

卒業論文

DirectX と Wine の互換性

2009年3月7日

函館工業高等専門学校
東海林研究室
5年 情報工学科 大川 貴嗣

目次

1章. はじめに	
1-1. 目的	・・・ 2
1-2. 本論文の構成	・・・ 3
2章. Wine とは	・・・ 4
3章. Direct X	・・・ 5
4章. ソフトウェア仕様	
4-1. 操作方法	・・・ 6
4-2. 画面説明	・・・ 6
4-3. ゲームフロー	・・・ 8
5章. 開発環境	・・・ 9
6章. Wine での動作実験	・・・ 10
7章. 問題点	・・・ 12
8章. 考察	・・・ 13
9章. まとめ	・・・ 14
参考文献	

1 章. はじめに

1-1. 目的

近年、Linux ディストリビューションの増加に伴ってユーザの増加も進んでいる。しかし、現在の OS のシェアは Microsoft 社の Windows がトップを占めており[1][2]、当然の事ながら Windows で動作するソフトウェアも非常に多い状態である。そういった背景があるため、Linux 上でも Windows のソフトウェアを動作することが出来ればユーザにとって非常に有益である。それを実現する方法として、現在いくつかの選択肢が存在する。

一つ目として VMware や Xen や QEmu の様に仮想マシンを使った方法がある[3][4][5][6]。これは Linux 上で仮想 Windows マシンを構築する技術であり、多くのリソースを使用する。二つ目の方法として仮想マシンではなく互換レイヤーとして動作する Wine を用いる方法がある[7][8]。本研究では Wine について取り扱う。

Wine は現在オープンソース[9][10][11]での開発が進められているソフトウェアである。Wine はソースコードが公開されており、誰でも自由にその派生物を作成、公開する事が可能である。本研究では Wine の問題点を発見し改善することで Wine の完成度を高めることを目指す。特に本研究では、Wine と DirectX[12][13]の互換性について調べることを目的とする。DirectX を利用した Windows ソフトウェアは数多く存在するため、DirectX を利用した際の問題点を改善する事が出来れば、多くのソフトウェアを Wine で動作させる事が可能になると考えたからである。

まず初めに、DirectX の基本的な関数を用いた簡単なゲームを作成し、それを Wine で動作させることによって、どのような欠陥があるかを発見する。次に、発見した問題点の解決方法を開発者からの視点とユーザからの視点の両方から考察し、Wine で動作させる際の問題を避けるプログラミング手法や、なるべく多くのプログラムが動作する Wine の設定を提供する。更に、それらを文章にまとめて Web 上で公開する。なお本研究では、Wine の動作に環境として Linux(Fedora9[14][15])を使用する。

1-2. 本論文の構成

1. はじめに 本研究の背景と目的を示す。
2. Wine とは 本研究で扱う Wine の動作原理について示す。
3. DirectX 本研究で取り扱う DirectX の動作原理について示す。
4. ソフトウェア仕様 Wine と DirectX の互換性を調べる際に作成したソフトウェアについて仕様を示す。
5. 開発環境 本研究で利用した PC のスペックを示す
6. Wine での動作実験 本研究で作成したソフトウェアを実際に Wine で動作させた時の挙動を示す。
7. 問題点 6. において見つかった問題点を示す。
8. 考察 7. において見つかった問題点の解決方法を示す。
9. まとめ 本論文のまとめである。

Interchangeability of Wine and DirectX

Wine is the application that be able to run Windows applications on POSIX compatible operating systems. But Wine is still under development, and it is not yet suitable for general usage. Then the purpose of this study is to find out and fix the problems of Wine. We researched the problems of Wine by running the original software that was made of DirectX. As a result, we found some DirectX instructions were ignored when using Wine's proprietary DLLs.

