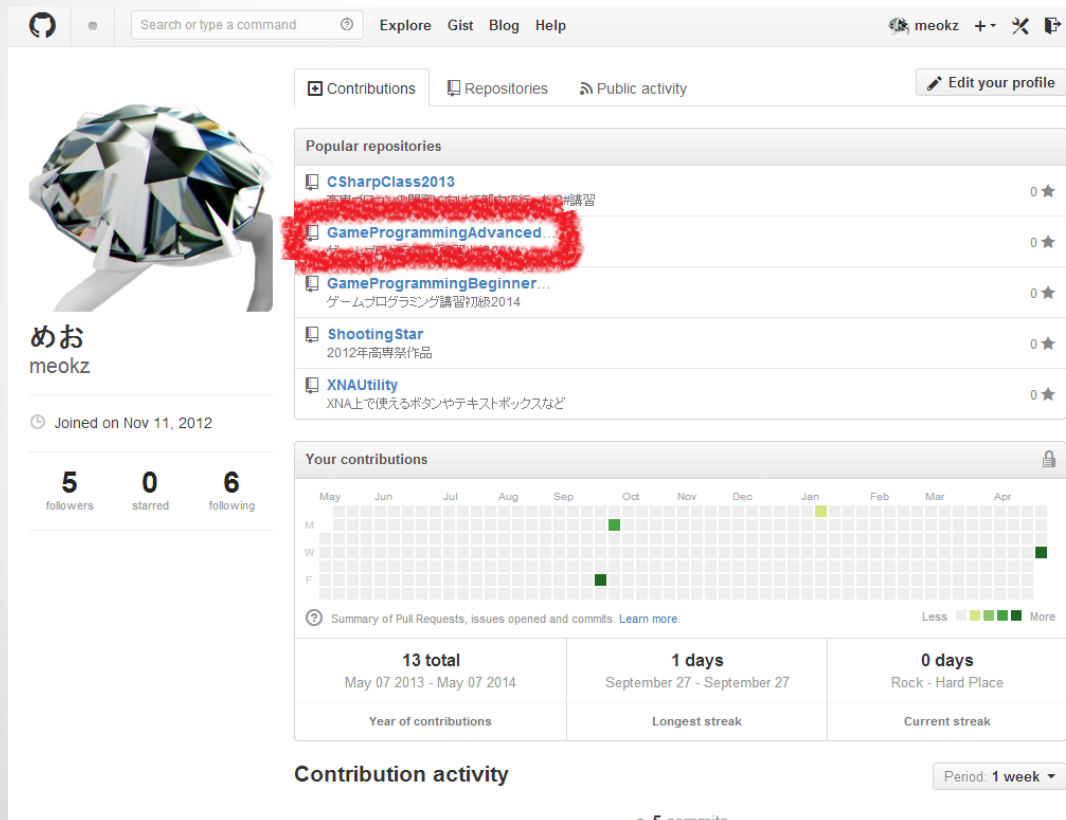


C++ゲームプログラミング講習 上級 第1回

イントロダクション

講習資料について

- 講習資料はGitHub上に置きます。プロジェクトファイルや素材なども同梱するので、リポジトリをクローンしておくことをおすすめします。



The screenshot shows the GitHub profile of user 'meokz'. The profile includes a bio, a profile picture of a diamond, and statistics: 5 followers, 0 starred repositories, and 6 following. The 'Popular repositories' section lists several repositories, with 'GameProgrammingAdvanced...' highlighted in red. The 'Your contributions' section shows a calendar grid with green squares indicating contributions. Below the calendar, there are statistics for total contributions (13 total), longest streak (1 days), and current streak (0 days). The 'Contribution activity' section shows 5 commits over a 1-week period.

