



情報研究会CACTUS

DxLib講座

DxLib とは何か？

...C言語で画像描画処理を行うための機能が
詰め込まれたライブラリ



動かしてみよう

```
1 #include <stdio.h>
2 #include "DxLib.h"
3
4 #define FPS      60    //fps
5 #define WIDTH    600  //画面の横 幅
6 #define HEIGHT   400  //画面の縦 幅
7
8 void Init() /*初期化関数*/
9 {
10     ChangeWindowMode(TRUE);
11     //ウィンドウモードを ON にする
12     SetGraphMode(WIDTH, HEIGHT, 32);
13     //ウィンドウの横幅を WIDTH、縦幅を HEIGHT、
14     //      描画画素数を 32 に設定する
15     if (DxLib_Init() == -1) exit(-1);
16     //DxLibの初期化処理がエラーを起したら直ちに終了する
17 }
18
19 WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,
20                LPSTR lpCmdLine, int nCmdShow) /*メイン関数*/
21 {
22     Init();
23
24     while(true){ /*ゲームループ*/
25
26         WaitTimer(1000 / FPS); // 1/FPS 秒間処理を停止する
27         if(ProcessMessage() == -1) break;
28         //ウィンドウが閉じられたらゲームループから離脱する
29
30     }
31
32     DxLib_End();
33     //DxLibを閉じる
34     return 0;
35 }[EOF]
```

①白い正方形を動かす

```
19 WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,  
20                LPSTR lpCmdLine, int nCmdShow) /*メイン関数*/  
21 {  
22     int box_x = 0, box_y = 0;  
23     char KeyBuf[256];  
24  
25     Init();  
26  
27     while(true){ /*ゲームループ*/  
28  
29         //描画内容を消去する  
30         ClearDrawScreen();  
31  
32         //全てのキーの押下状態を検知する  
33         GetHitKeyStateAll(KeyBuf);  
34  
35         //特定のキーが押されている時に特定の処理を行う  
36         if(KeyBuf[KEY_INPUT_LEFT]) box_x -= 3;  
37         if(KeyBuf[KEY_INPUT_RIGHT]) box_x += 3;  
38         if(KeyBuf[KEY_INPUT_UP]) box_y -= 3;  
39         if(KeyBuf[KEY_INPUT_DOWN]) box_y += 3;  
40  
41         //白く塗りつぶされた四角形を描画する  
42         DrawBox(box_x, box_y, box_x+60, box_y+60, 0xFFFFFFFF, TRUE);  
43  
44         WaitTimer(1000 / FPS); // 1/FPS 秒間処理を停止する  
45         if(ProcessMessage() == -1) break;  
46         //ウィンドウが閉じられたらゲームループから離脱する  
47  
48     }
```

座標

x

0

y

(box_x , box_y)

