

C言語初めてのゲーム(走り抜けジャンプ)

対象

描画と音の再生をある程度使えた人が対象となります。

内容

走り抜けのジャンプというある程度簡単なゲームを通して一通りのゲームを作り方を勉強します。C言語を終えていることを前提として説明するのでわからない部分が出たら学びなおしてください。

また、前回などで説明したDxlibの関数に関しては説明しません。

※作者が疲れたのもっと単調なものに変わってるかもしれません。

構成

[No.1 本編についての説明](#)

[No.2 おまじない](#)

[No.3 タイトル](#)

[No.3-1 準備および背景描画](#)

[No.3-2 選択肢のカーソルとボタン](#)

本編

No.1 本編についての説明

今回ですが、走り抜けのゲームを作ります。最低限のキャラの動作とあたり判定、敵の生成や疑似スクロールを勉強します。

といってもこれから書きながらソースコードを組むので少し変わるかもしれません。まあ、簡単なことには変わりはないので身構えずにだらだらとやってください。

ただどんな説明を行えばいいのか私も初心者ですのでいらん事書いてたりちょっと端折ってるかもしれません。

この構成ですが、必要な要素を伝えた後にそれを含んだソースコードを見せるといったやり方で説明します。

No.2 おまじない

おまじないです。とりえずこれを書いてください。



```
1. #include "DxLib.h"
2.
3. int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
4. {
5.     ChangeWindowMode(TRUE);           // ウィンドウモード変更
6.     DxLib_Init();                     // 初期化
7.     SetDrawScreen( DX_SCREEN_BACK ); // 裏画面設定
8.
9.     while( ScreenFlip()==0 && ProcessMessage()==0 && ClearDrawScreen()==0 && CheckHitKey(KEY_INPUT_ESCAPE)==0)
10.    {
11.
12.    }
13.
14.    DxLib_End(); // DXライブラリ終了処理
15.    return 0;
16. }
```

もう言うことない。説明なし。次行こう。

No.3 タイトル

ゲームを起動して最初に出会うやつ、タイトルを作ります。

タイトルなんて簡単でしょー？

いや簡単ですけど説明長そうなのでいくつかに分けて説明します。

正直プログラミングに対してある程度わかる人はNo.3の最後のソースコードでどういったことが行われているか見て学ぶだけでもいいかもしれないレベルです。

No.3-1 準備および背景描画

タイトルに必要なものの変数や関数はたくさんあります。

ですが、一気に出しても混乱するのでまずはタイトルとして実行される関数と、それだけでは味気ないので背景を描画させます。

タイトルに限らずですが、動作を管理するのに【描画】【操作】と分けた方が扱いやすいです。
ということでそれらの関数を作ります。

```
void Title_Draw(void);      // 描画の中身を管理
void Title_Process(void);   // 操作の中身を管理
```

こいつらが基本的にタイトルのほぼすべてを持ってると考えてください。
ですが、二つの関数をいちいち呼び出すのは煩わしいので私はそれらをもつ関数を作ります。

```
void Title_Update(void);      // 描画と操作の実体を管理
```

そうすることでタイトルの呼び出しはこいつだけで済みます。

管理をする関数は出来ましたが、これらはタイトルを扱う上でマイループ行われます。ですが、その中に画像のロードとか入れたくありません。ということで、初期化を一括で行う関数を作ります。

```
void Initialize_Title(void);  // タイトルの要素の初期化
```

また、使い終わった後に削除するための関数を作ります。

```
void Delete_Title(void);     // タイトルの要素の削除
```

これでタイトルの準備は終わりです。

それらを組み込んであげたソースコードが以下ようになります。

```
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.
6. // 削除
7. void Delete_Title(void);     // タイトルの要素の削除
8.
9. // タイトルの実体
10. void Title_Update(void);     // タイトルの描画と操作の実体を管理
11. void Title_Draw(void);      // タイトルの描画の中身を管理
12. void Title_Process(void);   // タイトルの操作の中身を管理
13.
14. // タイトル要素
15. int gr_title_back;         // タイトルの背景
16.
17. int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
18. {
19.     ChangeWindowMode(TRUE);  // ウィンドウモード変更
20.     DxLib_Init();            // 初期化
21.     SetDrawScreen( DX_SCREEN_BACK );  // 裏画面設定
22.
23.     Initialize_Title();      // タイトルの初期化の呼び出し
24.
25.     while( ScreenFlip()==0 && ProcessMessage()==0 && ClearDrawScreen()==0 && CheckHitKey(KEY_INPUT_ESCAPE)==0)
26.     {
27.         Title_Update();      // タイトル
```

```
28.     }
29.
30.     Delete_Title();           // タイトルの削除
31.
32.     DxLib_End(); // DXライブラリ終了処理
33.     return 0;
34. }
35.
36. // タイトルの初期化
37. void Initialize_Title()
38. {
39.     gr_title_back = LoadGraph("media¥¥title_background.png");
40. }
41.
42. // タイトルの削除
43. void Delete_Title()
44. {
45.     DeleteGraph(gr_title_back);
46. }
47.
48. // タイトルのすべて
49. void Title_Update()
50. {
51.     Title_Draw()
52.     ;
53.     Title_Process();
54. }
55. // タイトルの描画
56. void Title_Draw()
57. {
58.     DrawGraph(0, 0, gr_title_back, false);
59. }
60.
61. // タイトルの操作
62. void Title_Process()
63. {
64.     // 操作は今ないので何も書かないで構いません。
65. }
```

