

# C言語初めてのゲーム(走り抜けジャンプ)-part2

## 対象

---

前回の[C言語初めてのゲーム\(走る抜けジャンプ\)](#)を終えている人が対象となっています。終えてない人は以下のリンクから飛んでやってください。  
[C言語初めてのゲーム\(走り抜けジャンプ\)](#)

## 構成

---

[No.4 エンディング](#)

[No.4-1 準備](#)

[No.4-2 画像と操作と切り替え](#)

## 本編

---

### No.4 エンディング

---

ゲームをプレイした後のエンディングです。

今回はクリアとゲームオーバーを分けずにエンディングとするので至極簡単なものになります。

#### No.4-1 準備

---

エンディングもタイトルと同じように書けばいいと思いますがそうはいきませんでした。

タイトルとエンディングが同時に処理されると大変なことになりますよね。また、同様にゲーム中にタイトルとかエンディングが実行されていると困りますよね。

なので、シーンを管理する変数を用意します。今回はシーンに関しては簡単に済むようにします。あと【int型】を使っていますが、見やすさや分かりやすさを重視して【enum型】を使うことを推奨します。分かる人はそこだけ書き換えてみてください。

```
int scene;           // 現在のシーン
```

それが済めばあとはタイトル同様です。エンディングに必要なものはタイトルに比べると少なく済みます。あと今回は準備のみなので描画はしません。描画は次にまとめます。

タイトルと同じように【描画】【操作】と分けて扱います。関数名が違うだけで必要なものと役割は変わらないので説明は端折ります。

```
void End_Draw(void);      // エンディングの描画の中身を管理
void End_Process(void);   // エンディングの操作の中身を管理

void End_Update(void);    // エンディングの描画と操作の実体を管理
void Initialize_End(void); // エンディングの要素の初期化
```

```
void Delete_End(void);          // エンディングの要素の削除
```

それらを組み込んであげたソースコードが以下になります。

```
1. #include "DxLib.h"
2.
3. // 初期化
4. void Initialize_Title(void);    // タイトルの要素の初期化
5. void Initialize_End(void);      // エンディングの要素の初期化
6.
7. // 削除
8. void Delete_Title(void);        // タイトルの要素の削除
9. void Delete_End(void);          // エンディングの要素の削除
10.
11. // タイトルの実体
12. void Title_Update(void);        // タイトル描画と操作の実体を管理
13. void Title_Draw(void);         // タイトルの描画の中身を管理
14. void Title_Process(void);      // タイトルの操作の中身を管理
15.
16. // エンディングの実体
17. void End_Update(void);          // エンディングの描画と操作の実体を管理
18. void End_Draw(void);           // エンディングの描画の中身を管理
19. void End_Process(void);        // エンディングの操作の中身を管理
20.
21. // タイトル要素
22. // 描画
23. int gr_title_back;             // 背景
24. int gr_title_cursor;           // カーソル絵
25. int gr_title_start;           // 始めるアイコン画像
26. int gr_title_end;             // 終わるアイコン画像
27. // 操作
28. int title_cursor_y;            // カーソル動作
29.
30. // シーン
31. int scene;
32.
33. // 終了
34. bool endflag;
35.
36. int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
37. {
38.     ChangeWindowMode(TRUE);      // ウィンドウモード変更
39.     DxLib_Init();                // 初期化
40.     SetDrawScreen( DX_SCREEN_BACK ); // 裏画面設定
```

```
41.
42.     scene = 0;
43.     endflag = false;
44.
45.     Initialize_Title();           // タイトルの初期化の呼び出し
46.     Initialize_End();            // エンディングの初期化の呼び出し
47.
48.     while( ScreenFlip()==0 && ProcessMessage()==0 && ClearDrawScreen()==0 && CheckHitKey(KEY_INPUT_ESCAPE)==0 && endflag == false)
49.     {
50.         if(scene == 0)
51.         {
52.             Title_Update();       // タイトル
53.         }
54.         else if(scene == 1)
55.         {
56.             End_Update();         // エンディング
57.         }
58.     }
59.
60.     Delete_Title();              // タイトルの削除
61.     Delete_End();                // エンディングの削除
62.
63.     DxLib_End(); // DXライブラリ終了処理
64.     return 0;
65. }
66.
67. // タイトルの初期化
68. void Initialize_Title()
69. {
70.     // 描画
71.     gr_title_back = LoadGraph("media¥¥title_background.png");
72.     gr_title_cursor = LoadGraph("media¥¥title_cursor.png");
73.     gr_title_start = LoadGraph("media¥¥title_start.png");
74.     gr_title_end = LoadGraph("media¥¥title_end.png");
75.
76.     title_cursor_y = 0;
77. }
78.
79. // タイトルの削除
80. void Delete_Title()
81. {
82.     DeleteGraph(gr_title_back);
83.     DeleteGraph(gr_title_cursor);
84.     DeleteGraph(gr_title_start);
85.     DeleteGraph(gr_title_end);
86. }
```

```
87.
88. // タイトルのすべて
89. void Title_Update()
90. {
91.     Title_Draw();
92.     Title_Process();
93. }
94.
95. // タイトルの描画
96. void Title_Draw()
97. {
98.     DrawGraph(0, 0, gr_title_back, false);
99.     DrawGraph(240, 300 + (50 * title_cursor_y), gr_title_cursor, true);
100.    DrawGraph(290, 300, gr_title_start, true);
101.    DrawGraph(290, 350, gr_title_end, true);
102. }
103.
104. // タイトルの操作
105. void Title_Process()
106. {
107.     if(CheckHitKey(KEY_INPUT_UP) == 1)
108.     {
109.         if(title_cursor_y != 0)
110.         {
111.             title_cursor_y++;
112.         }
113.     }
114.     if(CheckHitKey(KEY_INPUT_DOWN) == 1)
115.     {
116.         if(title_cursor_y != 1)
117.         {
118.             title_cursor_y--;
119.         }
120.     }
121.
122.     if(CheckHitKey(KEY_INPUT_Z) == 1)
123.     {
124.         if(title_cursor_y == 0)
125.         {
126.
127.         }
128.         else if(title_cursor_y == 1)
129.         {
130.             endflag = true;
131.         }
132.     }
```

```
133. }  
134.  
135. // エンディングの初期化  
136. void Initialize_End()  
137. {  
138.  
139. }  
140.  
141. // エンディングの削除  
142. void Delete_End()  
143. {  
144.  
145. }  
146.  
147. // エンディングのすべて  
148. void End_Update()  
149. {  
150.     End_Draw();  
151.     End_Process();  
152. }  
153.  
154. // エンディングの描画  
155. void End_Draw()  
156. {  
157.  
158. }  
159.  
160. // エンディングの操作  
161. void End_Process()  
162. {  
163.  
164. }
```

