

D言語ゲーム開発入門

コレは何？

D言語勉強中のkenmoが、D言語でのゲーム開発情報をメモしていくページです。

- [目次](#)
-

- [D言語ゲーム開発入門](#)

- [コレは何？](#)
- [D言語の特徴](#)
 - [メリット](#)
 - [デメリット](#)
- [インストール ~ Hello, world.](#)
 - [コンパイラとリンカのインストール](#)
 - [Hello, world.](#)
- [SDLをインストール ~ 動かす](#)
 - [SDLを動かす \(画面を出す\)](#)
- [画像を表示](#)
- [マウスイベント取得](#)
- [マウスイベントの取得方法 2](#)
- [音楽の再生](#)
- [BGMとSEを同時に鳴らす](#)
- [コンソールを消す](#)
- [EXEファイルにリソースアイコンを埋め込む](#)
- [モジュール分割・ディレクトリ階層を作って配置する](#)
 - [Javaランタイムのインストール](#)
 - [Antのダウンロード・インストール](#)
 - [build.xmlを書く](#)
- [SDL小技](#)
- [参考 \(D言語\)](#)
- [参考 \(SDL\)](#)
- [コメント](#)

D言語の特徴

メリット

- ランタイムのいらない、コンパクトな実行ファイルが作れる

JavaとかC#とかRubyとかPythonにはないメリットだぞ。

- ガベコレがある

C/C++のメモリ管理に泣かされていた日々とはサヨナラです。

- クラスを使わなくても良い

Cライクな書き方でもOK！モダンな書き方もできるぞ！

- コンパイルが爆速

スクリプト言語なみに速いぞ

デメリット

- 情報が少ない

ある程度出揃っていますが、まだ初心者向きの情報がないです

- 未だベータ版

2006/10/23現在のコンパイラのバージョンは、0.172です。仕様が結構変わる。先は長いぞー（古いサンプルコードがビルドできないことがよくある）

- ライブラリ作るのがめんどい

SDLなりOpenGLなりを簡単に扱えるライブラリを自作しないと、ゲーム作るのは大変。その苦労はC/C++とかと一緒に。

- 統合環境がまだまだ

VisualStudioやEclipseと比べると、デバッガや定義ジャンプ、リファクタリングがやりにくい・できないのが少しつらいです。

今のところ、結局テキストエディタでコード書いて、コマンドプロンプトからビルド、、という流れに落ち着きます。

個人的には、C+SDLとか、C+DirectXとか、C+OpenGLなりやっていないとツライかな、という気がします。

ということで、そこらへんをやったことある人が

「そろそろD言語はじめっかなー？」

と思っている人が対象の内容となっております。

インストール～Hello, world.

とりあえず、インストールして「Hello, world.」を出すまで。

コンパイラとリンカのインストール

<http://www.kmonos.net/alang/d/dcompiler.html>

ここから、

- Win32版 dmd.zip (D コンパイラ)

と

- Win32版 dmc.zip (リンカなどの小物ツール)

をダウンロード。

dmd.zipを解凍すると、「dm」と「dmd」ができる。
とりあえず、「C:\work\D\」に置いてみます。

(「デスクトップ」とか「C:\Program Files」とかの日本語パス、半角スペースがあるところに
置いちゃダメ)

dmc.zipを解凍すると、「dm」ができる。それを先ほどの「dm」に上書き。

これでインストール完了。

Hello, world.

```
void main()
{
    printf("Hello, world.");
}
```

こんなプログラムを書いて、「test.d」で保存。

```
SET PATH=C:\work\D\dmd\bin\;C:\work\D\dm\bin\;%PATH%
dmd test.d
```

pause

で、こんなバッチファイル(例えば「build.bat」)を書いて保存。

バッチファイルをダブルクリックで、コンパイル。

「test.exe」ができれば、ビルド完了。

コマンドプロンプトから、「test.exe」を実行すると、

Hello, world.

