

## VisualC++,DX ライブラリの導入

ソフトゼミ A お疲れ様でした！今回からゼミ B に入ります。ゼミ B ではゼミ A で学んだ C 言語の知識を生かして実際に簡単なゲームを作成していきます。

### ➤ 初めに

---

ゼミ B では、ゼミ A で学んだ C 言語の基礎を土台に、Microsoft Visual C++ 2010(もしくは 2012) Express Edition(以下 VC++)の使い方と、DX ライブラリ(以下 DXLib)というゲーム制作を強力にサポートしてくれるプログラム群の使い方を学んでいきます。なお、VC++と DXLib を使うために、扱う言語は C ではなく C++になるのですが、DXLib を使用する分には C 言語の知識のみで使うことが可能(DXLib の内部では C++で動いていますが)なので、ゼミ B では C 言語的な書き方で書いていきます。ゼミ B では、2~5 回にシューティング、6~8 回にアクションゲームの制作を行います。

また、DXLib の使用で DXLib 内での関数が見えるようになりますが、その回の資料の最後に使用した関数の説明を書いておく予定です。説明を読んでもよくわからない場合は 2 年以上のソフト班員に気軽にご相談ください。何らかの力にはなれると思います。

### ➤ 注意(最初は読み飛ばしても構いません)

---

ゼミ B では、プログラムをわかりやすくするためにグローバル変数というどの関数からも使用可能な変数を用いていますが、本来は前回説明したポインタを用いたり、C++から追加されたクラスと呼ばれる機構を用いたりして変数の有効範囲を適切に保つべきです。もちろん、オブジェクト指向のプログラミングに慣れている場合は、使えるに越したことはないので、クラスでも何でも使ってください。あくまで、このゼミ B に出てくるプログラムはゼミ A の関数までの知識で書けるようにした「一例」です。

### ➤ Microsoft Visual C++ 2010 Express Edition の導入

---

ゼミ A の最終回でも VC++のインストール手順は示しましたが、もう一度載せておきます。既にインストールした方は読み飛ばしていいです。

<http://www.microsoft.com/visualstudio/jpn/downloads>

上のページから「Visual Studio 2010 Express」の中の「Visual C++ 2010 Express」をインストールしてきてください。また、間違えて「Visual C# 2010 Express」をインストールしないよう気を付けてください。

因みに、前回最後にも少し話しましたが、VC++2012でもゲームファイルは動作しますが、2010と2012の間ではプロジェクトファイルの互換がうまくいっていないようです。そのため、今後のゲーム制作で複数人での合作を考えている方はその点を覚えておいてください。また、2012以降の最新版も出ていますが、そちらの方はまだ動作確認を行っていません。問題なく動作するようだったら使用してください。

ところで、製品の登録はやっておきましたか？まだの場合は早めに登録しておいてください(要インターネット接続)。VC++の未登録での使用には日数制限があります。

## ➤ DXライブラリのダウンロード

---

こちらダウンロードまでの手順は前回の資料に掲載しましたが、再掲しておきます。

<http://homepage2.nifty.com/natupaji/DxLib/dxdload.html>

上のページから一番上にある「Visual C++用」をダウンロード→解凍しておいてください。解凍したフォルダはCドライブ直下などわかりやすい場所に置いておくことをお勧めします。

## ➤ プロジェクトの作成

---

VC++で「プロジェクト」を作成します。プロジェクトは今後VC++で1つのソフトウェアを作るたびに作成します。今回は、以下の手順で作成します。

1. VC++を起動します。デフォルトであれば、「スタート」→「すべてのプログラム」→「Microsoft Visual Studio (バージョン名) Express」→「VS Express for Desktop」  
で起動できるはずです。
2. VC++のウィンドウ上部、メニューバーの「ファイル(F)」から「新規作成(N)」  
→「プロジェクト(P)」を選択。
3. 出てきたウィンドウ左側「インストールされたテンプレート」で、「Visual C++」内の「Win32」を選択、右側で「Win32プロジェクト」を選択。ウィンドウ下部の「名前(N)」の欄に作るプロジェクトの名前を入力(例:zemi\_b)。右側「ソリューションのディレクトリを作成(D)」のチェックを外して OK

