

VisualC++・DX ライブラリの導入

ソフトゼミ A お疲れ様でした！今回からはゼミ B に入ります。
ゼミ B ではゼミ A で学んだ C 言語の知識を元にいよいよ実際にゲームを作っていきます。

◆ はじめに

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 Express Edition (以下、VisualC++, VC++) の使い方と、「DX ライブラリ」というゲームの作成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ゆきます。なお、VC++ と DX ライブラリを使うため、扱う言語は C 言語ではなく C++ となるのですが、DX ライブラリを使用する分には C 言語の知識のみで使用することが可能 (DX ライブラリの内部的には C++ で動いている) なので、ゼミ B では C 言語的な書き方で書いてゆきます。ゼミ B の 2~5 回では簡単なシューティングゲームの作成を、6~8 回では簡単なアクションゲームの制作を行います。DX ライブラリを用いることによって、DX ライブラリで用意されている関数を使うことができるのですが、各回の配布資料の末尾にはその回で新たに出てきた DX ライブラリの関数の説明を記しました。サンプルプログラムを読んで「ん!？」と思った場合には、資料の最後の方を見てみるといいかもしれません。それでも解決しない場合にはぜひ 2 年以上のソフト班員までどうぞ。それでも解決しない場合にはぜひ 2 年以上のソフト班員までどうぞ。

◆ 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(大域)変数」という、どの関数からでも使用可能な変数を用いていますが、本来はゼミ A 第 7 回で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よ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なんて場合は、使えるに越したことはないのですが、クラスでも何でもガリガリ使って構いません。あくまでも、このゼミ B で出てくるサンプルプログラムはゼミ A 第 6 回「関数」までの知識でも書けるようにした「一例」であることをご承知おき願いたいと思います。なお、C++ に関する内容は、夏休み中に予定しているゼミ C (任意参加) で扱う予定です。

◆ 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。

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

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つけてください。

<http://go.microsoft.com/?linkid=9709955>

また、直接ダウンロードする場合は上のURLにアクセスしてください。

ところで、製品の登録はやってきましたか？まだの場合は、早めに登録してください。(要インターネット接続) VisualC++の未登録での使用には日数制限があるため、インターネット接続があるうちに製品を登録しておくことを強く推奨します。(登録方法はゼミ A 最終回の資料をご覧ください。)

※以前に最新版の方の DL を指示したと思います。動作確認が出来ていませんが、動くようであれば使用してください。設定は基本的には 2010 版と同様と考えられます。

◆ DX ライブラリのダウンロード

こちらもダウンロードまでの手順はゼミA最終回に掲載しましたが、再掲。

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

から「VisualC++用」をダウンロード→解凍してください。

作ったプログラムを人と共有したりすることもあるので、

解凍して出てきたフォルダは C ドライブ直下に入れておくことをおすすめします。

◆ プロジェクトの作成

VC++で「プロジェクト」を作成します。このプロジェクトは、以降VC++で一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++を起動する。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 2010 Express」→「Microsoft Visual Studio 2010 Express」で起動できる。
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「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)」

明治大学エレクトロニクス研究部

ソフトゼミ B

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を押す。

4. 「Win32アプリケーション ウィザード」なるウィンドウが出てくるので、「次へ>」を押す。
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)」にチェックを入れる。その後、完了を押す。

◆ 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。「はじめに」で述べたとおり、C言語的な書き方をしていきますが、DXライブラリの内部的にはC++が使用されていますので、拡張子はC++を示す「.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。
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。右側は「C++ファイル(.cpp)」を選択。ウィンドウ下部「名前(N)」には「main.cpp」と入力し、「追加(A)」を押す。

◆ DX ライブラリの設定

このプロジェクトからDXライブラリを使用するための設定を行います。DXライブラリ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、他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、この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)」となります。) これでプロパティページが出てきます。