と出れば、OKです。

SDLをインストール～動かす

[D - porting](#)から、「SDL. Based on DedicateD's.」の「SDL」のリンクをぺちっと押して、「SDL.zip」をダウンロード。

解凍してできた「SDL」フォルダを「C:\work\D\」に配置。

「SDL」フォルダ内にある、「SDL.dll」と「SDL.lib」をコピーして、「C:\work\D\」に移動します。

これでインストール完了。

SDLを動かす(画面を出す)

```
import SDL;
void main()
{
    // 初期化ー
    if( SDL_Init(SDL_INIT_VIDEO) < 0 ) {
        throw new Error("Couldn't initialize SDL");
    }

    // 640x480で、デフォルトのbpp。ソフトウェアで頑張る
    SDL_Surface *screen = SDL_SetVideoMode(640, 480, 0, SDL_SWSURFACE);
    if ( screen == null ) {
        throw new Error("Couldn't set 640x480 video mode");
    }
}
```

```

        // イベントループ
bool done = false;
SDL_Event e;
while ( !done ) {
    while ( SDL_PollEvent(&e) ) {
        done = e.type == SDL_QUIT;
    }
    SDL_Delay(1000/30);
}
SDL_Quit();
}

```

これを「test.d」で保存。

(文字コードを「UTF-8」にして保存します)

```

SET PATH=C:\work\D\dmd\bin\;C:\work\D\dm\bin\;%PATH%
dmd test.d -ISDL SDL.lib
pause

```

こんなバッチファイルを書いて、ビルド。

ちなみに、

```
dmd test.d -ISDL SDL.lib
```

の部分は、

「-ISDL」はインポートディレクトリ「SDL」の指定で、
「SDL.lib」はライブラリをスタティックリンクしています。

それで、できたtest.exeを実行して、画面が出れば、とりあえずOKです。

画像を表示

BMPを読み込んで描画します。(エラー処理は省略)

```

import SDL;
void main()
{
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Surface *screen = SDL_SetVideoMode(640, 480, 0, SDL_SWSURFACE);

    // ビットマップ読み込み
    SDL_Surface *image = SDL_LoadBMP("hell.bmp");
    // 転送
    SDL_BlitSurface(image, cast(SDL_Rect*)0, screen, cast(SDL_Rect*)0);

    // イベントループ
    bool done = false;
    SDL_Event e;
    while ( !done ) {
        while ( SDL_PollEvent(&e) ) {
            done = e.type == SDL_QUIT;
        }
        SDL_Delay(1000/30);
    }
    SDL_Quit();
}

```

マウスイベント取得

```
import SDL;
void main()
{
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Surface *screen = SDL_SetVideoMode(640, 480, 0, SDL_SWSURFACE);

    // イベントループ
    bool done = false;
    SDL_Event e;
    while ( !done ) {
        while ( SDL_PollEvent(&e) ) {
            switch (e.type) {
                // マウス押下イベント取得
                case SDL_MOUSEBUTTONDOWN:
                    switch (e.button.button)
                    {
                        case SDL_BUTTON_LEFT:
                            SDL_WM_SetCaption("push left", null);
                            break;
                        case SDL_BUTTON_RIGHT:
                            SDL_WM_SetCaption("push right", null);
                            break;
                        default:
                            break;
                    }
                    break;
                case SDL_QUIT:
                    done = true;
                    break;
                default:
                    break;
            }
        }
        SDL_Delay(1000/30);
    }
    SDL_Quit();
}
```

