

Giochi Windows su GNU/Linux
Wine, DXVK e Lutris

di
Matteo Bini

© 2023 Matteo Bini

Il testo integrale, le diapositive e il codice ROFF di questa presentazione
sono rilasciati con la licenza CC BY-SA 4.0 di Creative Commons.

<https://creativecommons.org/licenses/by-sa/4.0/legalcode.it>

Introduzione

Il software proprietario è un'ingiustizia.

I videogiochi liberi come Minetest sono una manna dal cielo.

Quali sono le implicazioni, ma soprattutto le possibilità, di giocare a un gioco per Windows su GNU/Linux?

Prima un paio di considerazioni...

I giochi liberi sono meglio

Esaminare com'è fatto un videogioco è fantastico per imparare, o semplicemente per il gusto di farlo e poi condividere la versione modificata con un amico, per divertirsi assieme.

Tutto ciò non è possibile con un gioco proprietario.

C'è chi modifica giochi proprietari, *lemod*.

C'è un limite a quello che si può modificare di un programma, senza possederne il codice sorgente.

Le mod non hanno quasi nessuna tutela legale. Spesso gli autori delle mod sono perseguiti dagli autori del videogioco modificato.

Condivisione

Anche se considerassimo i videogiochi arte, e perciò sacrilego alterare la visione originale dell'autore, modificare un gioco e ridistribuirlo non impedisce di fruire della prima versione.

Non stiamo dipingendo sulla stessa del pittore, ma su un'altra.

Non è plagio, ma arricchimento.

Eseguibile o motore libero

Se non riuscite a digerire una società con meno enfasi sull'autore e con maggiore importanza all'arte per tutti, pubblicate solo il codice del motore del gioco e non i dati.

Liberate la parte logica che fa funzionare il gioco, in modo che tutti possano apprendere, però non potranno modificare liberamente il resto degli elementi, come la grafica, la musica, la storia, ecc.

È meglio un gioco libero, però un motore libero è un buon inizio.

Meglio un gioco libero, anche se per Windows

Un gioco proprietario su GNU/Linux non è meglio di uno proprietario per Windows, solo perché gira su un sistema libero.

Un purista asserirebbe che è peggio, poiché inquina il sistema libero.

Un gioco proprietario per GNU/Linux è peggio di un gioco libero su Windows, in quanto il gioco libero per Windows invita alla libertà, mettendo in discussione la logica di Microsoft e proponendo un modello equo.

Invece il gioco proprietario su GNU/Linux diminuisce la libertà di chi usa quel sistema, che di base sarebbe libero.

Steam non è libero

Valve investe su Linux per creare un suo sistema, per competere con Nintendo o Sony, non perché voglia allontanarsi dal modello proprietario.

Nel 2003 Valve aprì Steam, un negozio digitale dove comprare videogiochi con DRM.

Nel 2013 Valve scelse GNU/Linux come sistema operativo dei suoi computer e della sua console portatile.

Ciò ha aumentato il numero di titoli proprietari per questo sistema.

Se vi interessa solo giocare ai videogiochi col computer, questa è una buona notizia.

Se vi interessasse solo giocare ai videogiochi, sul computer non installereste GNU/Linux.

Il DRM è peggio dei giochi proprietari

I giochi proprietari sono brutti e un'ingiustizia, però ci volete giocare lo stesso.

La vita è una questione di equilibri e una partita ogni tanto non può erodere i vostri valori, se ci credete davvero.

Qual è il modo di fare questa partita più nella direzione del software libero?

Non comprate da Steam, meglio scaricare.

GOG.com vende giochi senza DRM. Lo stesso itch.io.

Trovate un negozio o uno sviluppatore che venda giochi senza sistemi di restrizione. Niente DRM!

Macchina virtuale o installazione a parte?

Per giocare a un gioco per Windows su GNU/Linux, dovete installare una macchina virtuale? Dovete tenere un computer o un disco apposta con Windows?

No!

Non regalate soldi e una fetta di mercato a Microsoft.

Vi presento Wine.

Wine

Wine non è un emulatore (Wine Is Not an Emulator).

È un programma il cui sviluppo è iniziato nel 1993 per far funzionare i programmi di Windows 3.1 su Linux.

Da allora si è evoluto ed è in grado di far girare i programmi di ogni versione di Windows.