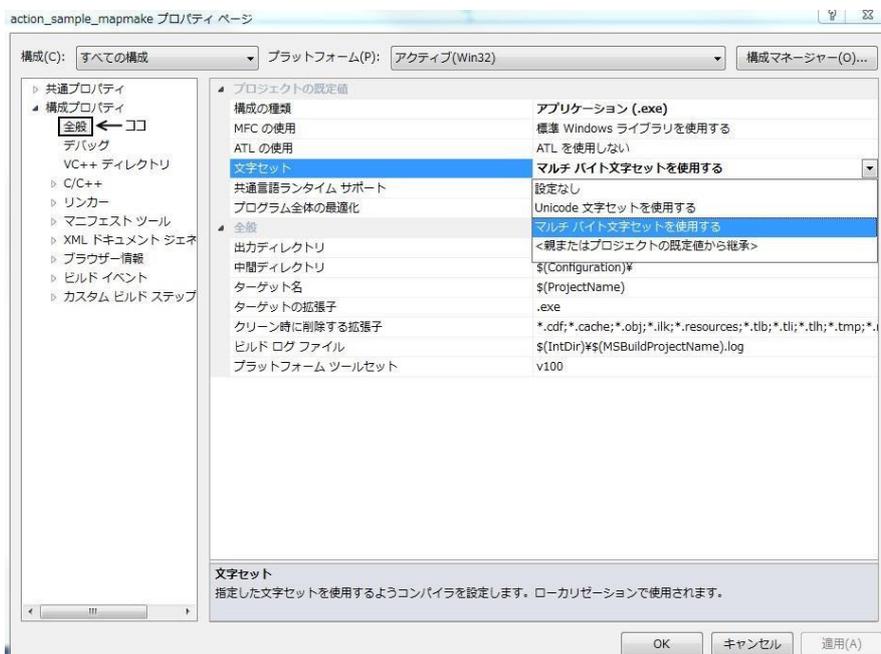
➤ 「プロパティ ページ」が出たらやること

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。

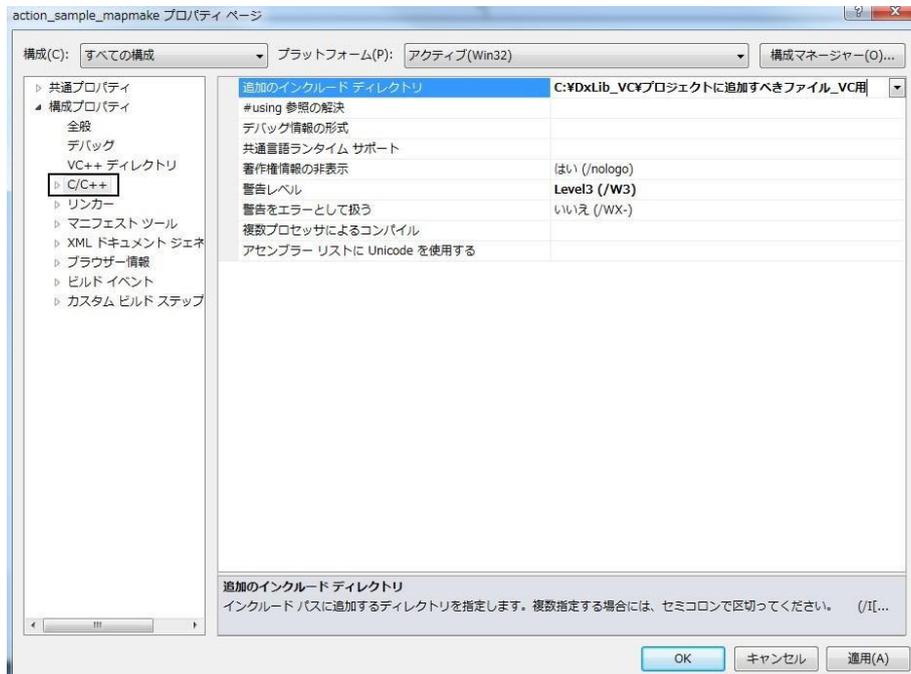
➤ すべての構成の設定

開発中にプログラムを走らせるときは「Debug」とよ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時には「Release」とよばれる構成の設定を利用するようになってはいるのですが、これから、それらの設定を行なっていきたいと思います。まずは、「Debug」と「Release」に共通する設定を設定していきます。主に、DX ライブラリがどこにあるかを VC++ に伝える作業です。