めお
meokz

Joined on Nov 11, 2012

5 followers 0 starred 6 following

Popular repositories

- CSharpClass2013
- GameProgrammingAdvanced...
- GameProgrammingBeginner...
- ShootingStar
- XNAUtility

Your contributions

Summary of Pull Requests, issues opened and commits. [Learn more.](#)

13 total
May 07 2013 - May 07 2014
Year of contributions

1 days
September 27 - September 27
Longest streak

0 days
Rock - Hard Place
Current streak

Contribution activity


Period: 1 week

5 commits

<https://github.com/meokz>
にアクセス

「GameProgramming
AdvancedClass2014」
をクリック

講習資料について

 **meokz / GameProgrammingAdvancedClass2014**

Watch 0 Star 0 Fork 0

ゲームプログラミング講習上級2014 — Edit

2 commits 1 branch 0 releases 1 contributor

branch: master GameProgrammingAdvancedClass2014 / +

First Commit

meokz authored 2 hours ago latest commit a1b77247fc

DxLib	First Commit	2 hours ago
第01回	First Commit	2 hours ago
.gitignore	Initial commit	4 hours ago
README.md	Initial commit	4 hours ago

README.md

GameProgrammingAdvancedClass2014

Code

Issues 0

Pull Requests 0

Wiki

Pulse

Graphs

Network

Settings

SSH clone URL

git@github.com:meo

You can clone with HTTPS, SSH,

Clone in Desktop

Download ZIP

講習資料について

- cygwinを起動する
- git clone [git://github.com/GameProgrammingAdvancedClass2014.git](https://github.com/GameProgrammingAdvancedClass2014.git)
- git clone [git@github.com:meokz/GameProgrammingAdvancedClass2014.git](https://github.com/meokz/GameProgrammingAdvancedClass2014.git)
- 情処じゃできないかも
- 2回目以降は git pull でOK

講習の趣旨

- プログラミング言語C++とDxLibを用いたWindows用ゲームアプリケーションの作り方を学ぶ
- リファレンスを読みながらプログラミングをすることを学ぶ
- オブジェクト指向プログラミングを学ぶ

C++

- C言語の拡張として開発された言語
- (語弊はあるが簡単に言うと)
C言語にクラスを追加したもの
- C++のコンパイラでC言語がコンパイルできるため、
C言語の記述で書くことも可能

一言でいうと

聞

Hello World(C言語風)

Standard Input Output

- 標準出力
- "スタジオ"ではない

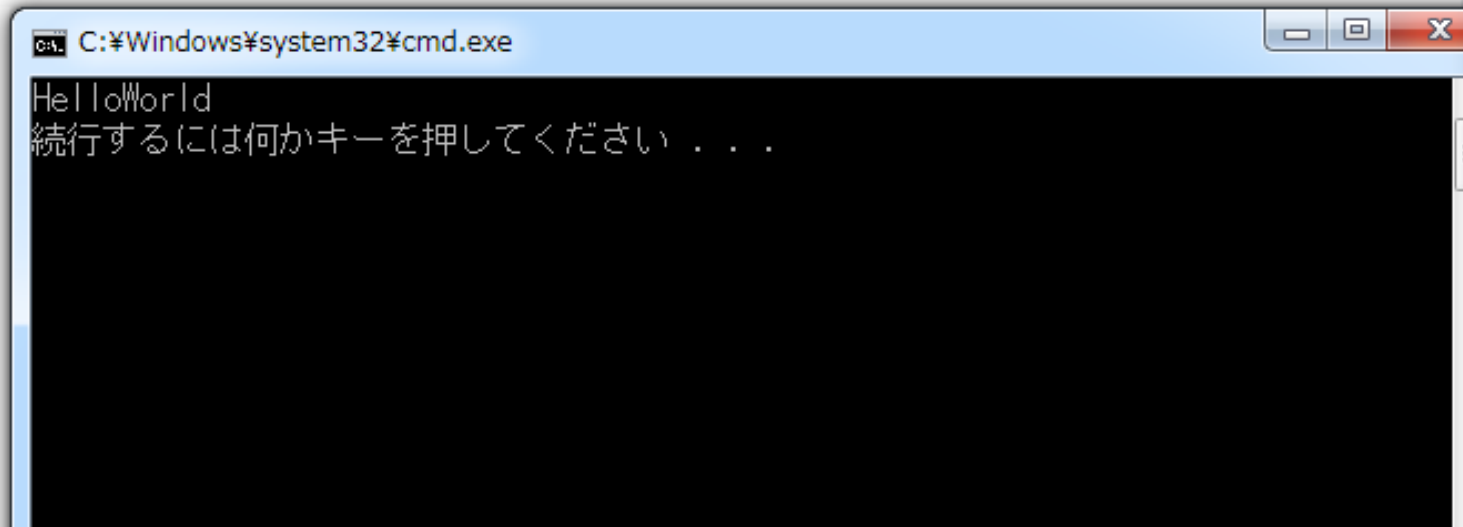
```
1 #include "stdio.h"
2
3 int main() {
4     printf("HelloWorld\n");
5     return 0;
6 }
```