Come funziona Wine

Wine esegue il programma desiderato e converte in tempo reale le chiamate alle API di Windows in chiamate POSIX.

Wine può girare su tutti i sistemi POSIX come GNU/Linux, Mac e i BSD.

Wine è come un amico che vi ascolta da ubriachi.

Programma: «Per favore, aprimi una finestra e disegnaci dentro il menù in alto e un'immagine di uno gnu».

Wine, anziché assecondare le richieste da ubriaco e aprire una finestra Windows, compie l'azione per GNU/Linux, disegnando una finestra di X11 con dentro uno gnu.

Pura magia! E tanta fatica di gente che si è sbattuta per riscrivere le chiamate di Windows in equivalenti.

Installare Wine

Su Debian basta `apt install wine`.

Se avete esperienza con GNU/Linux, seguite la guida sul sito di Wine per avere la versione più recente.

<https://wiki.winehq.org/Download>

Usare Wine

Basta cliccare su un programma per Windows.

Altrimenti dal terminale digitare `wine` e il percorso del programma per Windows.

Wine fa funzionare tanti programmi, ma non tutti.

Lo sviluppo di Wine è continuo per tenere conto dei cambiamenti di Windows.

Gli sviluppatori di Wine hanno meno risorse di Microsoft. Ciononostante alcuni giochi funzionano meglio con Wine, soprattutto quelli vecchi.

Sul sito di Wine trovate una lista delle compatibilità.

<<https://appdb.winehq.org/>>

Wine solo per i giochi?

Wine può essere usato per tutti i programmi di Windows.

Importanza di Wine

Se foste un'azienda o un professionista interessato a cambiare sistema operativo, a favore di uno che rispetti la vostra libertà, potreste sfruttare Wine per migrare i dati dai vecchi programmi per Windows a quelli nuovi per GNU/Linux.

Potreste usare dei programmi per Windows, senza dover installarlo e senza comprare una licenza da Microsoft.

Wine è un programma importante che dimostra la flessibilità del software libero, sebbene presenti degli aspetti controversi.

Pro e contro di Wine

- Wine serve per utilizzare software proprietario.
- La sua esistenza deriva da motivi pratici ed economici, piuttosto che ideologici.
- + Wine non aumenta la libertà informatica di chi lo usa, a meno che non si consideri il vantaggio di non dipendere da Windows.

Windows spia i suoi utenti e adotta varie pratiche discutibili.

<https://www.gnu.org/proprietary/malware-microsoft.it.html>

- Wine non contribuisce direttamente alla costruzione di un ecosistema informatico libero.
- + Tuttavia permette di studiare programmi esistenti su un altro sistema operativo, affinché vengano riscritti liberi e per sistemi liberi.

Pro e contro di Wine

- + Wine permette flessibilità, soprattutto quando non esiste ancora un programma di software libero per svolgere un determinato compito, o non viene ritenuto all'altezza.
- D'altro canto l'enfasi sulla funzionalità rischia di distogliere l'attenzione sulla libertà, spostandola sulla qualità.

Un programma proprietario può essere di buona qualità. Accontentarsi di farlo girare con Wine non risolve l'ingiustizia di non avere accesso al codice sorgente.

Il miglior uso di Wine è come toppa, o come compromesso.

- + Wine preserva i giochi "vecchi" che non girano su Windows.
- + I giochi vecchi costano meno.

Pro e contro di Wine

- + I giochi vecchi consumano meno (aiutano l'ambiente?).
- Oggi troppe energie vengono sprecate per titoli mediocri ed eccessivamente costosi.
- + Potremmo riscoprire un sacco di gemme dimenticate, risparmiare soldi ed essere felici.

DXVK

Per la grafica, spesso i giochi di Windows si appoggiano alle API Direct3D di Microsoft.

Quando Wine riceve chiamate alle Direct3D, le converte in OpenGL.

OpenGL è un'API multipiattaforma per accedere alla potenza di calcolo delle schede video, indipendentemente dal sistema operativo per cui state scrivendo il programma.

Dal 2016 esiste l'API Vulkan che ha lo stesso obiettivo di OpenGL, è sviluppata dagli autori di OpenGL e, per merito di alcune differenze strutturali, riesce a ottenere prestazioni migliori in molti casi.