Key words : Wine, DirectX, DLL

2章. Wine とは

Wine は仮想マシンとしてではなく、互換レイヤーとして動作する win32 エミュレータである (図 1)。Windows プログラムが要求する DLL の代替品を供給し、また Windows カーネルのプロセスを再現することによって Windows 以外の OS でも Windows プログラムを動作させることが出来る。したがって、Wine で Windows プログラムを動作させる上で Windows は不要である。また Wine が供給する DLL の他に Windows 純正の DLL も使用することが出来る (図 2)。

Wine は前述の通り独自の DLL を Windows が要求する DLL の代わりに供給するのだが、独自の DLL には問題点もあり、動かない場合も多々ある。そこで、本研究では独自の DLL と純正の DLL、両方を使う事になっている。



図 1 Wine の動作イメージ



図 2 Wine の動作イメージ

3章. DirectX

DirectXはマイクロソフト社が開発した、マルチメディア処理用のAPIの集合体であり、主にWindowsで広く使われている。DirectXはLinux上ではOpenGL[16]のコードに変換することで動作する。本研究ではDirectXを用いて開発した自作ソフトウェアをWineで動作させ、特に以下の項目について互換性に問題が無いかを調べている。

(1) Direct 3D

Direct 3Dは3Dや2D画像をモニタ上に出力するコンポーネントである。DirectX7まではDirectDrawという2D描画専用のコンポーネントも存在していたが、8以降ではDirect3Dに統合され3D処理を用いて2D描画を行っているのでそちらは考えないこととする。

(2) DirectMusic

DirectMusicはサウンドに関する処理を行うコンポーネントである。サウンドの処理にはDirectSoundというコンポーネントも存在するが、こちらはWAVE形式しか扱うことが出来ない。今回はWAVE形式に加えMIDI形式も使用するので両方の再生が可能なDirectMusicを選択した。

(3) DirectInput

DirectInputはキーボード、マウスなどからの入力処理を行うコンポーネントである。DirectInputは9からXInputというライブラリが加わった。しかし、これは主にMicrosoftのXbox360向けのランタイムで、制約も多いためDirectInputを使用した。

4章. ソフトウェア仕様

本研究では DirectX を用いた 3D シューティングゲームを作成して wine の互換性を調査した [17][18][19]。ここではその仕様について説明する。

4-1. 操作方法

移動にはテンキーを使用する(図3)。

8 : 前スライド移動

5 : 後スライド移動

4 : 左スライド移動

6 : 右スライド移動

7 : 左旋回

9 : 右旋回

スペース : ショット

7	8	9	
4	5	6	

図3 操作キー

4-2. 画面説明

動作画面を図4に示す。

- 1 : 自機の HP。初期値は 100 で敵の攻撃を受けると攻撃に応じて減っていく。0 になるとゲームオーバー。
- 2 : 敵機の HP。初期値は 100 で時期の攻撃を受けると減っていく。0 になるとゲームクリアー。
- 3 : 自弾。スペースを押すと発射され、発射時に自機の向いていた方向に等速直線運動を行う。
- 4 : 自機。上記の操作方法に従って移動する。敵弾と重なるとダメージ扱い

になり、爆風エフェクトが表示される。

立方体で構成されており各面に対応する画像として「前」「後」「上」「下」「右」「左」と描画されている。

- 5 : HIT表示。自弾が敵機と重なると” HIT” と一定時間表示される。
- 6 : ダメージ表示。敵弾が自機と重なると” DAMAGE” と一定時間表示される。
- 7 : 敵弾。一定感覚で発射されていて、発射時に敵機の向いていた方向に等速直線運動を行う。なお、敵弾は敵機のダメージに応じて変化し、ダメージ量もそれに伴い変化する。
- 8 : FPS表示。一秒間に何フレーム描画されているかを表示する。
- 9 : 敵機。一定のパターンにしたがって移動している。立方体で構成されており、全面に「敵」と描画されている。