C:\Windows\system32\cmd.exe

HelloWorld
続行するには何かキーを押してください . . .

Hello World(C++風)

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "HelloWorld" << endl;
6     return 0;
7 }
```

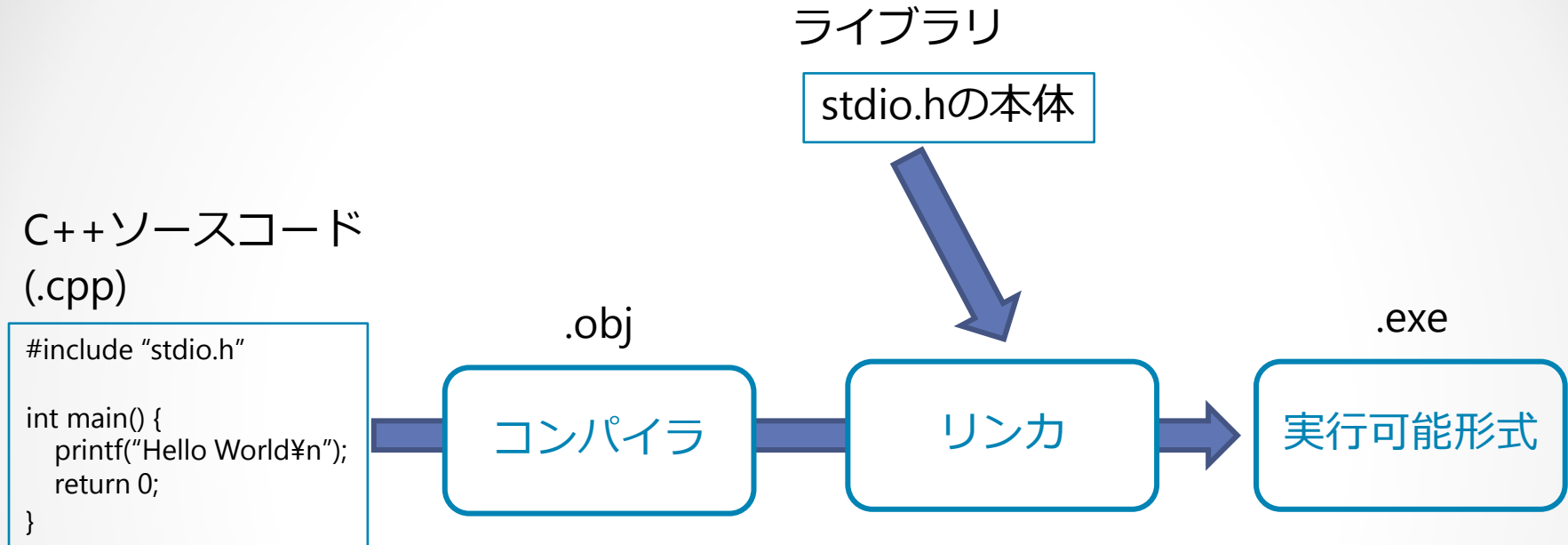


A screenshot of a Windows command prompt window. The title bar shows the path "C:\Windows\system32\cmd.exe". The command prompt displays the output "HelloWorld" on the first line. On the second line, it shows the Japanese text "続行するには何かキーを押してください . . ." (Press any key to continue . . .). The background of the command prompt is black, and the text is white.

C++とJavaの違い

	C++	Java	C#
コンパイル&実行方式	機械語にコンパイル	中間言語にコンパイルし、JVMで実行	中間言語にコンパイルし、.NETで実行
コンパイル速度	クソ遅い	普通	早い
実行速度	最速	遅い	早い
ガーベジコレクション	なし	あり	あり

コンパイル&実行形式



- コンパイルしたものはプラットフォームに依存する
- 直接 機械語にするため 実行時最速

コンパイル&実行形式

Javaソースコード