マウスイベントの取得方法 2

わんきちさんのリクエストにお答えして、マウスのイベントをコールバックで取る方法を、SDL_SetEventFilterを使ってみる。

```
import SDL;

extern (C) int filterEventsHandler(SDL_Event* e)
{
    if ( e.type == SDL_MOUSEBUTTONDOWN ) {
        if ( e.button.button == SDL_BUTTON_LEFT ) {
            printf("MOUSEBUTTONDOWN --- SDL_BUTTON_LEFT \n");
            return 0; /* すでに処理が終わったので捨てます */
        }
    }
    return 1;
}
```

```

}

void main()
{
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Surface *screen = SDL_SetVideoMode(640, 480, 0, SDL_SWSURFACE);

    SDL_SetEventFilter(&filterEventsHandler);

    // イベントループ
    bool done = false;
    SDL_Event e;
    while ( !done ) {
        while ( SDL_PollEvent(&e) ) {
            done = e.type == SDL_QUIT;
        }
        SDL_Delay(1000/30);
    }
    SDL_Quit();
}

```

音楽の再生

生SDLだとしんどいので、SDL_mixerを使用します。

やね先生のサイト「<http://www.sun-inet.or.jp/~yaneurao/dlang/Chapter-10.html#100100000000>」から、

「<http://www.sun-inet.or.jp/~yaneurao/dlang/lib/yaneSDK4D0401280603.zip>」をダウンロード。

「yaneSDK4D0401280603\SDL\SDL_mixer.d」を「SDL」フォルダに、

「yaneSDK4D0401280603\import\SDL_mixer.lib」を同一ディレクトリに入れます。

「yaneSDK4D0401280603\SDL_mixer.dll」を同一ディレクトリに入れます。

"chimes.wav"を鳴らします。

```

import SDL;
import SDL_mixer; // SDL_mixerを使うよー
void main()
{
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Surface *screen = SDL_SetVideoMode(640, 480, 0, SDL_SWSURFACE);

    // 44100Hz, 16bit, ステレオ, バッファサイズ4092byte
    Mix_OpenAudio(44100, AUDIO_S16, 2, 4092);
    Mix_Music* music = Mix_LoadMUS("chimes.wav");
    // 再生。2つ目の引数に再生回数を指定。-1だと無限ループ
    if (Mix_PlayMusic(music, -1) != 0) {
        throw new Error("Couldn't play music");
    }
    // イベントループ
    bool done = false;
    SDL_Event e;
    while ( !done ) {
        while ( SDL_PollEvent(&e) ) {
            done = e.type == SDL_QUIT;
        }
        SDL_Delay(1000/30);
    }
    // 演奏停止
    Mix_HaltMusic();
}

```

```

        // 曲データ開放
        Mix_FreeMusic(music);
    // サウンドデバイスクローズ
    Mix_CloseAudio();
    SDL_Quit();
}

```

バッチファイルはこんな感じ。

```

SET PATH=C:\work\D\dmd\bin\;C:\work\D\dm\bin\;%PATH%
dmd test.d -ISDL SDL.lib SDL_mixer.lib

```

pause

「SDL_mixer.lib」を追加します。

あと、サウンドフォーマットは他に「Ogg」も対応しているみたいです。

BGMとSEを同時に鳴らす

チャンクを使います。

```

import SDL;
import SDL_mixer; // SDL_mixerを使うよー
void main()
{
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Surface *screen = SDL_SetVideoMode(640, 480, 0, SDL_SWSURFACE);

    // 44100Hz, 16bit, ステレオ, バッファサイズ4092byte
    Mix_OpenAudio(44100, AUDIO_S16, 2, 4092);
    // BGM
    Mix_Music* music = Mix_LoadMUS("bgm.ogg");
    if (!music) {
        throw new Error("Couldn't load music");
    }
    // 再生。2つ目の引数に再生回数を指定。-1だと無限ループ
    if (Mix_PlayMusic(music, -1) != 0) {
        throw new Error("Couldn't play music");
    }
    // SE
    Mix_Chunk* chunk = Mix_LoadWAV("chimes.wav");
    if (!chunk) {
        throw new Error("Couldn't load chunk");
    }

    // イベントループ
    bool done = false;
    SDL_Event e;
    while ( !done ) {
        while ( SDL_PollEvent(&e) ) {
            switch (e.type) {
                case SDL_MOUSEBUTTONDOWN:
                    // 再生する。第1引数はチャンネル番号(-1で自動割り当て)
                    // 第3の引数には再生回数(「0」は1回再生。「-1」なら無限ループ)
                    Mix_PlayChannel(-1, chunk, 0);
                    break;
                case SDL_QUIT:
                    done = true;
                    break;
                default:

```

```

        break;
    }
}
SDL_Delay(1000/30);
}
// サウンド停止
Mix_HaltChannel(-1);
Mix_HaltMusic();
// サウンドデータ開放
Mix_FreeChunk(chunk);
Mix_FreeMusic(music);
// サウンドデバイスクローズ
Mix_CloseAudio();
SDL_Quit();
}

```

ただ、このままビルドしても、

```

test.obj(test)
Error 42: Symbol Undefined _Mix_LoadWAV
test.obj(test)
Error 42: Symbol Undefined _Mix_PlayChannel
--- errorlevel 2

```

というリンカエラーが出てしまいます。

理由は良く分らないですが、「SDL_mixer.d」のオブジェクトファイルを作ってリンクするとうまくいきます。

SDLフォルダから「SDL_mixer.d」を引っ張ってきて、

```

dmd -c -ISDL SDL_mixer.d

```

とすると、「SDL_mixer.obj」ファイルができます。

そして、

```

dmd test.d -ISDL SDL.lib SDL_mixer.lib SDL_mixer.obj

```

こんな感じで「SDL_mixer.obj」をリンクするとビルドできます。

コンソールを消す

コンソールを消すには以下のリンカオプションを指定すると消すことができます。

```

-L/exet:nt/su:windows:4.0

```

例えば、こんなバッチファイルでビルドするとコンソールが消えてくれます。

```

SET PATH=C:\work\D\dmd\bin\;C:\work\D\dm\bin\;%PATH%
dmd -L/exet:nt/su:windows:4.0 -ISDL test.d SDL.lib
pause

```

EXEファイルにリソースアイコンを埋め込む

D言語あんまり関係ないですが、念のため。
やり方は、「.res」ファイルを作ってリンクするだけです。

で、「.res」をどうやって作るのかというと、
VisualStudioとか持っていれば、そこからぺちぺち操作すれば作れます。

「そんなのないよ！」という人はここから、

[Borland C++ Compiler 5.5無償ダウンロード](#)
Borland C++ Compiler 5.5をダウンロード。

インストーラー形式なので、「OK」「完了」でインストール完了。

リソースをビルドするには、「.rc」が必要なので、

```
test ICON PRELOAD "test.ico"
```

と書いて、「test.rc」で保存。

```
SET PATH=C:\borland\bcc55\Bin;%PATH%
```

```
brcc32 test.rc
```

```
pause
```