La differenza lampante con OpenGL è che, nella stessa scena, Vulkan diminuisce il carico sul processore spostandolo sulla scheda grafica.

Vulkan sfrutta la scheda grafica di più di OpenGL, liberando il processore per

DXVK

altri compiti.

Quindi i giochi girano meglio.

Wine ancora non sfrutta Vulkan. Ci stanno lavorando.

DXVK è un progetto scollegato da Wine che permette di trasformare le chiamate alle API Direct3D 9, 10 e 11 in chiamate Vulkan.

Con DXVK, potrete ottenere delle prestazioni migliori sui giochi che si appoggiano alle Direct3D 9, 10 e 11, nel mentre il progetto Wine mette a punto l'interfaccia con Vulkan, come alternativa a quella attuale con OpenGL.

DXVK è una toppa per una parte in lavorazione di Wine, che è a sua volta una toppa.

Installare DXVK

Installare DXVK è più complicato d'installare Wine, però non è obbligatorio per usare Wine.

È un in più per far andare meglio i programmi che si appoggiano alle Direct3D 9, 10 e 11.

Scaricate DXVK da GitHub <<https://github.com/doitsujin/dxvk/releases>>

o compilate il codice sorgente

<<https://github.com/doitsujin/dxvk#build-instructions>>.

Più persone dovrebbero compilare il codice sorgente: la rivoluzione del software libero passa attraverso la compilazione indipendente del codice sorgente!

Installare DXVK su Wine a 32 bit

Se la cartella di Wine, o in gergo prefisso, è a 32 bit,

copiate tutti i file DLL nella cartella x32 di DXVK

nella cartella di Wine, sotto `drive_c/windows/system32`.

Di solito il prefisso si trova in `$HOME/.wine`.

```
$ export WINEPREFIX=$HOME/.wine
$ cp x32/*.dll $WINEPREFIX/drive_c/windows/system32
```

Installare DXVK su Wine a 64 bit

Copiate tutti i file DLL nella cartella x64 di DXVK

nella cartella di Wine, sotto `drive_c/windows/system32` .

Dopodiché copiate tutti i file DLL nella cartella x32 di DXVK

nella cartella di Wine sotto `drive_c/windows/syswow64` .

```
$ export WINEPREFIX=$HOME/.wine
$ cp x64/*.dll $WINEPREFIX/drive_c/windows/system32
$ cp x32/*.dll $WINEPREFIX/drive_c/windows/syswow64
```

Wine a 32 o a 64 bit?

Per sapere se il prefisso di Wine è a 32 o a 64 bit:

- basta controllare se, all'interno del prefisso, esiste la cartella `drive_c/windows/syswow64`.

Se c'è, quello è un prefisso a 64 bit.

Altrimenti è a 32 bit.

Scavalcare le DLL di DXVK

Dobbiamo dire a Wine che, quando cerca le librerie contenute nei file DLL che abbiamo copiato, non deve eseguire le sue funzioni di traduzione, ma quelle contenute nelle DLL di DXVK.

Eseguiamo Wine con la variabile di ambiente `WINEDLLLOVERRIDES` valorizzata nel seguente modo: `d3d9 , d3d11 , d3d10core , dxgi=n .`

```
$ WINEDLLLOVERRIDES="d3d9 , d3d11 , d3d10core , dxgi=n" wine programma
```

La `n` di *native* (nativo) indica che quelle librerie non devono essere tradotte da Wine, ma caricate come DLL.

La `b` di *builtin* (integrato) indicherebbe il contrario.

Potremmo specificare una precedenza, separando i caratteri con una virgola.

Scavalcare le DLL di DXVK

n , b indicherebbe che Wine deve prima provare a caricare la libreria dai file DLL all'interno del prefisso e, solo se fallisce, può provare a tradurre quella libreria con le sue funzioni interne.

```
$ WINEDLLOVERRIDES="d3d9 ,d3d11 ,d3d10core ,dxgi=n ,b" wine programma
```

winecfg

Il comando di prima non salva nulla nella configurazione di Wine, esegue e basta un programma con Wine, sfruttando le DLL di DXVK, contrariamente a quello che sarebbe il comportamento di base.

Ogni volta dev'essere digitato.

