

ゲームプログラミング講習

第2章 関数の使い方

DXライブラリの関数を使います

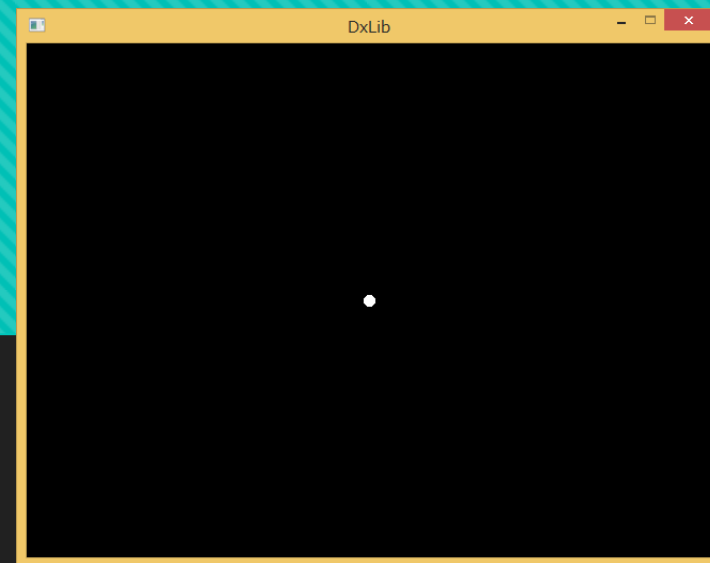
今回やること

- はじめに
 - 座標
 - 色
- 関数リファレンス
- 図形
- 画像

はじめに

はじめに

- 次の通り入力して実行しよう

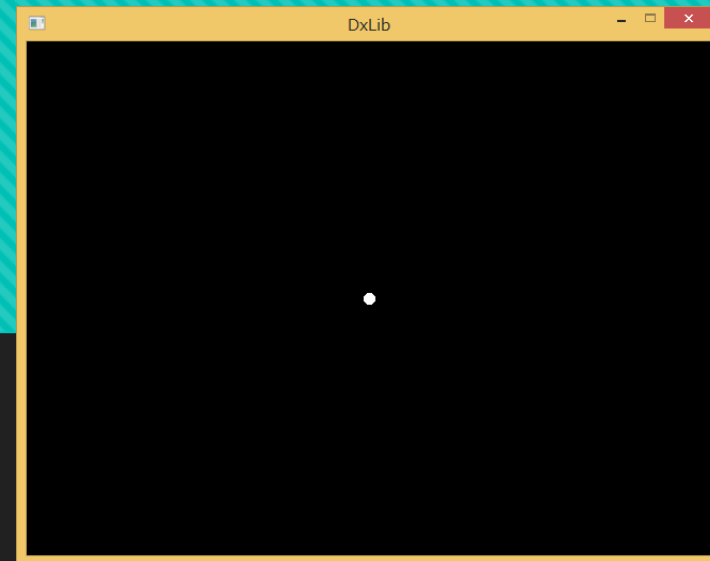


```
/*          ↓ ↓ ↓ ↓ ↓          ゲーム進行・描画          ↓ ↓ ↓ ↓ ↓          *
```

```
DrawCircle(320, 240, 5, GetColor(255, 255, 255), TRUE);
```

```
/*          ↑ ↑ ↑ ↑ ↑          ゲーム進行・描画          ↑ ↑ ↑ ↑ ↑          *
```

はじめに



```
/*      ↓ ↓ ↓ ↓ ↓      ゲーム進行・描画      ↓ ↓ ↓ ↓ ↓      */
円を描画      円の詳細
DrawCircle(320, 240, 5, GetColor(255, 255, 255), TRUE);
          x   y   半径      色      塗りつぶす
/*      ↑ ↑ ↑ ↑ ↑      ゲーム進行・描画      ↑ ↑ ↑ ↑ ↑      */
```