こんなバッチファイルを書いて、「brcc32.exe」で「test.rc」をビルドすると、「test.res」ができません。

で、

```
dmd -ISDL test.d SDL.lib test.res
```

というように、「test.res」をリンクすれば、EXEファイルにアイコンが埋め込まれます。

モジュール分割・ディレクトリ階層を作って配置する

1ファイルで頑張っても良いのですが、

Javaっぽく1ファイル1クラス、ディレクトリ階層を使ったりしたい人のために、

ビルドには「Ant」というツールを使います。

Javaランタイムのインストール

ただ、「Ant」を使うには、Javaのランタイム（JRE）をインストールする必要があります。

<http://www.java.com/ja/>

「Ant」の機能を全部使うにはJDKを入れなければならないそうですが、とりあえず基本的なタスクしか使わないので、これでOK。

Antのダウンロード・インストール

次にAntをダウンロード。

<http://ant.apache.org/bindownload.cgi>

から、apache-ant-1.6.5-bin.zipをダウンロードします。

解凍したら、フォルダ名を「ant」に変えて、とりあえず「C:\work\D\」に置いてみます。

build.xmlを書く

「build.xml」とは、Antを使ってビルドするときの設定ファイルみたいなものです。

```
<!-- Antを使ってビルドするよー (http://www.jajakarta.org/ant/ant-1.6.1/docs/ja/) -->
```

```
<project name="d_build" default="all" basedir=". ">
```

```
  <!-- ターゲット名 (実行ファイル名) -->
```

```
  <property name="name" value="test"/>
```

```
  <property name="prog" value="${name}.exe"/>
```

```
  <!-- ライブラリディレクトリ -->
```

```
  <property name="libdir" value="lib \"/>
```

```
  <!-- ここに使うライブラリを追加します -->
```

```
  <property name="libs" value="${libdir}SDL.lib"/>
```

```
  <!-- 「.d」のあるインポートディレクトリ -->
```

```
  <property name="import" location="SDL"/>
```

```
  <!-- ソースファイルのディレクトリ -->
```

```
  <property name="src" location="src"/>
```

```
  <!-- リソースのディレクトリ -->
```

```
  <property name="resource" location="resource"/>
```

```
  <!-- インポートライブラリのディレクトリ -->
```

```
  <property name="builtimport" value="SDL_mixer.d"/>
```

```
  <!-- 全部ビルド -->
```

```
  <target name="all" depends="compile, link"/>
```

```

    <target name="rebuild" depends="clean, compile, link"/>
    <target name="compile">
<apply executable="dmd" dir="${src}" dest="${src}" parallel="true" failonerror="true">
    <mapper type="glob" from="*.d" to="*.obj"/>
    <fileset dir="${src}" includes="**/*.d"/>
    <arg value="-c"/>
    <arg value="-I${import}"/>
    <arg value="-op"/>
    <arg value="-O"/>
    <arg value="-release"/>
    <arg value="-version=Win32_release"/>
    <srcfile/>
</apply>
</target>
<target name="link">
    <apply executable="dmd" dir="." parallel="true" failonerror="true">
    <fileset dir="${src}" includes="**/*.obj"/>
    <fileset dir="${import}" includes="**/*.obj"/>
    <fileset dir="${resource}" includes="test.RES"/> <!-- アイコンリソース -->
    <fileset dir="${resource}" includes="test.def"/> <!-- コンソールを表示させない定義 -->
    <arg value="${prog}"/>
    <arg value="${libs}"/>
    <srcfile/>
</apply>
</target>

<!-- クリーン -->
<target name="clean">
    <delete file="${prog}"/>
    <delete file="${name}.map"/>
    <delete>
        <fileset dir="${src}" includes="**/*.obj"/>
    </delete>
</target>

<!-- 実行 -->
<target name="run">
    <exec executable="${name}" dir=".">
    </exec>
</target>

<!-- インポートディレクトリのモジュールをビルド -->
<target name="buildlib">
    <apply executable="dmd" dir="${import}" dest="${import}" parallel="false" failonerror="true">
    <mapper type="glob" from="*.d" to="*.obj"/>
    <fileset dir="${import}" includes="${builtimport}"/>
    <arg value="-c"/>
    <arg value="-I${import}"/>
    <arg value="-op"/>
    <srcfile/>
</apply>
</target>
</project>

```

これを「build.xml」という名前で保存します。

さて、この設定ファイルを使う前に以下の準備をしておきます。

ソースコードは「src」ディレクトリに入れる（src\test.d）

- インポートモジュール（SDL関連）は「SDL」ディレクトリに入れる（そのまま）
- ライブラリは「lib」ディレクトリに入れる（lib\SDL.libなど）
- リソース（.def/.res）は「resource」ディレクトリに入れる

それで、こんなバッチファイルを実行すると、

```
SET PATH=C: \ work \ D \ ant \ bin;%PATH%
```

```
SET PATH=C: \ work \ D \ dmd \ bin \ ;C: \ work \ D \ dm \ bin \ ;%PATH%
```

```
ant
```

```
pause
```