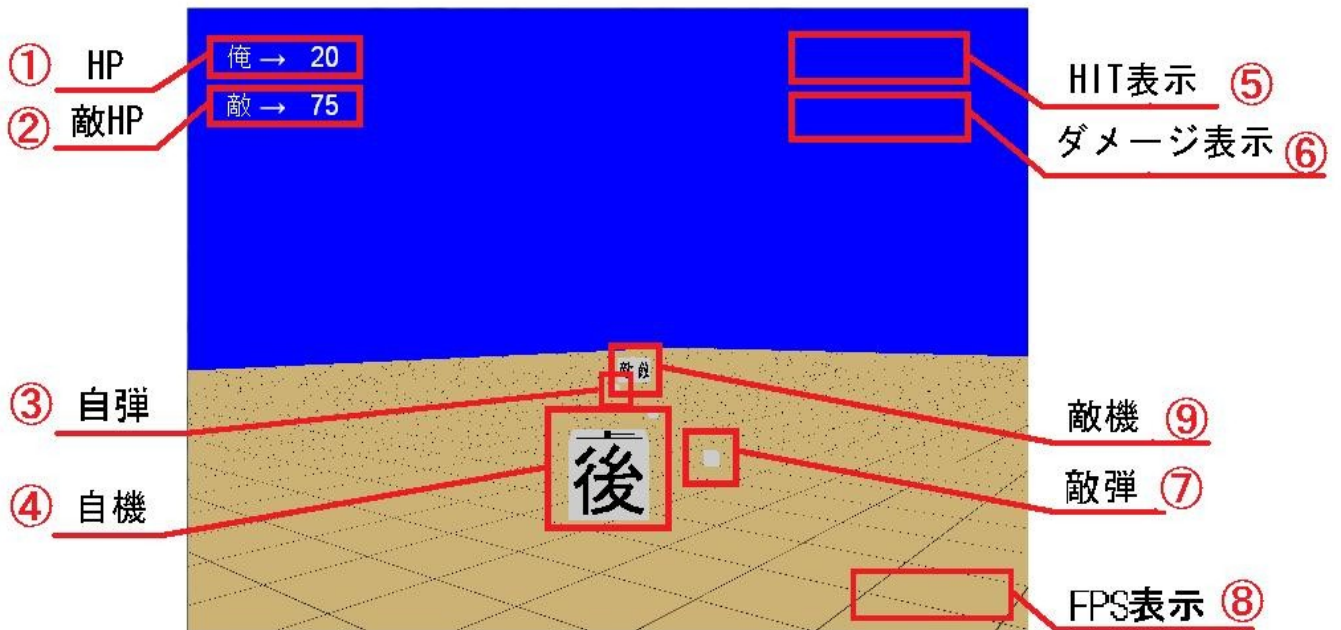


図4 ゲーム動作画面

4-3. ゲームフロー

本ゲームのフローは図5の様になっている。括弧で括られている名前は内部のゲーム状態である。状態の遷移はキー入力や、自機HP、敵機HPをフラグにして行われる。キー入力による終了処理は今回用意していないが、終了メッセージで終わるようになっているので、終了するときはAlt+F4、右上の×ボタン等で終了メッセージを送る。