ここで行われているかわからないと絶望です。

No.3-2 選択肢のカーソルとボタン

今回の重要な部分はカーソルです。

カーソルはY座標を移動し、場所によって決定キーの実行結果が変わります。
それらについて必要な要素としてカーソルの画像とそれぞれのアイコンの画像とカーソルの位置を把握する変数です。用意します。

```
int gr_title_cursor;    // カーソル絵
int gr_title_start;     // 始めるアイコン画像
int gr_title_end;       // 終わるアイコン画像
// 操作
int title_cursor_y;     // カーソル動作
```

また、ゲーム中の処理はまだ作っていないので終了するときの処理だけ付け加えます。

```
// 終了
bool endflag;
```

追加要素は以上です。ですが、今までに記入してきた関数内は変更します。

```
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.
6. // 削除
7. void Delete_Title(void);        // タイトルの要素の削除
8.
9. // タイトルの実体
10. void Title_Update(void);        // タイトル描画と操作の実体を管理
11. void Title_Draw(void);          // タイトルの描画の中身を管理
12. void Title_Process(void);       // 操作の中身を管理
13.
14. // タイトル要素
15. // 描画
16. int gr_title_back;             // 背景
17. int gr_title_cursor;           // カーソル絵
18. int gr_title_start;            // 始めるアイコン画像
19. int gr_title_end;              // 終わるアイコン画像
20. // 操作
21. int title_cursor_y;            // カーソル動作
22.
23. // 終了
24. bool endflag;
25.
26. int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
```

```
27. {
28.     ChangeWindowMode(TRUE);                // ウィンドウモード変更
29.     DxLib_Init();                          // 初期化
30.     SetDrawScreen( DX_SCREEN_BACK );        // 裏画面設定
31.
32.     endflag = false;
33.
34.     Initialize_Title();                     // タイトルの初期化の呼び出し
35.
36.     while( ScreenFlip()==0 && ProcessMessage()==0 && ClearDrawScreen()==0 && CheckHitKey(KEY_INPUT_ESCAPE)==0 && endflag == false)
37.     {
38.         Title_Update();                     // タイトル
39.     }
40.
41.     Delete_Title();                         // タイトルの削除
42.
43.     DxLib_End(); // DXライブラリ終了処理
44.     return 0;
45. }
46.
47. // タイトルの初期化
48. void Initialize_Title()
49. {
50.     // 描画
51.     gr_title_back = LoadGraph("media¥¥title_background.png");
52.     gr_title_cursor = LoadGraph("media¥¥title_cursor.png");
53.     gr_title_start = LoadGraph("media¥¥title_start.png");
54.     gr_title_end = LoadGraph("media¥¥title_end.png");
55.
56.     title_cursor_y = 0;
57. }
58.
59. // タイトルの削除
60. void Delete_Title()
61. {
62.     DeleteGraph(gr_title_back);
63.     DeleteGraph(gr_title_cursor);
64.     DeleteGraph(gr_title_start);
65.     DeleteGraph(gr_title_end);
66. }
67.
68. // タイトルのすべて
69. void Title_Update()
70. {
71.     Title_Draw();
72.     Title_Process();
```

```
73. }
74.
75. // タイトルの描画
76. void Title_Draw()
77. {
78.     DrawGraph(0, 0, gr_title_back, false);
79.     DrawGraph(240, 300 + (50 * title_cursor_y), gr_title_cursor, true);
80.     DrawGraph(290, 300, gr_title_start, true);
81.     DrawGraph(290, 350, gr_title_end, true);
82. }
83.
84. // タイトルの操作
85. void Title_Process()
86. {
87.     if(CheckHitKey(KEY_INPUT_UP) == 1)
88.     {
89.         if(title_cursor_y != 0)
90.         {
91.             title_cursor_y++;
92.         }
93.     }
94.     if(CheckHitKey(KEY_INPUT_DOWN) == 1)
95.     {
96.         if(title_cursor_y != 1)
97.         {
98.             title_cursor_y--;
99.         }
100.    }
101.
102.    if(CheckHitKey(KEY_INPUT_Z) == 1)
103.    {
104.        if(title_cursor_y == 0)
105.        {
106.
107.        }
108.        else if(title_cursor_y == 1)
109.        {
110.            endflag = true;
111.        }
112.    }
113. }
```

今回の関数内の変更ではそこまで難しくないので一応説明します。

【WinMain()】

ここでは、先ほど追加した【endflag】を終了処理として増やしました。【false】だったらwhile文をずっと繰り返すというものです。

【Title_Process()】

ここで、カーソルに関する処理を行っています。
十字キーの入力を取得して、その場所によってZキーを押されたら別々の処理を行うようにするものです。

一つ一つ見ていくと大変なので同じ関数は端折ります。

【CheckHitKey()】

これはユーザーの入力を取得する関数です。

【== 1】というのは一回入力されたら、という意味です。if文はいつものなので身構えなくて大丈夫です。

気づいた人はいるかもしれませんが、ある部分を変えればキーを変えれます。

【KEY_INPUT_○】のまるの部分です。
気になる人はいろいろいじってみてください。

これに関する説明は大変、大変なので端折ります。もっと気になる人はリファレンスを参照するといいたと思います。
[リファレンス](#)

次にProcess内の【title_cursor_y】に関するif文ですが・・・

上二つのif文ではカーソルの移動制限を行っています。下二つのif文ではカーソルの位置によってZキーを押すと別々の処理が行われる。

そういったものです。小難しいように思えますがものすごく単純なことなのでちょっと考えてみてください。

これでタイトルの大まかな処理は終了です。次はエンディング画面を作ります。エンディング画面と言ってもそこまでたいそうなものは作りません。
また、ゲーム画面は単純なゲームと言っても大変なので最後に説明します。

[C言語初めてのゲーム\(走り抜けジャンプ\)-part2](#)

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.