「src」以下のディレクトリ内のソースコードを自動でビルドしてくれます。

（ Antはデフォルトで、カレントディレクトリ内の「build.xml」を探しに行ってくれるため）

ただ、Antは終了するとpauseしてくれないので、コマンドプロンプトから、

```
ant
```

と入力した方が良さそうです。

あと、

```
ant clean
```

と入力すると、「.obj/.map/.exe」を全て削除してくれます。

D言語は.objの依存関係が強すぎるのか、EXEファイルを実行して、不明なエラーが出る場合は、「ant clean」して、ビルドし直すと、うまくいく場合があります。

ほかに

```
ant run
```

すると、実行ファイルを実行してくれたり、

```
ant buildlib
```

すると、インポートディレクトリ内の「.d」をコンパイルしてくれます。（この設定では「SDL_mixer.d」のみ）

Antの詳細については、

- [Apache Ant 1.6.1 マニュアル](#)

が参考になると思います。

SDL小技

これもD言語関係ないですが、SDLを使った、あると嬉しい機能のサンプルを、

- ESCキー：終了
- F5キー：ボスが来た！モード
- F9キー：フルスクリーン・ウィンドウモード切替
- F10キー：マウスカーソル表示切替
- F12キー：スクリーンショット保存

フルスクリーン・ウィンドウモード切替の処理については、リアルタイムゲームだと、毎フレーム再描画するので、バッファへの書込みは必要ないかもしれません。

```
import SDL;
```

```
static int g_width = 640;
```

```

static int g_height = 480;
static int g_videoBpp = 0;
static int g_videoFlags = SDL_SWSURFACE;

/**
 * フルスクリーンモード切替
 */
void toggleFullScreen(SDL_Surface* screen, SDL_Surface* buffer)
{
    // バッファに保存
    SDL_BlitSurface(screen, cast(SDL_Rect*)0, buffer, cast(SDL_Rect*)0);
    g_videoFlags ^= SDL_FULLSCREEN; // フルスクリーンフラグ反転
    // スクリーン生成
    screen = SDL_SetVideoMode(g_width, g_height, g_videoBpp, g_videoFlags);
    // 再描画
    SDL_BlitSurface(buffer, cast(SDL_Rect*)0, screen, cast(SDL_Rect*)0);
}

/**
 * マウスカーソル表示切替
 */
void toggleShowCursor()
{
    if(SDL_ShowCursor(SDL_QUERY) == SDL_ENABLE) SDL_ShowCursor(SDL_DISABLE);
    else SDL_ShowCursor(SDL_ENABLE);
}

void main()
{
    SDL_Init(SDL_INIT_VIDEO);
    SDL_Surface* screen = SDL_SetVideoMode(g_width, g_height, g_videoBpp, g_videoFlags);
    // バックバッファ生成
    SDL_Surface* buffer = SDL_ConvertSurface(screen, screen.format, g_videoFlags);
    // 赤い四角を描いてみる
    SDL_Rect rect;
    rect.x = 100; rect.y = 200; rect.w = 200; rect.h = 100;
    SDL_FillRect(screen, &rect, SDL_MapRGB(screen.format, 0xff, 0x00, 0x00));
    // イベントループ
    bool done = false;
    SDL_Event e;
    while ( !done ) {
        while ( SDL_PollEvent(&e) ) done = e.type == SDL_QUIT;
        int numkeys;
        Uint8* keystate = SDL_GetKeyState(&numkeys);
        if(keystate[SDLK_F9]) toggleFullScreen(screen, buffer); // フルスクリー
        else if(keystate[SDLK_F10]) toggleShowCursor(); // マウスカーソ
        else if(keystate[SDLK_F12]) SDL_SaveBMP(screen, "screenshot.bmp"); // スクリーンシ
        else if(keystate[SDLK_F5]) SDL_WM_IconifyWindow(); // ボスが来た！
        else if(keystate[SDLK_ESCAPE]) done = true; // 終了
        SDL_Delay(1000/30);
    }
    SDL_Quit();
}

```

参考 (D言語)