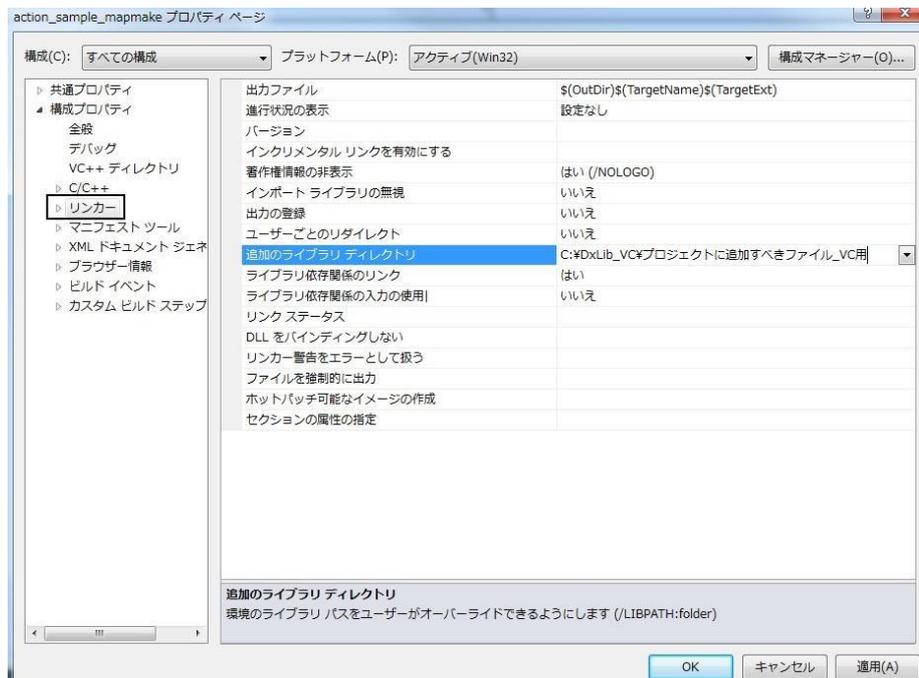
1. 画面左上の「構成(C)」を「すべての構成」に切り替えます。
2. 画面左側のリストから「構成プロパティ」→「全般」を選んでください。
3. 画面右側「文字セット」の項目を「Unicode 文字セットを使用する」から「マルチバイト文字セットを使用する」に変更します。



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 DX ライブラリ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。（パスの例 → C:\DxLib_VC\プロジェクトに追加すべきファイル_VC 用）

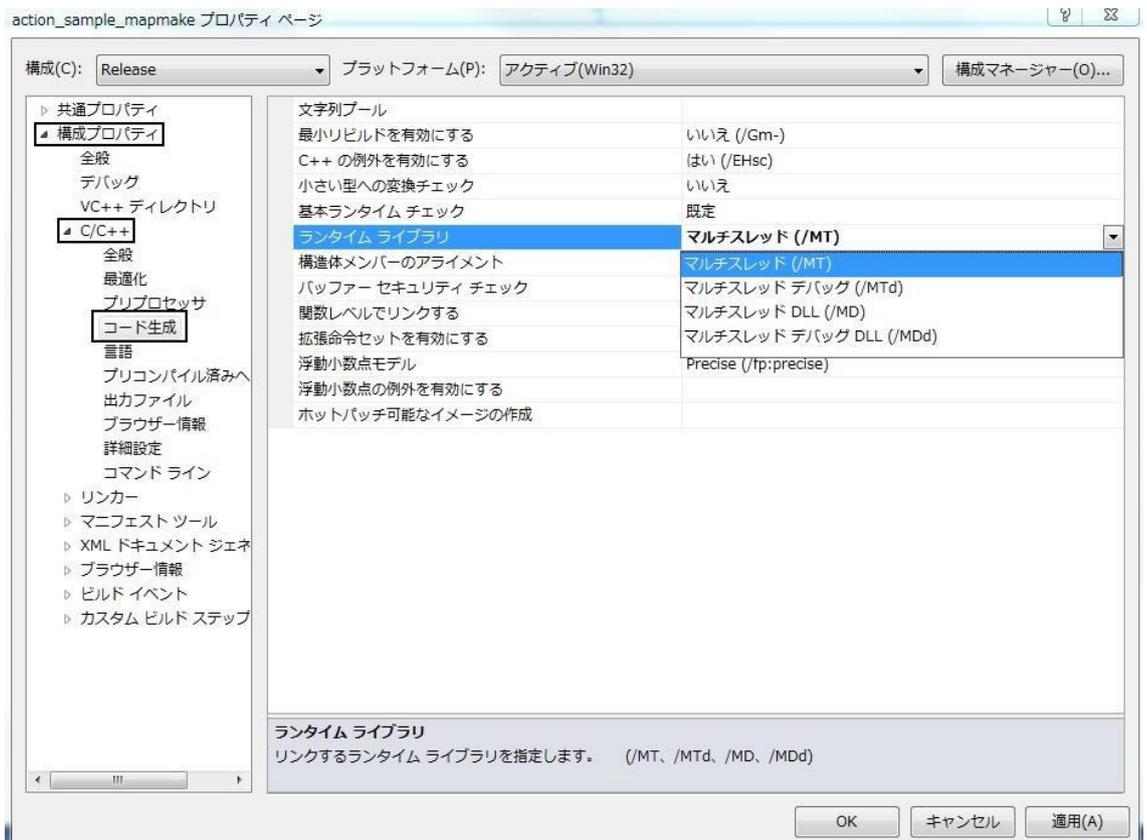


7. 画面左側のリストから「構成プロパティ」→「リンカー」→「全般」を選んでください。
8. 「追加のライブラリディレクトリ」に先ほどと同じフォルダへのパスを入力してください。終わったらやっぱり「適用」ボタンを押します。