```
class Hoge {  
    public static void main(  
        String[] args) {  
        System.out.println(  
            "HelloWorld");  
        }  
    }  
}
```



- 中間言語をかまして、[JavaVirtualMachine](#)で仮想上で動かすのでプラットフォームを問わない
- 仮想上で動かすので遅くなる

ガーベジコレクション(GC)

プログラムがメモリを動的に確保するとき、
確保したメモリは解放しなければならない
(メモリリークの発生)

自動でメモリを解放してくれる！

メモリの確保と解放を続けると
メモリフラグメンテーションが発生

メモリを再配置し、断片化を解消してくれる！

(厳密には)メモリコンパクション

まとめ

- C++は実行時最速
- ガベージコレクションがないので、メモリに気を使う必要がある
- マルチプラットフォームに対応し、実行速度もはやく、ガベージコレクションも備えているのがC#

DXライブラリとは

- DirectXやWindowsAPIを簡単に使えるようにしたもの
- 2Dゲームならこれだけでどんなゲームでも作れる
といっても過言ではない
(ただし、最近の2Dゲームは描画が3Dであることも多いので注意。)
- カタカナ表記ならDXライブラリ
- 英語表記ならDxLib

開発環境

- Visual Studio 2012

(無料版のVisualStudio for Windows Desktop でも可。

高専生ならDreamSparkを利用すれば、無償でVS2012 Professional
が使える)

- DXライブラリ

プロジェクトを作る

Since 2001.5.01 Last Up 2014.4.19 ライブラリバージョン 3.11f

DXライブラリ置き場



DXライブラリとは、DirectXを使ったWindowsソフトの開発に必ず付いて回るDirectXやWindows関連のプログラムを使い易くまとめた形で利用できるようにしたC++言語用のゲームライブラリです。(使用する際はC言語の知識だけで大丈夫です)
これによってプログラマーはゲームの本質的なプログラムに専念することが出来ます。かなり本格的なソフト制作からお遊び程度のミニゲーム制作まで幅広くカバーしています！

プログラムソースも公開していますので気兼ねなく使用してください。

[<しばらくCM>](#)

14歳からはじめる C言語わくわくゲームプログラミング教室
Visual Studio 2010編(左) と

14歳からはじめる C++わくわくゲームプログラミング教室(右)
著者はどちらも 大槻 有一郎様



ゲームプログラムの入門書です
C言語版は、初めてC言語に触れる方や、これからゲームプログラムを始める方、DXライブラリの使い方が分からない方におすすめです。
(解説されているゲームはアクションゲーム)
C++版は、C言語を知っている方向けの続編で、C++言語に触れてみたい方におすすめです。(解説されているゲームはSTGとリアルタイムSLG)

[DXライブラリ更新履歴](#)

[DXライブラリHP更新履歴](#)

- <http://homepage2.nifty.com/natupaji/DxLib/> にアクセス(または"DXライブラリ"で検索)

- 「DXライブラリの使い方」から自分の持っているVisualStudioのバージョンの構築方法を調べてください

- 結構めんどくさいので、この講習ではあらかじめ構築したプロジェクトを使用する

Let's Programming

- DxLib/DxLib.slnを開きましょう

ウィンドウを作る