Alternative:

- alias o comando che valorizzi prima la variabile di ambiente `WINEDLLOVERRIDES` e poi chiami Wine,
- esportare la variabile di ambiente nella linea di comando da cui lanciamo Wine;
- più semplicemente usare winecfg (Wine Configuration Editor) per salvare le impostazioni delle DLL.

winecfg

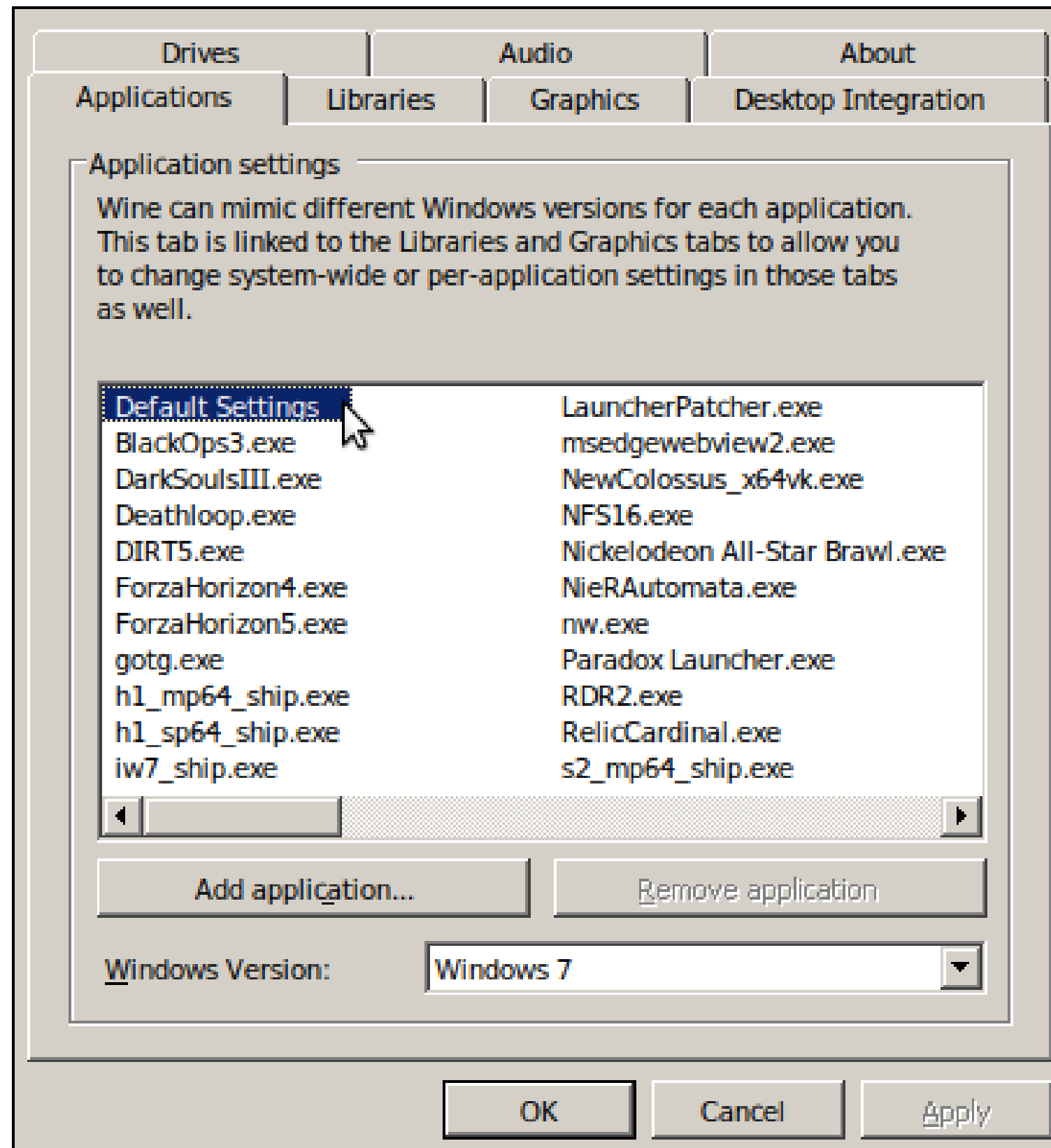
winecfg può salvare le impostazioni in generale o per un eseguibile specifico.