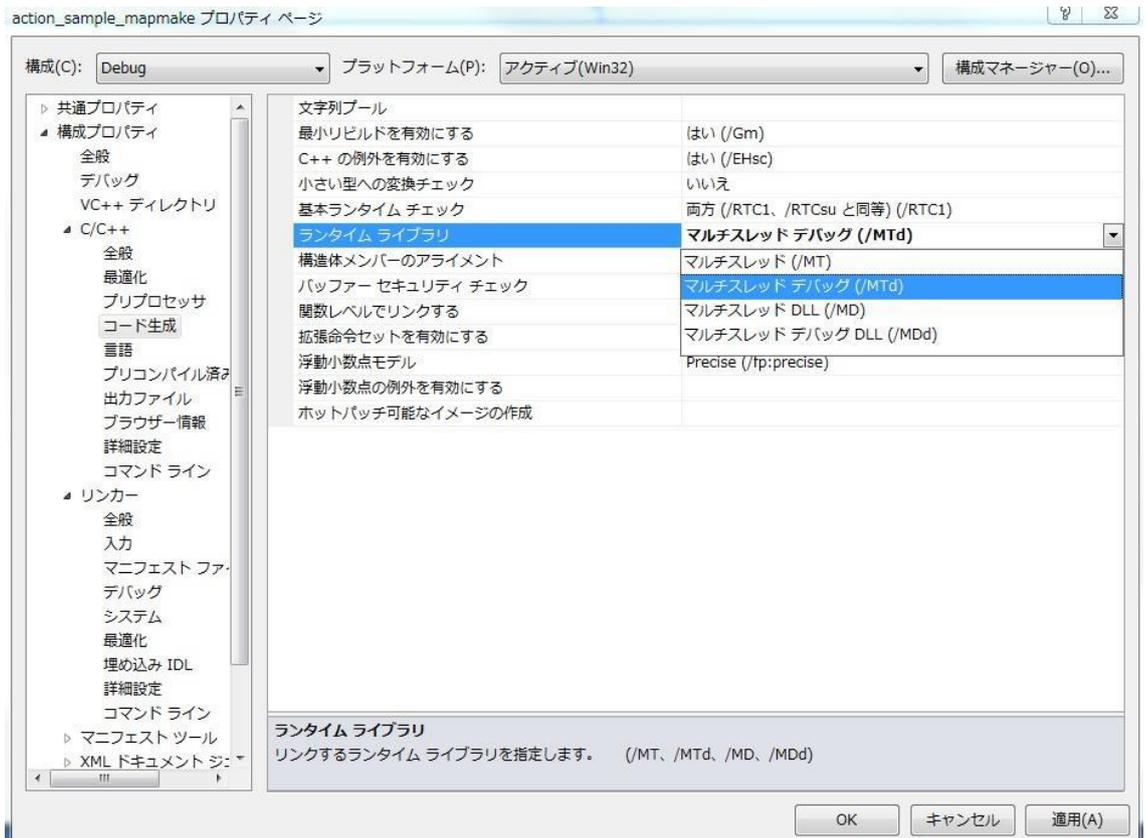
➤ 構成「Release」の変更

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。

DXライブラリ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DXライブラリのパスを上で述べた2箇所を変更しなければなりません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。画面上部の緑の三角印を押すか、**F5**キーでコンパイルを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。真っ黒な画面の左上隅に白い正方形が出れば成功です。もし、「ビルド エラー」が出た場合には、もう一度プログラムを見なおしてください。VC++では、文法上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。

◆ 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のと同じで、DX ライブラリの関数を使えるようにするためには `#include "Dxlib.h"`と書く必要があります。

➤ **int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nShowCmd)**

すごく長いですが、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、最初に行われる関数です。DX ライブラリを使う場合含め、Windows アプリケーションではじめに行われるのがこの `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)

DX ライブラリの関数の一覧は、DX ライブラリが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。