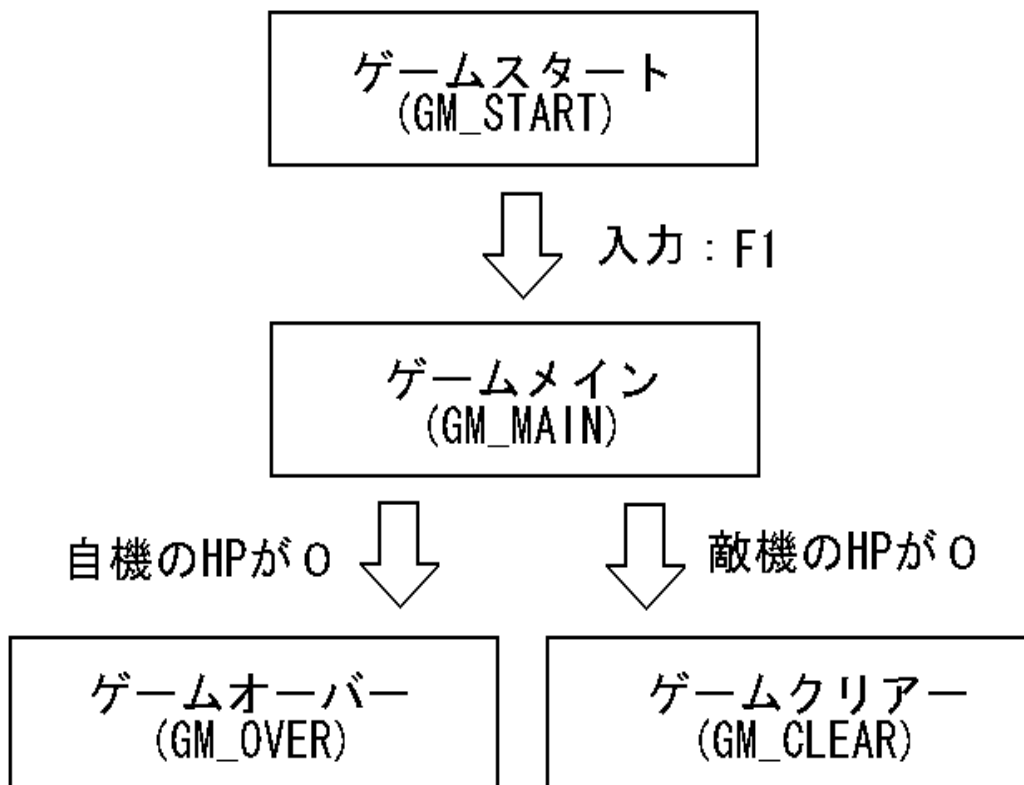


図5 ゲームフロー

5 章. 開発環境

ゲームを開発した環境は以下の通りである。

Windows Vista Home Premium
Microsoft Visual C++ 2008 Express Edition
DirectX (Microsoft DirectX SDK (February 2007))
CPU Intel(R)Core(TM)2 Duo E6550
メモリ 2020MB
GPU Intel(R) G33/G31 Express Chipset Family

Wine を用いて動作確認をした環境は以下の通りである。

fedora 9 (Sulphur)
Linux 2.6.25-14.fc9.i685
CPU Intel(R) Pentium(R)4 CPU 2.40Ghz
メモリ 374.5MB
GPU Intel(R) 845G Chipset
Wine 1.1.6

6 章. Wine での動作実験

Wine での動作について解説する。まず Wine オリジナルの DLL を用いてソフトウェアの起動を行った場合、サウンドのロード (IDirectMusicLoader) においてエラーメッセージが出る。そして Wine は d3dx9_36.dll を要求してくる。表示されるメッセージは以下の通りだった。

```
err:dmloader:IDirectMusicLoaderImpl_IDirectMusicLoader_SetObject :
could not attach stream to file

wine: Call from 0x77591e50 to unimplemented
function d3dx9_36.dll.D3DXLoadMeshFromXW, aborting
wine: Unimplemented function d3dx9_36.dll.D3DXLoadMeshFromXW called at address
0x77591e50 (thread 0009), starting debugger...
```

次に d3dx_36.dll を純正の物と差し替えた場合、サウンドが再生されない。表示されるエラーメッセージは以下のようになる。

```
fixme:dmime:IDirectMusicPerformance8Impl_PlaySegmentEx (0x14d0e8, 0xdbad6c,
(nil), (nil), 128, 0x0, (nil), (nil), (nil)): stub
fixme:dmime:IDirectMusicPerformance8Impl_Stop (0x14d0e8, (nil), (nil), 0, 0):
stub
fixme:dmime:IDirectMusicPerformance8Impl_CloseDown (0x14d0e8): stub
```