Potreste volere le DLL di DXVK solo per alcuni giochi.

Dal pannello `Applications` (applicazioni) scegliete se cambiare le impostazioni per tutti programmi, selezionando `Default Settings` (impostazioni di base) o per un programma in particolare.

Potete aggiungere un programma cliccando su `Add application...` (aggiungi applicazione).

winecfg



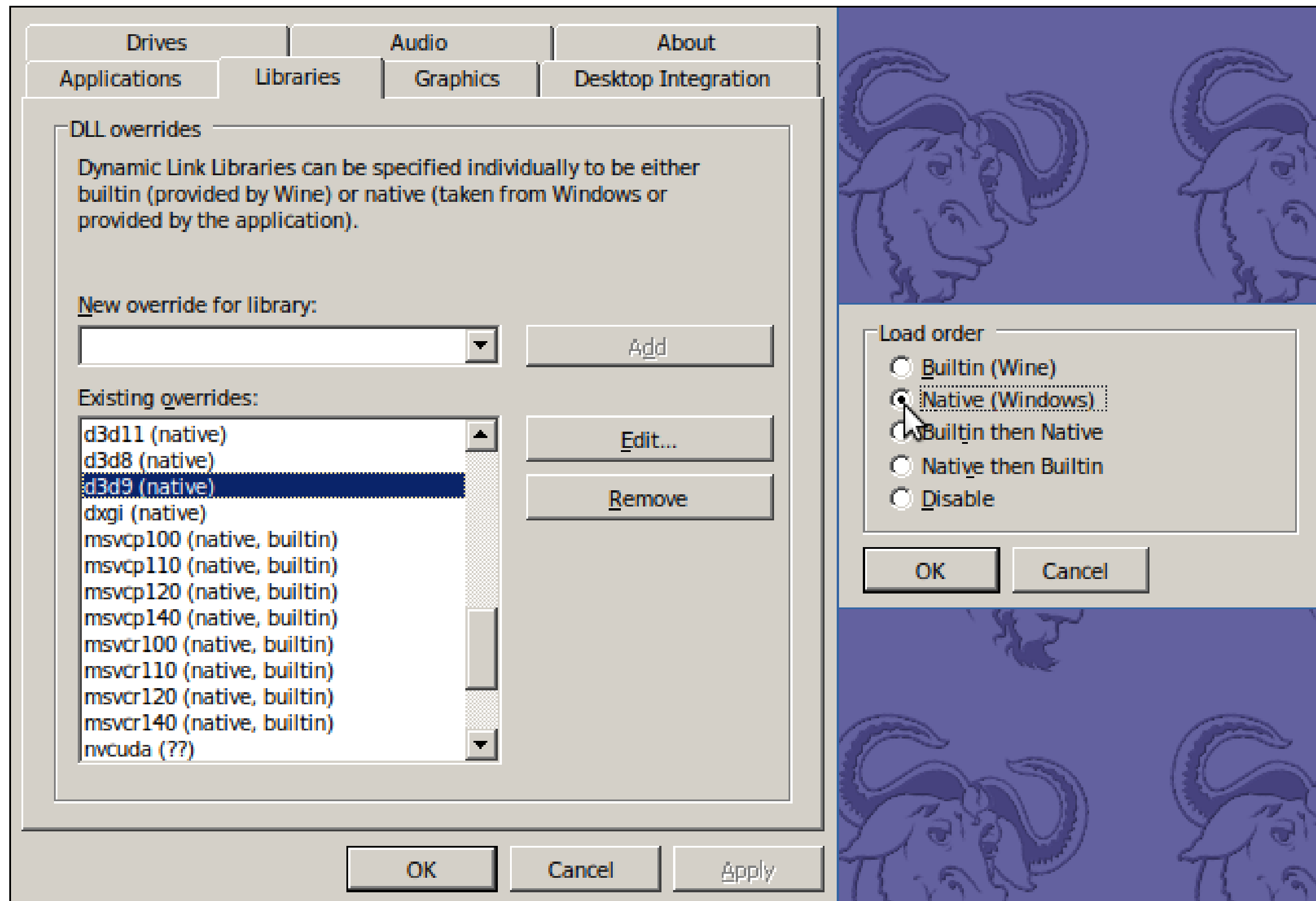
winecfg

Dopodiché spostatevi sul pannello `Libraries` (librerie), scrivete o cercate la libreria da scavalcare, rispettivamente nel campo `New override for library:` (nuova eccezione di libreria) o nella lista sotto `Existing overrides:` (eccezioni esistenti).