座標

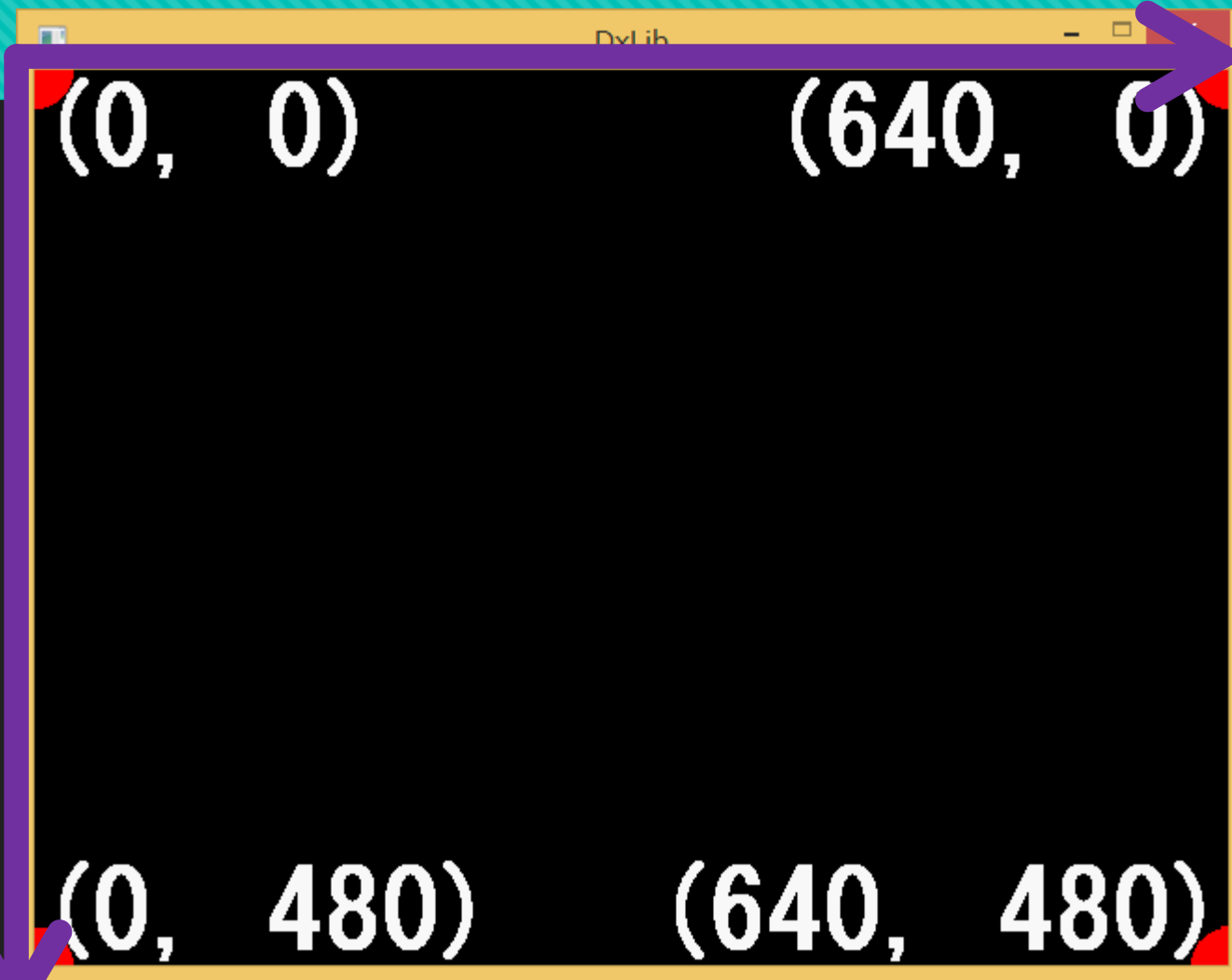
(320, 240,

X Y

- 原点は左上
- y軸は下向き
- 右下の座標は (640, 480)
- 整数を使う

0

y



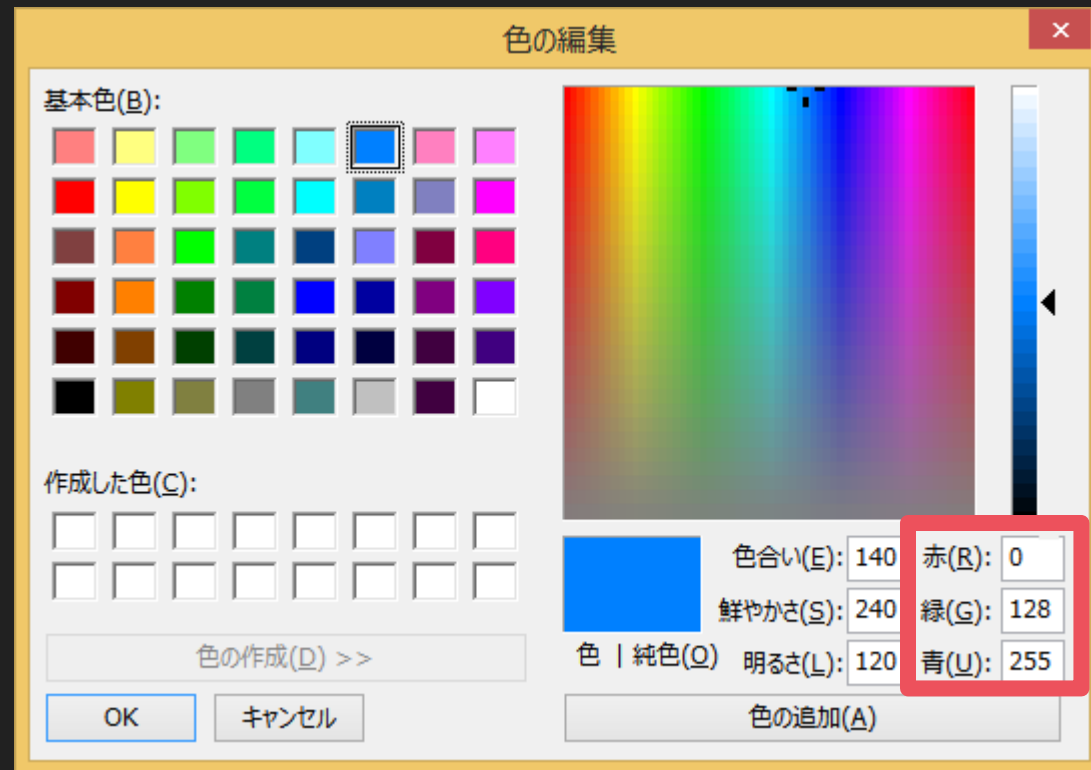
x

色

```
GetColor(255, 255, 255)
```

赤 緑 青

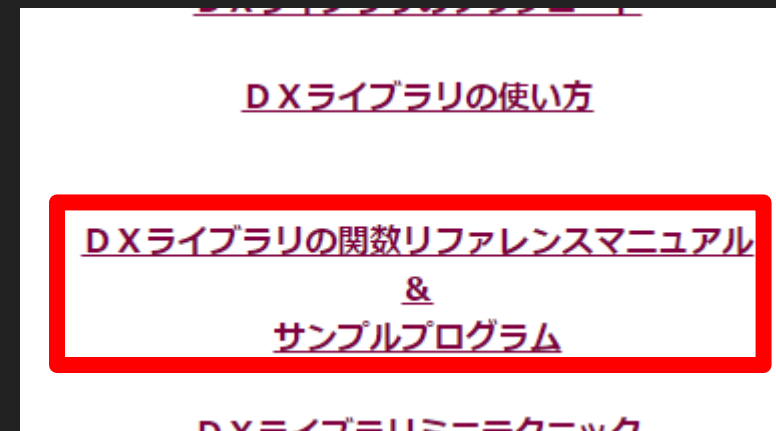
- GetColor(, ,) は覚えよう
- 光の三原色それぞれの強さを 0~255 で指定
- ペイントで「色の編集」をクリックすると、三原色それぞれの強さがわかる
- 右の例の色は以下のように入力する
 - GetColor(0, 128, 255)



関数リファレンス

関数リファレンス

1. 「DXライブラリ置き場 HOME」に移動
2. お気に入り または ブックマーク に追加
3. 下へスクロールし、「DXライブラリの関数リファレンスマニュアル」をクリック



関数リファレンス

- このページはとても**大事**
- 辞書のように何度も使う
- DXライブラリで用意された「関数」の一覧
- 分類分けされていて、**緑色の文字**を見るとすぐに目的の関数が見つかる

DXライブラリ 関数リファレンスページ

このページではDXライブラリの主な関数の説明と簡単な サンプルプログラムを載せています。

使用必須関数

[DxLib_Init](#)

ライブラリの初期化

[DxLib_End](#)

ライブラリ使用の終了関数

[ProcessMessage](#)

ウインドウズのメッセージを処理する

3D関係関数

[一覧はこちら](#)

数が多いので分けました

図形描画関数

[DrawLine](#)

線を描画

[DrawBox](#)

四角を描画

[DrawCircle](#)

円を描画

[DrawOval](#)

楕円を描画

[DrawTriangle](#)

三角形を描画

[DrawPixel](#)

点を描画する

[GetPixel](#)

指定点の色を取得

関数リファレンス

- 図形描画関数の「DrawCircle」に注目
 - 右の説明の通り、円の描画をする関数である
 - 英語からだいたい予想できる
- 「DrawCircle」をクリック

図形描画関数

DrawLine	線を描画
DrawBox	四角を描画
DrawCircle	円の描画
DrawOval	楕円の描画
DrawTriangle	三角形の描画
DrawPixel	点を描画する
GetPixel	指定点の色を取得