を押してください。

4. 「Win32 アプリケーション ウィザード」というウィンドウが出てくるので、「次へ」を押してください。
5. 「アプリケーションの種類」が「Windows アプリケーション(W)」になっていることを確認して、「追加のオプション」で「空のプロジェクト(E)」にチェックを入れ、「完了」を押してください。

## ➤ プログラムファイル(main.cpp)の追加

---

これから実際にプログラムを書いていくファイル「main.cpp」を作成します。先述の通り、C 言語的な書き方をしていきますが、DXLib の内部では C++が使われているので、拡張子は C++を表す「.cpp」となります。では、以下の手順で「main.cpp」を追加してみましょう。

1. VC++のウィンドウ上部、メニューバーの「プロジェクト(P)」から「新しい項目の追加(W)」を選択してください。
2. 出てきたウィンドウ左側「インストールされたテンプレート」で、「Visual C++」内「コード」を選択し、右側は「C++ファイル(.cpp)」を選択してください。ウィンドウ下部「名前(N)」には「main.cpp」と入力し、「追加(A)」を押してください。

## ➤ DX ライブラリの設定

---

このプロジェクトから DXLib を使用するための設定を行います。DXLib を使うプロジェクトでは、最初のプログラムファイル(ここでは main.cpp)作成後に必ず行ってください(main.cpp 作成前だと出てこない項目があります。)。やや長いですが、プロジェクト作成時(と、ほかの人からプログラムを受け取った時)のみなので頑張りましょう。

### ◆ 「プロパティページ」の出し方

まず、プロジェクトに関わる設定を変更できる画面「プロパティ ページ」を出します。VC++のウィンドウ上部、メニューバーの「プロジェクト(P)」から「(プロジェクト名)のプロパティ(P)」を選択します。(例えば、プロジェクト名が「zemi\_b」なら「zemi\_b のプロパティ(P)」と表示されます。) これでプロパティページが出てきます。