サウンドに関しては表示されるエラーメッセージの関数に対応する dll を差し替えても再生されることは無かった。

図 6 が Windows でのタイトル画面、図 7 が Wine を使って動作させた時のタイトル画面である。図 8 は Windows でのメインゲーム画面。図 9 は Wine でのメインゲーム画面で、どちらも動作に関しては問題ないように見える。



図6 Windowでのタイトル画面



図7 Wineでのタイトル画面

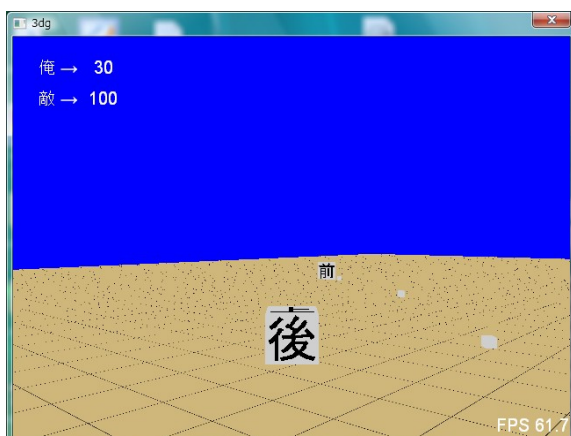


図8 Windowsでのゲーム画面

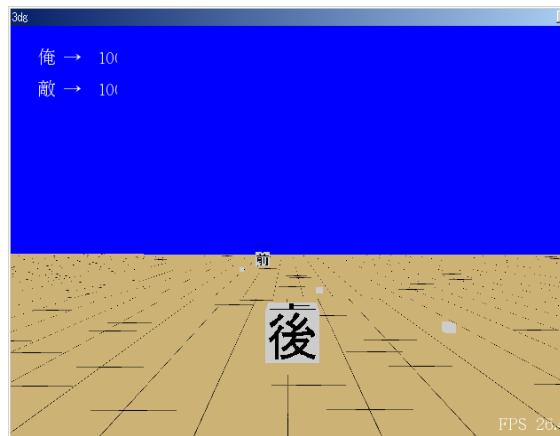


図9 Wineでのゲーム画面

7 章. 問題点

実験の結果、以下のような問題があることが分かった。

(1) マルチスレッド DLL コンパイルすると DLL の位置に関わらず起動できなかった。エラーメッセージは以下の通りである。

```
err:module:import_dll Library MSVCR90.dll (which is needed by L"3dg.exe") not found
err:module:LdrInitializeThunk Main exe initialization for L"3dg.exe" failed, status c0000135
```

(2) X ファイルの読み込みが出来なかった。

(3) サウンドの再生が出来なかった。

(4) フォントの表示が崩れた。Windows での表示は図 10、Wine では図 11 となるが、日本語フォントの表示が崩れて数字の右側が表示されなくなっていた。



図 10 Windows での表示

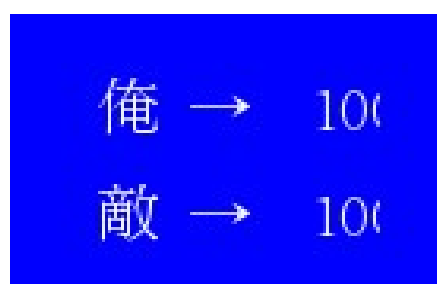


図 11 Wine での表示

8 章. 考察

7 章での 4 つの問題は以下の手法で解決出来た。

(1) マルチスレッド DLL コンパイルすると DLL の位置に関わらず起動できない問題に関してはマルチスレッドでコンパイルすると問題は起きない。今回用いた Visual C++2008EE ではソリューションエクスプローラからプロジェクト名を右クリックし、プロパティを開き、構成プロパティ→C/C++→コード生成タブでマルチスレッドを選択すると良い。

