) != 0)) {
83         SceneMgr_ChangeScene(eScene_Menu); //シーンをメニューに変更
84     }
85
86     Player_Update();
87     Enemy_Update();
88     Effect_Update();
89
90     mGameTime++;
91
92     Stage1();
93 }
94 }
95
96
97 /*****/
98 * ゲーム 描画処理
99 * 引 数: なし
100 * 戻り値: なし
101 *****/
102 void Game_Draw() {
103
104     BackScroll();
105     Player_Draw();
106     Effect_Draw();
107     Enemy_Draw();
108
109     // ゲーム画面 (フレーム描画)
110     DrawGraph(0, 0, mImageHandle, TRUE);
111
112     if (mgameStatus == FALSE) {
113         GameClear();
114     }
115     if (mPlayer.hp < 0) {
116         GameOver();
117     }
118     // *** +3lines
119     else {
120         ScoreDraw();
```

```
121     }
122
123 }
124
125
126 /*****
127  * ゲームクリア処理
128  * 引 数：なし
129  * 戻り値：なし
130  *****/
131 void GameClear(void)
132 {
133     if (iKeyFlg == PAD_INPUT_2) { //Xキーが押されていたら
134         SceneMgr_ChangeScene(eScene_Menu); //シーンをゲーム画面に変更
135         mgameStatus = true;
136     }
137
138     // フォントサイズの設定
139     SetFontSize(40);
140     DrawString(20, 220, "GAME CLEAR", 0xffffffff, 0);
141
142     // フォントサイズの設定
143     SetFontSize(20);
144     DrawString(20, 440, "---- Xキーを押してタイトルへ戻る ----", 0xffffffff, 0);
145 }
146
147
148
149 /*****
150  * ゲームオーバー処理
151  * 引 数：なし
152  * 戻り値：なし
153  *****/
154 void GameOver(void)
155 {
156     if (iKeyFlg == PAD_INPUT_2) { //Xキーが押されていたら
157         SceneMgr_ChangeScene(eScene_Menu); //シーンをゲーム画面に変更
158     }
159
160     // *** +6lines
161     // フォントサイズの設定
162     SetFontSize(40);
163     DrawString(35, 220, "GAME OVER", 0xffffffff, 0);
164
165     // フォントサイズの設定
166     SetFontSize(20);
167     DrawString(150, 450, "---- Xキーを押してタイトルへ戻る ----", 0xffffffff, 0);
168 }
169
170
171
172 /*****
173  * 背景スクロール等
174  * 引 数：なし
175  * 戻り値：なし
176  *****/
177 void BackScroll()
178 {
179     //背景表示
180     mScroll1 = mScroll1 + 5;
```

```
181     if (mScroll1 >= 960) { mScroll1 = -960; }
182     mScroll2 = mScroll2 + 5;
183     if (mScroll2 >= 960) { mScroll2 = -960; }
184
185     DrawGraph(0, mScroll1, mImageBackground, TRUE);
186     DrawGraph(0, mScroll2, mImageBackground, TRUE);
187
188 }
189
190
191 // *** +23lines(1-function)
192 /*****
193  * スコアの表示
194  * 引 数: なし
195  * 戻り値: なし
196  *****/
197 void ScoreDraw()
198 {
199     // フォントサイズの設定
200     SetFontSize(20);
201
202     // スコア表示
203     DrawFormatString(496, 45, GetColor(0, 0, 0), "%010d", Ranking[0].score);
204     DrawFormatString(494, 43, GetColor(255, 255, 255), "%010d", Ranking[0].score);
205     DrawFormatString(496, 98, GetColor(0, 0, 0), "%010d", mPlayer.score);
206     DrawFormatString(494, 96, GetColor(255, 255, 255), "%010d", mPlayer.score);
207
208     //残機表示
209     for (int i = 0; i < mPlayer.hp; i++) {
210         DrawGraph(455 + i * 25, 255, mImageHp, TRUE);
211     }
212 }
213
214
215
216
217 /*****
218  * ゲームスコア取得
219  * 引 数: なし
220  * 戻り値: gameScore
221  *****/
222 int Game_GetScore() {
223     //デバッグ用
224     gameScore = 50;
225     return gameScore;
226 }
227
228
229 /*****
230  * ゲーム時間取得
231  * 引 数: なし
232  * 戻り値: mGameTime
233  *****/
234 int Game_GetmGameTime() {
235     return mGameTime;
236 }
237
238
239 /*****
240  * ゲーム状況セット
```

```
241 * 引 数 : flg
242 * 戻り値 : mGameTime
243 *****/
244 void Game_SetmgameStatus( int flg ) {
245     mgameStatus = flg;
246 }
247
```