### ◆ 「プロパティ ページ」が出たらやること

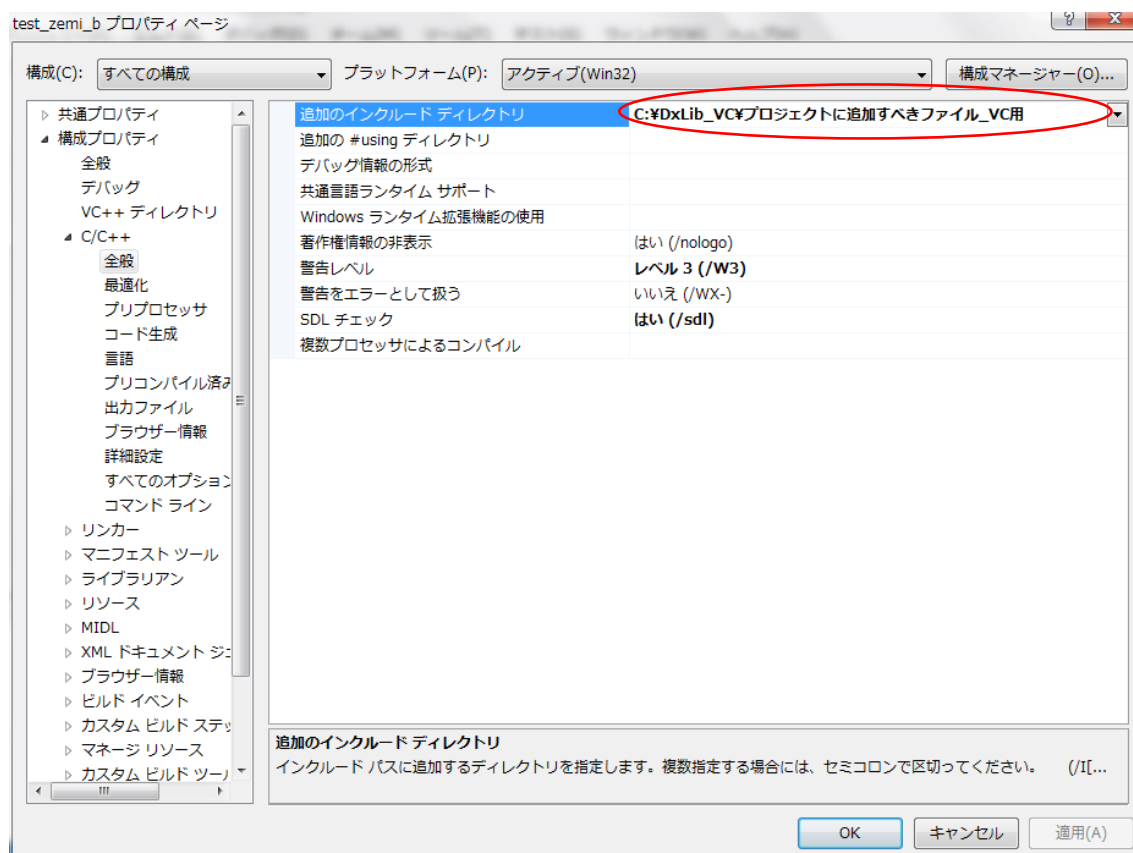
画面左側のリストから「構成プロパティ」を選んでください。

### ◆ 全ての構成の設定

開発中のプログラムを走らせるときは「Debug」と呼ばれる構成の設定を、実際に完成したプログラムを走らせるときには「Release」と呼ばれる構成の設定を利用するようになっているのですが、これからそれらの設定をしていきます。主に、DXLib がどこにあるのかを VC++ に伝える作業です。

1. 画面左上の「構成(C)」を「全ての構成」に切り替えます。
2. 画面左側のリストから「構成プロパティ」→「全般」を選んでください。
3. 画面右側「文字セット」の項目を「Unicode 文字セットを使用する」から「マルチバイト文字セットを使用する」に変更します。
4. ダイアログ右下にある「適用」ボタンを押します。
5. 画面左側のリストから「構成プロパティ」→「C/C++」→「全般」を選んでください。
6. 「追加のインクルードディレクトリ」の項目に DXLib のパッケージ内に入っている「プロジェクトに追加すべきファイル\_VC 用」フォルダへのパスを入力してから、再度ダイアログ右下にある「適用」ボタンを押します。

(入力例)