(2) X ファイルの読み込みが不可能である問題に関しては、ユーザー側は `d3dx_36.dll` を Microsoft 社が提供している DirectX9 ランタイムのものに差し替えることで解決できる。また、開発者側は `D3DXLoadMeshFromX` 関数を使わないで、代替関数として `D3DXLoadMeshHierarchyFromX` を使っても良いが、X ファイル形式は DirectX10 では推奨されない形式であるので[20]、独自形式で自力でローダを組んだほうが良いだろう。

(3) サウンドの再生が不可能な問題に関しては解決できなかった。したがって `Direct Music` を利用したサウンドの再生は行わず、`Direct Sound` を利用することが考えられる。他にも `WindowsAPI` を利用した音楽再生なら問題が起きない可能性がある。

(4) フォントの表示が崩れる問題に関しては `Wine Configuration` でフォントを設定する。このツールを開き、デスクトップ統合タブをクリックする。次に項目のメニューをクリックし、変更したいフォントを選び右側のフォントボタンを押して変更すれば表示が崩れなくなる。

9章. まとめ

本研究の「Wine と DirectX の互換性について調べる」という目的は、動作実験によって Wine の問題点をいくつか発見し、修正する方法を複数提示できたことから達成できたといえる。

発見した 4 つの問題点は、開発者側からもユーザ側からも修正できるようなものであったが、出来るだけ開発者側が意識してソースコードを書くことで Wine ユーザが苦勞せずに利用することが出来るソフトウェアを作ることが可能ということが分かった。しかし、開発者すべてが Wine の存在を知っていて、互換性があるプログラムを組むとは限らないため、Wine ユーザは DirectX を使用する開発者に Wine を意識したプログラミング手法を心がける必要性を訴えるべきであると思われる。Wine は現在も開発が進んでいるプロジェクトであるため、今後の進行によっては現在動かないようなソフトウェアも動作するようになっていくだろう。

なお、「サウンドが再生されない」という問題点については 8 章で非常に使い勝手の良い高レベル API である Direct Music を使わずに、低レベル API である Direct Sound や Windows32API を使うことでの解決法を示したが、まだ解決の確認を取れていない。この問題については今後確認を取る予定である。

[参考文献]

[1] MARKET SHARE

<http://marketshare.hitslink.com/>

[2] Microsoft Windows ホームページ

<http://www.microsoft.com/japan/Windows/default.aspx>

[3] VMware

<http://www.vmware.com/jp/>

[4] Citrix Systems Citrix XenServer

<http://www.xensource.com/>

[5] QEMU on Windows

<http://www.h7.dion.ne.jp/~qemu-win/index-ja.html>

[6] 中島 久弥,日系 Linux4月号,2008年4月8日,日系BP社

[7] Wine hq

<http://www.winehq.org/>

[8] 中島 久弥,日系 Linux10月号,2008年10月8日,日系BP社

[9] WINE License

<http://www.winehq.org/site/license>

[10] GNU Operating system

<http://www.gnu.org/copyleft/lesser.html>

[11] OSG-JP GNU LGPLv3 日本語訳

http://sourceforge.jp/projects/opensource/wiki/licenses%2FGNU_Lesser_General_Public_License_version_3.0

[12] DirectX ホームページ

<http://www.microsoft.com/japan/windows/DirectX/default.aspx>

[13] DirectX ホーム

<http://msdn.microsoft.com/ja-jp/directx/default.aspx>

[14] fedora

<http://fedoraproject.org/ja/>

[15]中島 久弥,日系 Linux7 月号,2008 年 7 月 8 日,日系 BP 社

[16] OpenGL

<http://www.opengl.org/>

[17] 大槻 雄一郎,15 歳からはじめる DirectX9 3D ゲームプログラミング C++編,
2007 年,ラトルズ

[18] 星 正明,DirectX9 実践プログラミング,2003 年、工学社

[19] NRTTKER,DirectX9 DirectX Graphics,2005 年,工学者

[20] DirectX ラーニング

<http://msdn.microsoft.com/ja-jp/directx/bb896684.aspx>