```
1  #include "DxLib.h"
2
3  int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance,
4                      LPSTR lpCmdLine, int nCmdShow ) {
5
6      ChangeWindowMode(TRUE);
7
8      DxLib_Init();
9
10     WaitKey();
11
12     DxLib_End();
13
14     return 0;
15
16 }
```

- ☑プロジェクトを開き、6~12行目のコードを追加して、F5キーを押して実行する

メイン関数

```
int WINAPI WinMain(HINSTANCE hInstance,  
HINSTANCE hPrevInstance, LPSTR lpCmdLine, int  
nCmdShow) {}
```

- public static void main(String[] args) {} みたいなもん
- 覚えなくていい。コピペでOK

読み流してよい

- `int WINAPI WinMain(~~~~)` → “メイン関数”
- `HINSTANCE` → プログラムを識別するためのもの。
Windows側から与えられるので考慮しなくていい
- 2つ目の`HINSTANCE`は → 常にNULLだから無視でいい
- `LPSTR`は → コマンドラインから受け取るオプション
- `int nShowCmd`は → ウィンドウの表示方法(最大, 最小, 通常など)

つまり

おまじない

ChangeWindowMode関数

宣言	<code>int ChangeWindowMode(int Flag);</code>
概略	ウィンドウモード・フルスクリーンモードの変更
引数	Flag – TRUE : ウィンドウモードで起動 / FALSE : フルスクリーンモードで起動
戻り値	リファレンス参照
解説	リファレンス参照

- ☑ChangeWindowMode(FALSE) にして実行してみる
- ☑ChangeWindowModeを省略して実行してみる

DxLib_Init関数

宣言	int DxLib_init();
概略	DXライブラリの初期化
戻り値	0 : 成功 / -1 : エラー発生
解説	DXライブラリを使用する際、はじめに呼び出す関数

☑エラーが発生した際、プログラムを終了するように
してみる

WaitKey関数

宣言	<code>int WaitKey();</code>
概略	キーの入力待ち
戻り値	入力されたキーのコード(リファレンス参照)
解説	何かのキーが入力されるまで待つ

☑WaitKey()をコメントアウトしてプログラムを実行
してみる

DxLib_End関数

宣言	int DxLib_End();
概略	DXライブラリの終了
戻り値	0 : 成功 / -1 : エラー発生
解説	DXライブラリの使用を終了する関数

HelloWorldを描画してみる！



ゲームプログラムの入門書です

C言語版は、初めてC言語に触れる方や、これからゲームプログラムを始める方、DXライブラリの使い方が分からない方におすすめです。
(解説されているゲームはアクションゲーム)
C++版は、C言語を知っている方向けの続編で、C++言語に触れてみたい方におすすめです。(解説されているゲームはSTGとリアルタイムSLG)

[DXライブラリ更新履歴](#)

[DXライブラリHP更新履歴](#)

[DXライブラリの仕様と特徴](#)

[どこまでできるDXライブラリ](#)

[DXライブラリのダウンロード](#)

[DXライブラリの使い方](#)

[DXライブラリの関数リファレンスマニュアル
&
サンプルプログラム](#)

[DXライブラリミニテクニック](#)

[DXライブラリサンプルプログラム](#)

[DXライブラリゲームプログラム開発講座](#)

- [webページにアクセス](#)

- 「DXライブラリの関数リファレンス(ry)を開く

関数リファレンスページ

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](#) 数が多いので分けた

図形描画関数

[Draw_line](#) 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

グラフィックデータ制御関数

[LoadGraphScreen](#) 画像ファイルを読みこんで画面に表示する
[LoadGraph](#) 画像ファイルのメモリへの読みこみ、及び動画像ファイルのロード
[LoadDivGraph](#) 画像ファイルのメモリへの分割読みこみ
[MakeGraph](#) 空のグラフィックを作成する
[MakeScreen](#) 描画対象にできるグラフィックを作成する
[SetDrawValidMultiSample](#) 描画対象にできるグラフィックのマルチサンプリング設定を行う
[SetCreateGraphColorBitDepth](#) 作成するグラフィックのビット深度を設定する