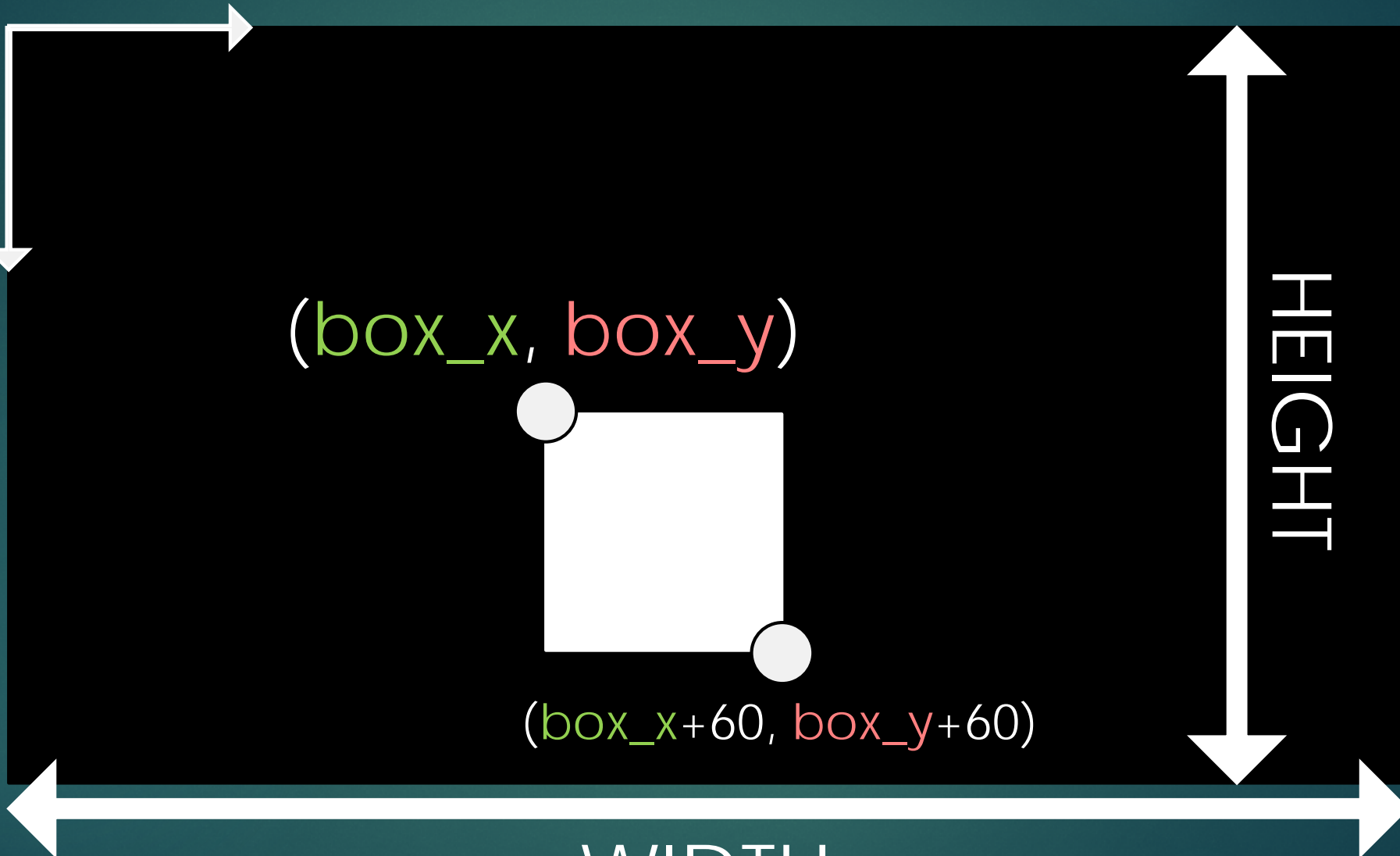


($\text{box_x}+60$, $\text{box_y}+60$)



HEIGHT

WIDTH



②緑の円を配置する

```
1 #include <stdio.h>
2 #include "DxLib.h"
3 #include <stdlib.h>
4 #include <time.h>
5
6 #define FPS      60 //fps
7 #define WIDTH    600 //画面の横幅
8 #define HEIGHT  400 //画面の縦幅
9
10 void Init(); /*初期化関数*/
11
12 /*二次座標の構造体POSI*/
13 typedef struct{
14     int x, y;
15 }POSI;
16
17 WINAPI WinMain(HINSTANCE hInstance, HINSTA
18                 LPSTR lpCmdLine, int nCmdS
19 {
20     int i;
21     int box_x = 0, box_y = 0;
22     POSI circle[3];
23     char KeyBuf[256];
24
25     Init();
26
27     //乱数のシード値を現在時刻から得る
28     srand((unsigned int)time(NULL));
29
30     //円の座標を乱数で決定する
31     for(i=0;i<3;i++){
32         circle[i].x = rand() % WIDTH;
33         circle[i].y = rand() % HEIGHT;
34     }
```

```
36 while(true){ /*ゲームループ*/
37
38     //描画内容を消去する
39     ClearDrawScreen();
40
41     //全てのキーの押下状態を検知する
42     GetHitKeyStateAll(KeyBuf);
43
44     //特定のキーが押されている時に特定の処理を行う
45     if(KeyBuf[KEY_INPUT_LEFT]) box_x -= 3;
46     if(KeyBuf[KEY_INPUT_RIGHT]) box_x += 3;
47     if(KeyBuf[KEY_INPUT_UP]) box_y -= 3;
48     if(KeyBuf[KEY_INPUT_DOWN]) box_y += 3;
49
50     //白く塗りつぶされた四角形を描画する
51     DrawBox(box_x, box_y, box_x+60, box_y+60, 0xFFFFFFFF, TRUE);
52
53     //緑色に塗りつぶされた円を描画する
54     for(i=0;i<3;i++){
55         DrawCircle(circle[i].x, circle[i].y, 15, 0x00FF00, TRUE);
56     }
57
58     WaitTimer(1000 / FPS); // 1/FPS 秒間処理を停止する
59     if(ProcessMessage() == -1) break;
60     //ウィンドウが閉じられたらゲームループから離脱する
61 }
```

色の表現

円の描画

0xFF00FF

6桁の16進数で表現

0xFFFF00 ... 