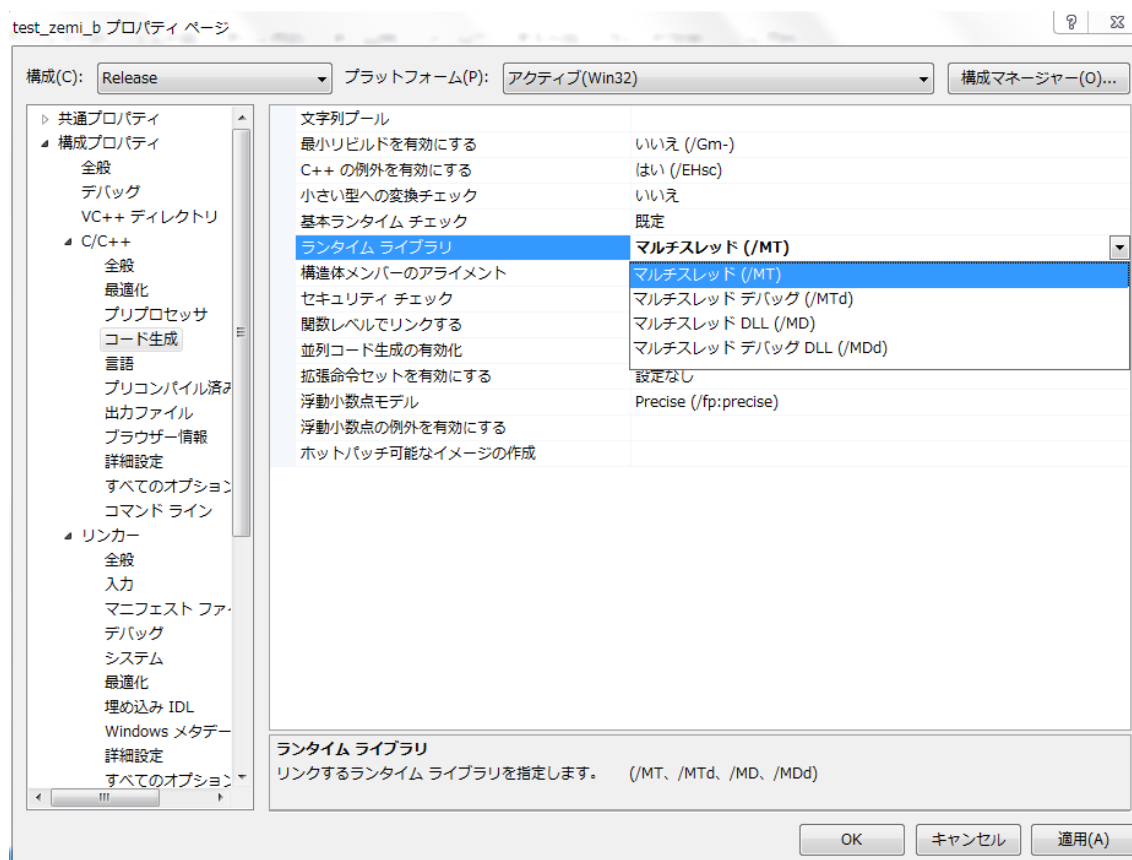
※ここでダウンロード時の DXLib の解凍先を忘れていたりフォルダの奥まったところにしてしまったりすると不便なので、DXLib のフォルダは C ドライブ直下において

おくことをお勧めします。

7. 画面左側リストから「構成プロパティ」→「リンカー」→「全般」を選んでください。
8. 「追加のライブラリディレクトリ」に先ほどと同じフォルダへのパスを入力してください。終わったらやっぱり「適用」ボタンを押します。