[SetDrawValidFloatTypeGraphCreateFlag](#) 描画可能な浮動小数点型のグラフィックを作成するかどうかの設定を行う
[SetCreateDrawValidGraphChannelNum](#) 作成する描画可能なグラフィックのチャンネル数の設定を行う
[SetUsePremulAlphaConvertLoad](#) 読み込み時に画像を乗算済みα画像に変換するかを設定する
[DrawGraph](#) メモリに読みこんだグラフィックの描画
[DrawTurnGraph](#) メモリに読みこんだグラフィックのLR反転描画
[DrawExtendGraph](#) メモリに読みこんだグラフィックの拡大縮小描画
[DrawRotaGraph](#) メモリに読みこんだグラフィックの回転描画
[DrawRotaGraph2](#) メモリに読みこんだグラフィックの回転描画(回転中心指定あり)
[DrawRotaGraph3](#) メモリに読みこんだグラフィックの回転描画(回転中心指定あり、縦横拡大率別指定)
[DrawModiGraph](#) メモリに読みこんだグラフィックの自由変形描画
[DrawRectGraph](#) グラフィックの指定矩形部分のみを描画
[DerivationGraph](#) 指定のグラフィックの指定部分だけを抜き出して新たなグラフィックを作成する
[GetDrawScreenGraph](#) 描画先に設定されているグラフィック領域から指定領域のグラフィックを読みこむ
[GetGraphSize](#) グラフィックのサイズを得る
[InitGraph](#) 読みこんだグラフィックデータをすべて削除する
[DeleteGraph](#) 指定のグラフィックをメモリ上から削除する
[SetDrawMode](#) 描画モードをセットする
[SetDrawBlendMode](#) 描画の際のブレンドモードをセットする
[SetDrawBright](#) 描画輝度をセット
[SetTransparentColor](#) グラフィックに設定する透過色をセットする

- DXライブラリで使える関数(機能)が解説, サンプル付きで載っている
- 何かあったらここを覗いてみよう。新しい発見もあるかも
- (ここに載っていない裏関数も存在する)

• [Ctrl + F から「文字列」で検索をかける](#)

ありました

文字描画関係関数

DrawString
DrawFormatString
GetDrawStringWidth
GetDrawFormatStringWidth
SetFontSize
SetFontThickness
ChangeFont
ChangeFontType

CreateFontToHandle
DeleteFontToHandle
SetFontCacheUsePremulAlphaFlag
DrawStringToHandle
DrawFormatStringToHandle
GetDrawStringWidthToHandle
GetDrawFormatStringWidthToHandle
GetFontStateToHandle
InitFontToHandle

文字列を描画する
書式付き文字列を描画する
DrawString で描画される文字列の幅(ドット単位)を得る
DrawFormatString 関数書式付き文字列の描画幅(ドット単位)を得る
描画する文字列のサイズをセットする
描画する文字列の文字の太さをセットする
描画するフォントを変更する
文字列描画に使用するフォントのタイプを変更する

新しいフォントデータを作成
フォントデータを削除する
作成するフォントデータを『乗算済み α 』用にするかどうかを設定する
指定のフォントデータで文字列を描画する
指定のフォントデータで書式付き文字列を描画する
指定のフォントデータで描画する文字列の幅を得る
指定のフォントデータで書式付き文字列の描画幅を得る
指定のフォントデータの情報を得る
フォントデータを全て初期化する

DrawString

宣言	<code>int DrawString(int x, int y, char *String, int Color);</code>
概略	文字列を描画する
引数	x, y – 文字列を描画する座標 String – 描画する文字列のポインタ Color – 描画する文字列の色
戻り値	リファレンス参照
解説	char *String はstring型という認識でOK

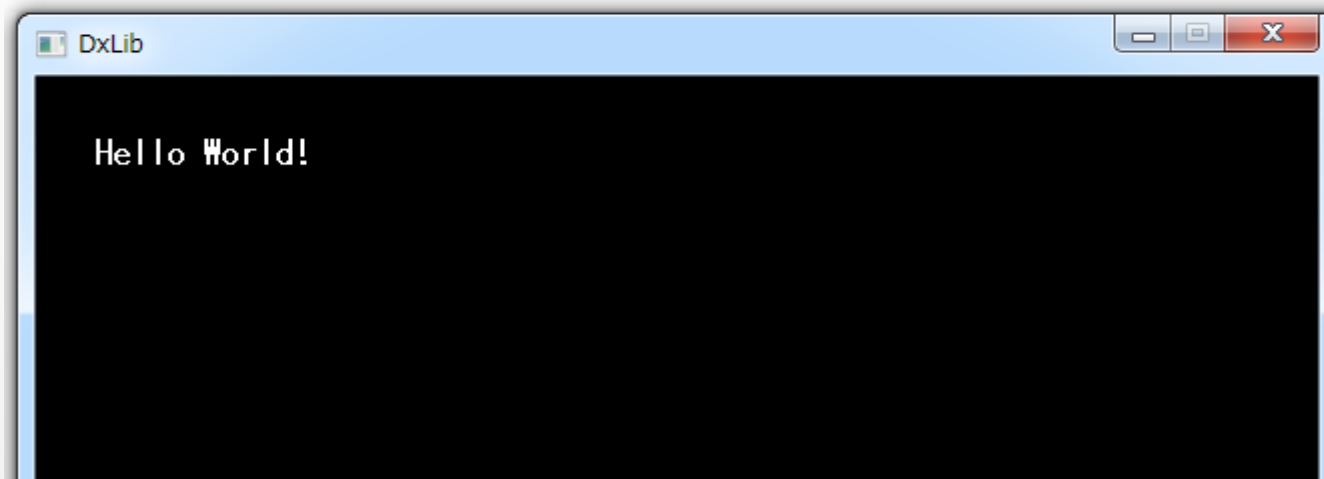
- ☑ サンプルを見たところ、GetColor関数なるもので色を取得しなければならないらしい。
- ☑ GetColor関数を調べる

GetColor

宣言	<code>int GetColor(int Red, int Green, int Blue)</code>
概略	色コードを取得する
引数	Red, Green, Blue – 取得したい色の各輝度値(0~255)
戻り値	カラーコード
解説	単体で使うことはまずなく、何か描画したいとき、その色を指定するのに使う

ソース&実行結果