Se la DLL non è presente nella lista, scrivete il nome della DLL e premete `Add` (aggiungi), poi selezionatela dalla lista e col tasto `Edit` (modifica) scegliete la preferenza, nel nostro caso `native`.

Premete `Ok` per salvare.

winecfg



D8VK

Se il gioco è d'inizio 2000, probabilmente utilizza Direct3D 8.

In quel caso c'è D8VK, un progetto derivato da DXVK che fa la stessa cosa di DXVK, ma per Direct3D 8. <<https://github.com/AlpyneDreams/d8vk>>

Sia l'installazione che la configurazione sono identiche a DXVK, con l'unica differenza del nome della DLL, che in questo caso è d3d8.

Ancora più indietro

Se il gioco risale a prima del 2000, le cose si semplificano o si complicano.

Per i giochi che usano le versioni precedenti a DirectX 8, Wine da solo va più che bene: sono giochi semplici e il vantaggio di usare Vulkan al posto di OpenGL è minimo, soprattutto vista la potenza dei computer odierni.

Per quanto riguarda i giochi che sfruttano Glide, al momento non esistono librerie di conversione libere e attualmente sviluppate.

C'è nGlide <<http://www.zeus-software.com/downloads/nglide>>

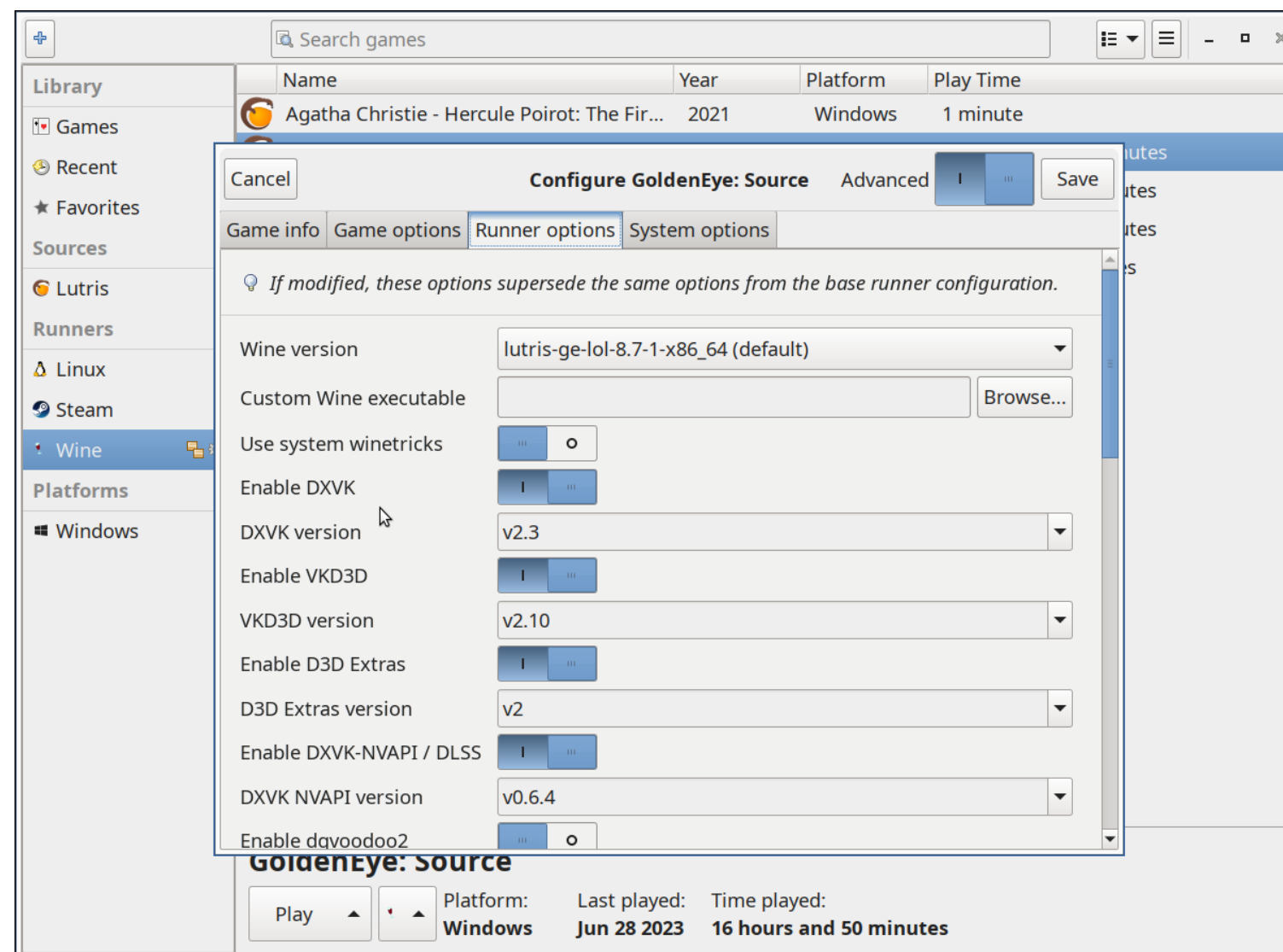
e dgVoodoo2 <<http://dege.freeweb.hu/dgVoodoo2/>>.