0xFF00FF ... 

0x000000 ... 

0x808080 ... 

(circle[i].x, circle[i].y)



③判定を作る

```
12 /*二次座標とフラグの構造体POSI*/
13 typedef struct{
14     int x, y, flag;
15 }POSI;
```

```
17 WINAPI WinMain(HINSTANCE hInstance,
18                 LPSTR lpCmdLine, int
19                 {
20     int i;
21     int box_x = 0, box_y = 0;
22     POSI circle[3];
23     char KeyBuf[256];
```

```
24
25     Init();
```

```
26
27     //乱数のシード値を現在時刻から得る
28     srand((unsigned int)time(NULL));
```

```
29
30     //円の座標を乱数で決定し、フラグを1にする
31     for(i=0;i<3;i++){
32         circle[i].x = rand() % WIDTH;
33         circle[i].y = rand() % HEIGHT;
34         circle[i].flag = 1;
35     }
```

```
51 //白く塗りつぶされた四角形を描画する
52 DrawBox(box_x, box_y, box_x+60, box_y+60, 0xFFFFFFFF, TRUE);
```

```
53
54 //フラグが0でなければ緑色に塗りつぶされた円を描画する
55 for(i=0;i<3;i++) if(circle[i].flag != 0)
56     DrawCircle(circle[i].x, circle[i].y, 15, 0x00FF00, TRUE);
```

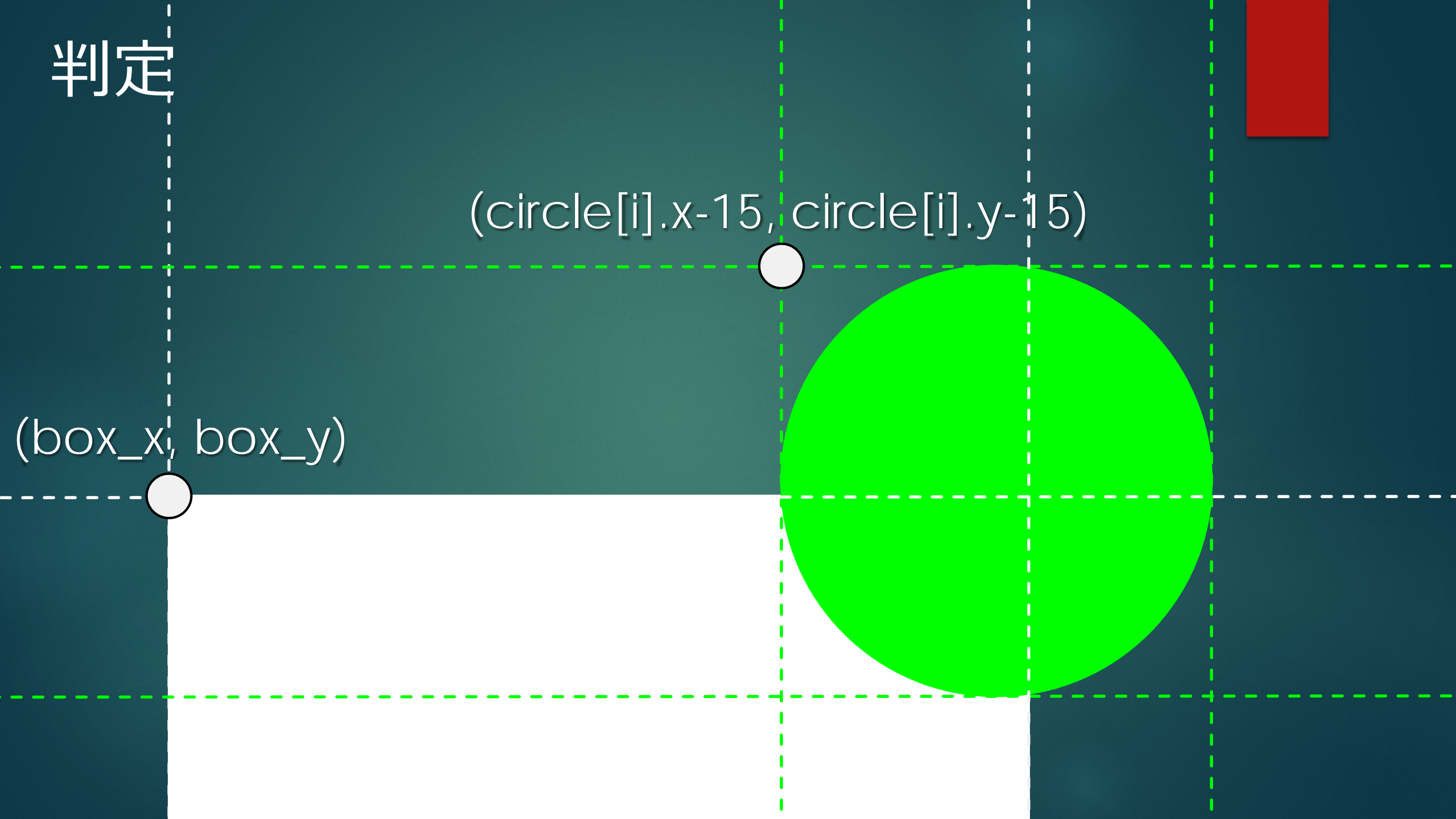
```
57
58 //緑の円が白い四角形に接触したらフラグを0にする
59 for(i=0;i<3;i++){
60     if(((circle[i].x-15 > box_x && circle[i].x-15 < box_x+60)
61         || (circle[i].x+15 > box_x && circle[i].x+15 < box_x+60))
62         && ((circle[i].y-15 > box_y && circle[i].y-15 < box_y+60)
63             || (circle[i].y+15 > box_y && circle[i].y+15 < box_y+60)))
64         circle[i].flag = 0;
65 }
```

```
66
67 WaitTimer(1000 / FPS); // 1/FPS 秒間処理を停止する
68 if(ProcessMessage() == -1) break;
69 //ウィンドウが閉じられたらゲームループから離脱する
70 }
```