関数リファレンス

- 「int」「unsigned int」はとりあえず無視
- これを見ながらプログラミングしよう
- 例えば下のようになる

宣言

```
int DrawCircle(int x, int y, int r, unsigned int Color, int FillFlag );
```

概略

円の描画

引数

x, y : 描く円の中心座標
r : 描く円の半径
Color : 円の色
FillFlag : TRUE(1の意)で円の中身も塗りつぶし、FALSE(0の意)で輪郭のみ

戻り値

0 : 成功
-1 : エラー発生

解説

(x , y)を円の中心座標とした半径 r の円を描きます。
円はColorで指定した色で描かれ、FillFlag にTRUEを指定した場合は中身を塗りつぶし、FALSEを指定した場合は輪郭のみ描かれます。
(パソコン画面上での座標の取り方、色の指定のし方は [DrawLine](#) の解説を参照してください。)

```
DrawCircle(320, 240, 5, GetColor(255, 255, 255), TRUE);
```

図形

図形

- まず、関数リファレンスを開く
 - リファレンスと聞いたらこのページを開こう

DXライブラリ 関数リファレンスページ

このページではDXライブラリの主な関数の説明と簡単な サンプルプログラムを載せています。

使用必須関数

[DxLib_Init](#)

ライブラリの初期化

[DxLib_End](#)

ライブラリ使用の終了関数

[ProcessMessage](#)

ウインドウズのメッセージを処理する

3D関係関数

[一覧はこちら](#)

数が多いので分けました

図形描画関数

[DrawLine](#)

線を描画

[DrawBox](#)

四角を描画

[DrawCircle](#)

円を描画

[DrawOval](#)

楕円を描画

[DrawTriangle](#)

三角形を描画

[DrawPixel](#)

点を描画する

[GetPixel](#)

指定点の色を取得

図形

- このページを参考に、「四角を描画」してみよう
 - 座標、色はご自由にどうぞ
- やればきっとできる！

DXライブラリ 関数リファレンスページ

このページではDXライブラリの主な関数の説明と簡単な サンプルプログラムを載せています。

使用必須関数

[DxLib_Init](#)

ライブラリの初期化

[DxLib_End](#)

ライブラリ使用の終了関数

[ProcessMessage](#)

ウインドウズのメッセージを処理する

3D関係関数

[一覧はこちら](#)

数が多いので分けました

図形描画関数

[DrawLine](#)

線を描画

[DrawBox](#)

四角を描画

[DrawCircle](#)

円を描画

[DrawOval](#)

楕円を描画

[DrawTriangle](#)

三角形の描画

[DrawPixel](#)

点を描画する

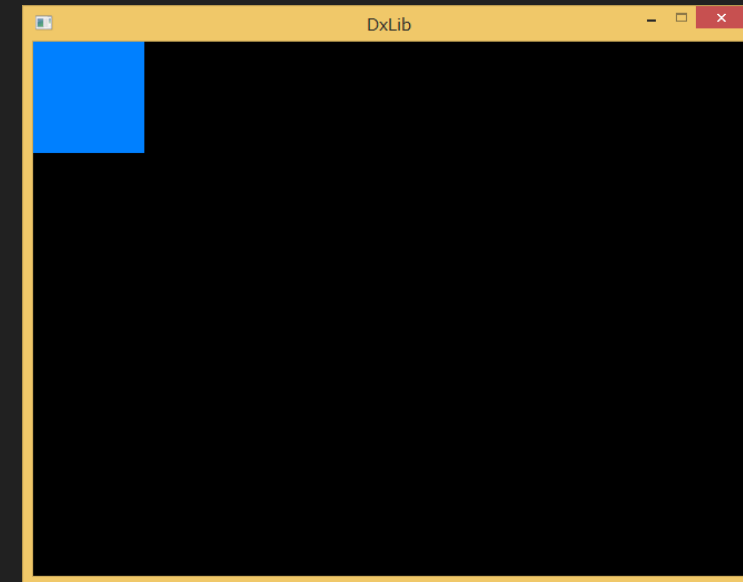
[GetPixel](#)

指定点の色を取得

図形

○ 「四角を描画」の例

```
/*      ↓ ↓ ↓ ↓ ↓      ゲーム進行・描画      ↓ ↓ ↓ ↓ ↓      */  
DrawBox(0, 0, 100, 100, GetColor(0, 128, 255), TRUE);  
/*      ↑ ↑ ↑ ↑ ↑      ゲーム進行・描画      ↑ ↑ ↑ ↑ ↑      */
```



画像

画像

終わり

お疲れ様でした