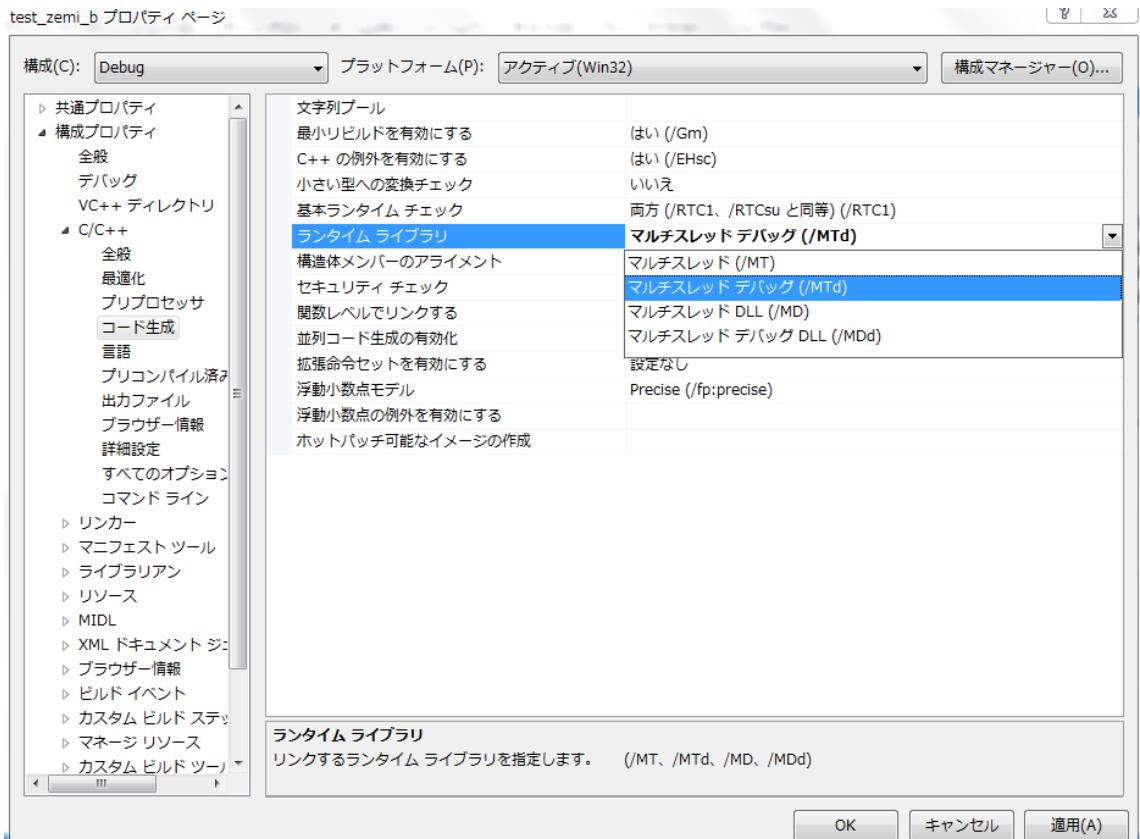
#### ◆ 構成「Release」の変更

1. 「構成(C)」を「Release」に変更します。
2. 画面左側のリストから「構成プロパティ」→「C/C++」→「コード生成」を選んでください。
3. 右側に表示されている「ランタイム ライブラリ」の項目を「マルチスレッド(MT)」に変更して「適用」。



#### ◆ 構成「Debug」の変更

1. 「構成(C)」を「Debug」に変更します。
2. 「ランタイムライブラリ」の項目を、今度は「マルチスレッド デバッグ(MTd)」に変更して「適用」。



以上で設定は終了です。「OK」を押すと、「プロパティ ページ」が閉じます。DXLib へのパスが異なる他の人からプログラムを受け取った時は「すべての構成」で DXLib のパスを上で述べた 2 か所を変更しなければならないので注意してください。

これでプログラムを書く準備ができました。いよいよです。

## ➤ プログラムを書く

---

```
#include"DxLib.h"
int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR
lpCmdLine,int nShowCmd){
//ウィンドウモードにする
ChangeWindowMode(TRUE);
//解像度とカラービット数を設定
SetGraphMode(640,480,32);
//DX ライブラリの初期化に失敗すると、即終了
if(DxLib_Init()==-1){return -1;}
//白い 32[px]*32[px]の正方形を描画
DrawBox(0,0,32,32,0xffffffff,TRUE);
//キーが押されるまで待機
WaitKey();
//DX ライブラリを終了する
DxLib_End();
return 0;
}
```