Però non sono software libero.

Lutris

Lutris è un'interfaccia grafica scritta in Python e GTK per gestire Wine, DXVK ed eventualmente dgVoodoo2, oltre che per semplificare il giocare su GNU/Linux, emulatori e interpreti inclusi.

Sul sito di Lutris trovate come installare e configurare tantissimi giochi, automaticamente. <<https://lutris.net/>>



Lutris

Nonostante Lutris sia software libero, permette d'installare software proprietario, oltre ai giochi, come dgVoodoo2.

Gli sviluppatori di Lutris sono più interessati a far funzionare i giochi, piuttosto che alla libertà del software.

Il progetto Lutris usa Discord.

È la sensazione a pelle di un integralista GNUslamico, che non conosce nessuno degli sviluppatori di persona.

Curiosità

Secondo il *Wine Application Database* (AppDB), la versione Mac di The Sims 3 è basata su Wine ed è uno dei motivi per cui The Sims 3 funziona bene con Wine.

DXVK è nato dall'esigenza del suo sviluppatore, Philip Rebohle, di giocare a *Nier: Automata* su GNU/Linux.

<<https://github.com/doitsujin/dxvk/releases/tag/v0.20>>

L'autore di D8VK, Jeffrey Ellison, ha invitato gli sviluppatori di DXVK a includere il suo codice nel progetto, affinché DXVK traduca anche Direct3D 8 in Vulkan. La proposta è stata accettata, ma l'unione ancora non è terminata.

<<https://github.com/doitsujin/dxvk/pull/3411>>

Nel prossimo futuro non sarà più necessario installare D8VK a parte: basterà avere DXVK per tradurre le chiamate alle Direct3D 8, 9, 10 e 11 in Vulkan.

Curiosità

Il codice dell'API Glide fu pubblicato nel 2003.

<<https://sourceforge.net/projects/glide/>>

Non c'è nessun ostacolo tecnico che impedisca a un programmatore di buona volontà di scrivere una libreria libera di conversione da Glide a Vulkan.

Scala delle libertà

1. Gioco libero
2. Motore libero
3. Wine
4. Gioco proprietario
5. Gioco proprietario con DRM (Steam)

Conclusioni

Wine è un gran bel pezzo di programma!

È interessante e istruttivo capire come funziona.

DXVK e D8VK sono dei progetti utili, che in futuro non serviranno, quando gli sviluppatori di Wine finiranno d'implementare l'interfaccia con Vulkan.

Ad ora sono un in più, per i curiosi e gli smanettoni, desiderosi di migliorare le prestazioni, già più che buone.

Wine traduce in tempo reale le API di Windows in chiamate POSIX.

Provate rispetto per un'impresa del genere!

Linux gamers don't wine (i giocatori su Linux non si lamentano e non usano Wine).

Diapositive



<https://choice.tiepi.it/~matteobin/scritti/giochi-winzoz-su-gnulinix.html#diapositive>