```
1  #include "DxLib.h"
2
3  int WINAPI WinMain( HINSTANCE hInstance, HINSTANCE hPrevInstance,
4                      LPSTR lpCmdLine, int nCmdShow ){
5
6      ChangeWindowMode(TRUE);
7
8      if(DxLib_Init() == -1) return -1;
9
10     int white = GetColor(255, 255, 255);
11
12     DrawString(30, 30, "Hello World!", white);
13
14     WaitKey();
15
16     DxLib_End();
17
18     return 0;
19 }
```



まとめ

- Javaが書ければC++の基本文法もOK
(C++とJavaの違いも説明していきます)
- DxLibはリファレンスとにらめっこしてみよう

参考になりそうなwebページ

- DXライブラリ置き場 <http://homepage2.nifty.com/natupaji/DxLib/>
本家。環境構築、リファレンス、サンプルなど多数
- 新・ゲームプログラミングの館 <http://dixq.net/g/>
DxLibの解説。ゲームプログラミング

課題

- 文字の色を赤に変えて、
「Tokuyama College of Technology」を出力する
- リファレンスを調べながら、
ウィンドウのタイトルを変更する

(家に帰ってから)

- 自分のパソコンにVisualStudioをインストール
- DXライブラリの環境を構築する

追記

- Github公開用のリポジトリを作り直したため、「講習資料について」のスライドに記載されているリポジトリにはアクセスできません