main.cpp で上記の内容を入力し、保存してください。

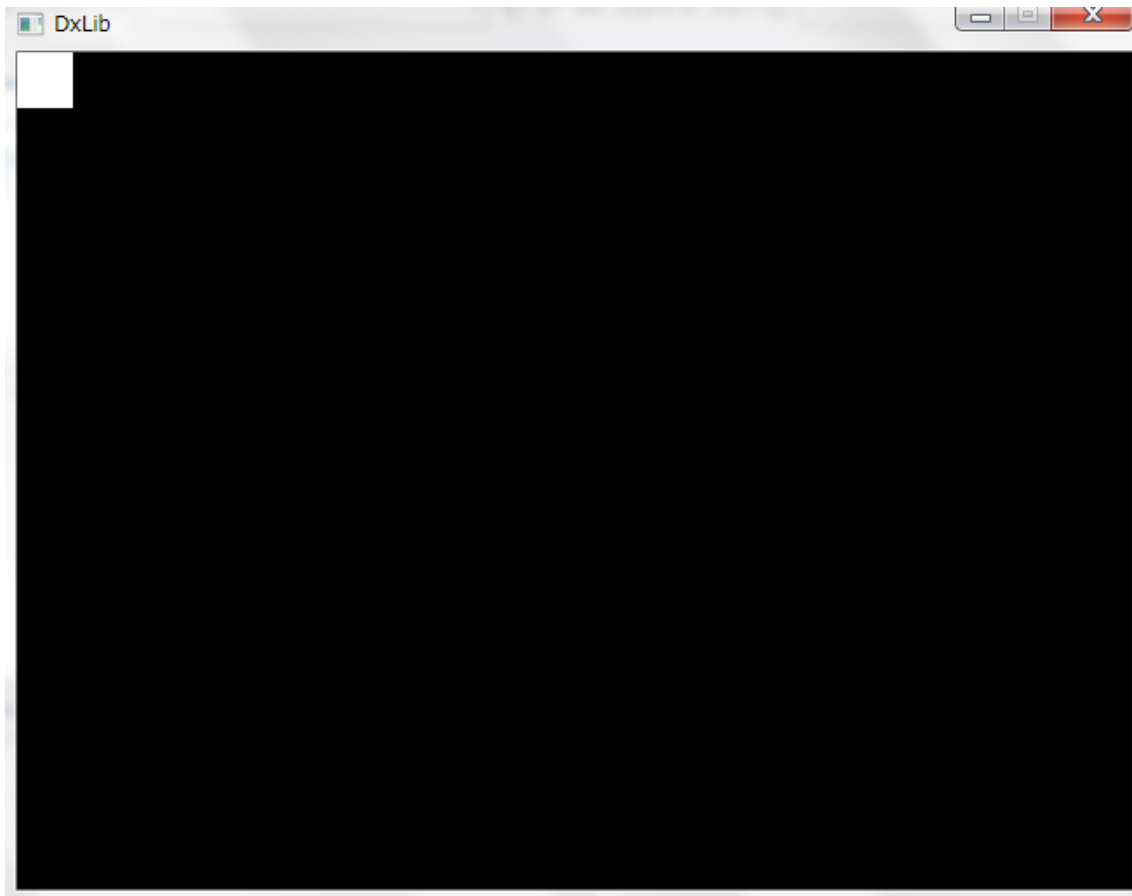
## ➤ コンパイル(ビルド)・実行

---

書いたコードをコンパイルします。画面上部の緑の「▶」を押すか、F5 キーでコンパイルすることができます。

コンパイル(ビルド)が終了すると、自動でプログラムが実行されます。次ページの図のように黒い画面の左上に白い正方形が表示されたら成功です。もし、「ビルドエラー」が出たら、もう一度プログラムを見直してみてください。VC++では、文法のミスなどを赤い下線で示してくれるので、確認してみましょう。

※以前にも書きましたが、大概この手のミスはセミコロンの抜け落ちや0,{の数の多すぎ少なすぎが原因であることが多いです。



➤ サンプルプログラムの大雑把な説明

---

◆ include”DxLib.h”

DX ライブラリを取り込みます。ちょうど、`#include<stdio.h>`で `printf` や `scanf` が使えるようになったように、DX ライブラリの関数を使うようにするには`#include”DxLib.h”`と書く必要があります。

◆ `int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,LPSTR lpCmdLine,int nShowCmd){`

非常に長いですが、これはちょうどゼミ A の C 言語で `int main(void){`と書いていたようなもので、最初に実行される関数です。DX ライブラリを使う場合含め、Windows アプリケーションで最初に実行されるのが `WinMain` 関数です。引数については、今は特に気にする必要はありません(ゼミ B では扱いません)。

DxLib の関数の一覧は、DxLib の入っているフォルダ内の「help」フォルダ内「`dxfunc.html`」、「`dxfunc2.html`」から参照できます。