- [プログラミング言語 D](#)--D言語リファレンスマニュアルです。
- [D言語研究室](#)--やね先生のD言語研究室。マニアックな情報がたっぷりあります
- [D言語研究](#)--D言語のWikiページ
- [D言語入門](#)--D言語の文法について細かく書いてあります
- [D Memo](#)--これも、D言語の文法について。情報がやや古いですが、参考になる内容も多いです
- [ABA Games](#)--D言語で作ったゲームがたくさん。ソースコードを公開されているので、それが膨大なドキュメントです。
- [MacでSDLアプリケーションの作り方](#)
 - MacOSXで動かしたい人向け。基本.app形式を作成することになります。

参考 (SDL)

- [SDLドキュメント \(日本語訳\)](#)

コメント

なにかあればコメントをどうぞ。

- マウスイベント取得はポンプでくみ上げる以外の方法ってあるんですよね？自分で調べればいいんだけど...マウスイベントハンドラの実装例をおねがいします -- わんきち (2006-10-24 13:19:26)
- 初回コメント特典でやり方を書いておきましたー -- kenmo (2006-10-24 15:20:03)
- やり方ありがとうございます。続けて質問、SDL_MOUSEBUTTONDOWNにあたる部分が見当たらないのですが、、ああ今度こそ自分で探しますよw -- わんきち (2006-10-24 17:04:36)
- SDL_GetMouseStateするためのトリガーとして最低SDL_MOUSEBUTTONDOWNは拾ってことですね -- わんきち (2006-10-24 17:14:07)
- 何が言いたかったかっていうと、Javaのイベントリスナーみたいな実装が希望だったんですが、自分でやれてことですね、うん -- わんきち (2006-10-24 17:20:05)
- あ、勘違いしてました...。うーん、SDL_PeepEventsする処理をSDL_AddTimerすればいいのかなー -- kenmo (2006-10-24 18:16:44)
- ふー、頑張って取れたよ。>マウスイベントハンドラ -- kenmo (2006-10-24 20:16:34)
- おお、ありがとうございます。タイマースレッドで管理ですね。D言語の精度がどれくらいなのかしらないのでなんともなんですが、ご存知のとおり、タイマーはタイマーイベントをメッセージキューに投げるだけなので緊急度の高いメッセージがキューに投げ込まれると後送りにされてしまいます。できればスレッド制御にしたいですね。あとで調べてみます。スレッド版。 -- わんきち (2006-10-24 21:24:32)
- yaneSDK4Dはだいぶ古くなってるので <http://shinh.skr.jp/> のD言語始めました、からインポートヘッダ取ってきたほうがいいかもしれません。importの仕様とか結構変わったので。 -- みかげ (2006-10-24 22:34:20)
- あと、SDL.dllもバージョンアップしてるみたいなんで本家から取ってきて上書きするといいかもしれません。SDL_SetVideoMode() の引数で0入れるとウィンドウがデスクトップのサイズになったりとか。 -- みかげ (2006-10-24 22:38:59)
- あうあー、早とちりだー書いてあるのに・・・申し訳ないです。 -- みかげ (2006-10-24 22:45:31)
- <http://zinnia.dyndns.org/~cvswb/slddoc-jp/guideeventexamples.html> 試していないのですが、もっとシンプルにSDL_SetEventFilterでマウスイベントを別スレッドでとることができるみたいですね。 -- kenmo (2006-10-25 19:13:58)
- SDL_SetEventFilterで処理する方法を追加しました。 -- kenmo (2006-10-25 20:11:32)

おお、まさにハンドラなハンドラ！コレを期待してました。ありがとうございます。 -- わん

- きち (2006-10-25 21:26:39)
- 小技便利ですね。実装させていただきます -- HIZ (2006-10-26 20:38:23)
- ディレクトリ名にスペースが入っていても、dos形式の方のパスにcdしてからantを起動すればなんとかなるようです(C:\Documents and.. -> c:\docume~1 みたいに) -- zzz (2008-07-20 20:30:06)

名前:

コメント:

投稿