今回は前のタイトルの関数に加えて【scene】という変数を追加しました。これは名前の通りシーンを切り替えるものです。  
今は特に切り替えを実装してませんが今後必要になります。ただ今させても何もないエンディングに移るだけなのでさせてません。

次で画像が追加されるので実装させます。といっても簡単な感じにしますけどね。どこかのタイミングで言ったかもしれませんがちゃんとしたシークエンス処理は違う場所で細かくやります。

#### No.4-2 画像と操作と切り替え

---

エンディングに必要な画像と操作の実装をします。簡単なものとします。

エンディングでの操作はタイトルへ戻る操作のみとします。これを行うことでクリアしても何度もプレイできるようにします。  
先ほどの【scene】がそれなので特に変数の追加などはありません。

基本的にエンディングではこのような動作のみとします。人によってはクリアした後にタイトルへ戻るとおまけモードが追加されてるとかいう処理をしてもいいと思います。

次にエンディングの画像に関することですが、背景だけでいいです。クリアーといった感じの画像のみとします。

なので今回は特に特別な変数や関数などはないのですぐにソースコードを出します。

```
1. #include "DxLib.h"
2.
3. // 初期化
4. void Initialize_Title(void);    // タイトルの要素の初期化
5. void Initialize_End(void);      // エンディングの要素の初期化
6.
7. // 削除
8. void Delete_Title(void);        // タイトルの要素の削除
9. void Delete_End(void);          // エンディングの要素の削除
10.
11. // タイトルの実体
12. void Title_Update(void);        // タイトル描画と操作の実体を管理
13. void Title_Draw(void);          // タイトルの描画の中身を管理
14. void Title_Process(void);       // タイトルの操作の中身を管理
15.
16. // エンディングの実体
17. void End_Update(void);          // エンディングの描画と操作の実体を管理
18. void End_Draw(void);           // エンディングの描画の中身を管理
19. void End_Process(void);         // エンディングの操作の中身を管理
20.
21. // タイトル要素
22. // 描画
23. int gr_title_back;             // 背景
24. int gr_title_cursor;           // カーソル絵
25. int gr_title_start;           // 始めるアイコン画像
26. int gr_title_end;              // 終わるアイコン画像
27. // 操作
28. int title_cursor_y;            // カーソル動作
29.
30. // エンディング要素
31. // 描画
32. int gr_end_back;               // 背景
33.
34. // シーン
35. int scene;
36.
```

```
37. // 終了
38. bool endflag;
39.
40. int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
41. {
42.     ChangeWindowMode(TRUE);                // ウィンドウモード変更
43.     DxLib_Init();                          // 初期化
44.     SetDrawScreen( DX_SCREEN_BACK );       // 裏画面設定
45.
46.     scene = 0;
47.     endflag = false;
48.
49.     Initialize_Title();                    // タイトルの初期化の呼び出し
50.     Initialize_End();                     // エンディングの初期化の呼び出し
51.
52.     while( ScreenFlip()==0 && ProcessMessage()==0 && ClearDrawScreen()==0 && CheckHitKey(KEY_INPUT_ESCAPE)==0 && endflag == false)
53.     {
54.         if(scene == 0)
55.         {
56.             Title_Update();                // タイトル
57.         }
58.         else if(scene == 1)
59.         {
60.             End_Update();                  // エンディング
61.         }
62.     }
63.
64.     Delete_Title();                        // タイトルの削除
65.     Delete_End();                          // エンディングの削除
66.
67.     DxLib_End(); // DXライブラリ終了処理
68.     return 0;
69. }
70.
71. // タイトルの初期化
72. void Initialize_Title()
73. {
74.     // 描画
75.     gr_title_back = LoadGraph("media\\title_background.png");
76.     gr_title_cursor = LoadGraph("media\\title_cursor.png");