判定

$(\text{circle}[i].x-15, \text{circle}[i].y-15)$

$(\text{box}_x, \text{box}_y)$



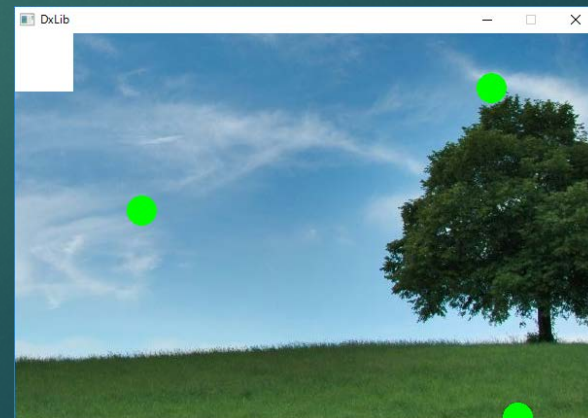
+ α

Q. 画像を表示するには？

A. LoadGraph関数、DrawGraph関数を使います。

```
19 1
20   int i, g_Handle;
21   int box_x = 0, box_y = 0;
22   POSI circle[3];
23   char KeyBuf[256];
24
25   Init();
26
27   //画像ファイルの読み込み
28   g_Handle = LoadGraph("風景.jpg");
29
```

```
52   if(KeyBuf[KEY_INPUT_DOWN]) box_y +=
53
54   //左上端を座標0,0として画像を描画
55   DrawGraph(0, 0, g_Handle, FALSE);
56
57   //白く塗りつぶされた四角形を描画する
58   DrawBox(box_x, box_y, box_x+60, box_y+60, 0);
59
```



+ α

Q. BGMや効果音が欲しい

A. LoadSoundMem関数、PlaySoundMem関数を使いましょう。

Q. マウスで操作ってできないの？

A. GetMousePoint関数、GetMouseInput関数を用いれば可能です。

Q. 動画は使える

A. ちゃんと専用の関数があります。

Q. 時代は3Dでしょ

A. 3Dモデルの読み込みや3次元空間での描画制御等もできます。

Q. 好きなフォントを使いたい

A. ChangeFont関数等で実現できます。

Q. その他にはどんなことができるんだろうか

A. タッチパネルやジョイパッドなんかも使えるらしいですよ？



Let's Enjoy the DxLib!!