77.     gr_title_start = LoadGraph("media\\title_start.png");
78.     gr_title_end = LoadGraph("media\\title_end.png");
79.
80.     title_cursor_y = 0;
81. }
82.
```

```
83. // タイトルの削除
84. void Delete_Title()
85. {
86.     DeleteGraph(gr_title_back);
87.     DeleteGraph(gr_title_cursor);
88.     DeleteGraph(gr_title_start);
89.     DeleteGraph(gr_title_end);
90. }
91.
92. // タイトルのすべて
93. void Title_Update()
94. {
95.     Title_Draw();
96.     Title_Process();
97. }
98.
99. // タイトルの描画
100. void Title_Draw()
101. {
102.     DrawGraph(0, 0, gr_title_back, false);
103.     DrawGraph(240, 300 + (50 * title_cursor_y), gr_title_cursor, true);
104.     DrawGraph(290, 300, gr_title_start, true);
105.     DrawGraph(290, 350, gr_title_end, true);
106. }
107.
108. // タイトルの操作
109. void Title_Process()
110. {
111.     if(CheckHitKey(KEY_INPUT_UP) == 1)
112.     {
113.         if(title_cursor_y != 0)
114.         {
115.             title_cursor_y++;
116.         }
117.     }
118.     if(CheckHitKey(KEY_INPUT_DOWN) == 1)
119.     {
120.         if(title_cursor_y != 1)
121.         {
122.             title_cursor_y--;
123.         }
124.     }
125.
126.     if(CheckHitKey(KEY_INPUT_Z) == 1)
127.     {
128.         if(title_cursor_y == 0)
```



```
129.         {
130.             scene = 1;
131.         }
132.         else if(title_cursor_y == 1)
133.         {
134.             endflag = true;
135.         }
136.     }
137. }
138.
139. // エンディングの初期化
140. void Initialize_End()
141. {
142.     // 描画
143.     gr_end_back = LoadGraph("media\\end_background.png");
144. }
145.
146. // エンディングの削除
147. void Delete_End()
148. {
149.     DeleteGraph(gr_end_back);
150. }
151.
152. // エンディングのすべて
153. void End_Update()
154. {
155.     End_Draw();
156.     End_Process();
157. }
158.
159. // エンディングの描画
160. void End_Draw()
161. {
162.     DrawGraph(0, 0, gr_end_back, false);
163. }
164.
165. // エンディングの操作
166. void End_Process()
167. {
168.     if(CheckHitKey(KEY_INPUT_Z) == 1)
169.     {
170.         scene = 0;
171.     }
172. }
```

変更した赤字を見ればわかると思いますが、タイトルのほうも少し変更しました。というのもシーンをエンディングとタイトルを行き来できるように少し弄りました。

今はZキーの入力でシーンが簡単に切り替わるようにしています。

ほかには特に説明はないと思うので次へ行きます。ですが、長くなったので次のページへ移ります。

### [C言語初めてのゲーム\(走り抜けジャンプ\)-part3](#)

Copyright (C) 2002-2009 Xiph.org Foundation Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: - Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. - Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. - Neither the name of the Xiph.org Foundation nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. libpng Copyright (C) 2004, 2006-2012 Glenn Randers-Pehrson. zlib Copyright (C) 1995-2012 Jean-loup Gailly and Mark Adler. DX Library Copyright (C) 2001-2014 Takumi Yamada.