

*Software libero e minimalismo
per uno sviluppo davvero indipendente*

di
Matteo Bini

Il testo integrale e le diapositive di questa presentazione sono rilasciate con licenza CC0 di Creative Commons. <<https://creativecommons.org/publicdomain/zero/1.0/legalcode.it>>

Nei limiti permessi dalla legge, ho rinunciato a tutti i diritti d'autore e derivati o connessi su quest'opera. Quest'opera fu pubblicata in Italia nel 2023.

Introduzione

Vogliamo creare un videogioco sovversivo, slegato dal profitto, sperando che sia veicolo di cambiamento sociale. Come?

Dobbiamo sempre domandarci cosa comportino i mezzi che impieghiamo, anche se pensassimo che il fine li giustifichi.

Nell'informatica c'è bisogno di libertà. Non possiamo affidarci alle ditte che controllano il mezzo digitale, poiché esse sono asservite al capitalismo e distruggerebbero un'ideologia diversa.

Abbiamo bisogno di una struttura che protegga e faccia crescere una cultura sovversiva.

Introduzione

Una cultura equa pensata per durare dev'essere decentralizzata:
per mitigare la perdita delle informazioni,
per aumentarne la diffusione,
per diminuire il potere di chi l'ha creata.

Profitto è sinonimo di potere. Perciò per creare dei videogiochi scollegati dal profitto, dobbiamo scollegarli dal potere. Il modo più efficace è tramite la condivisione della conoscenza.

Se condividiamo la conoscenza, chiunque potrà imparare a creare videogiochi gratis, aggirando le barriere del potere economico e sociale come le scuole. Così restituiremo il mezzo videoludico al popolo.

A tal fine, propongo di adottare la filosofia del software libero di Richard Stallman, fondatore della Free Software Foundation e del progetto GNU.

Introduzione

Le quattro libertà fondamentali del software libero:

0. Libertà di eseguire il programma come si desidera, per qualsiasi scopo.
1. Libertà di studiare come funziona il programma e di modificarlo in modo da adattarlo alle proprie necessità. L'accesso al codice sorgente ne è un prerequisito.
2. Libertà di ridistribuire copie in modo da aiutare gli altri.
3. Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti da voi apportati (e le vostre versioni modificate in genere), in modo tale che tutta la comunità ne tragga beneficio. L'accesso al codice sorgente ne è un prerequisito.

Introduzione

L'intento del software libero è di restituire l'informatica al proletariato.

Dal software libero e dal progetto GNU è nato il sistema operativo GNU/Linux con le sue distribuzioni, per esempio Ubuntu.

L'intero codice sorgente di questo sistema operativo gratuito è pubblico e modificabile da chiunque.

Per ulteriori informazioni <<https://www.fsf.org/it/what-is-fs-ita>>

Introduzione

Ora passiamo dalla teoria alla pratica: ecco una serie di gradini da scalare per avvicinarci alla meta.

La nostra meta è l'appropriazione di mezzi che siano anch'essi sovversivi, per creare un gioco il cui messaggio possa crescere grazie a un ecosistema indipendente ed equo.

Ho scelto la metafora della scala perché ogni passaggio non è più importante di quello precedente, bensì più faticoso o, nel nostro caso, richiede maggiori competenze tecniche per essere messo in atto.

I passaggi sono scollegati tra loro. Ciò vi permette di compierli in ordine sparso, a seconda del vostro livello di conoscenza o dello stadio di lavorazione del gioco.

Sapere che GNU/Linux esiste, cos'è e perché è importante

Il primo passo l'abbiamo già compiuto insieme, a meno che non sia stato un completo inetto nell'esposizione.

Distribuire il gioco attraverso negozi che rispettino gli acquirenti

Per scollegare i videogiochi dal profitto la soluzione più ovvia è abolirne la vendita.

Tuttavia, per sostenere un processo di cambiamento, è accettabile vendere videogiochi sovversivi.

Meglio se con un prezzo opzionale, come per le donazioni, per non innalzare barriere economiche.

Distribuire il gioco attraverso negozi che rispettino gli acquirenti

Criteri di valutazione per i negozi digitali:

1. Niente DRM: i file si devono poter copiare e trasferire. Nessun obbligo di connessione a Internet.
2. Possibilità di scaricare il gioco senza un programma aggiuntivo.
3. Possibilità di avere un prezzo opzionale o di mettere il gioco gratis.
4. Salvare il meno possibile delle informazioni sui giocatori. Nel caso di un gioco gratis, ciò significa non salvare alcunché!

Distribuire il gioco attraverso negozi che rispettino gli acquirenti

Con queste premesse, ecco una classifica di merito di alcuni dei maggiori siti che vendono videogiochi:

1. **itch.io**

2. **GOG.com**

Non permette il prezzo opzionale e richiede la registrazione per giochi gratuiti

3. **Steam**

DRM, necessita di un'applicazione per scaricare i giochi, non permette il prezzo opzionale e richiede la registrazione per giochi gratuiti

Distribuire il gioco attraverso negozi che rispettino gli acquirenti

Con itch.io potete pubblicare il vostro gioco quasi istantaneamente, senza pagamenti o burocrazia.

Un'opzione ottima per decentralizzare la distribuzione è la pubblicazione indipendente. Anche se può sembrare démodé, fatevi un sito: è facile e veloce.

Da non scartare pure i vettori della pirateria come i protocolli peer-to-peer BitTorrent ed eDonkey.

Cercate di diversificare per non dipendere da un unico mezzo e non contate solo su terzi.

Provare se il gioco funziona con Wine o Proton

Wine permette di far girare i programmi di Windows su GNU/Linux.

Proton è la versione di Wine modificata da Valve per Steam.

Wine non dev'essere una scusa per non scrivere del codice multiplatforma, però può essere una toppa nell'attesa di fare meglio in futuro.

Chiedete a qualcuno che usa GNU/Linux di provare il vostro gioco con Wine e di consigliarvi per aumentarne la compatibilità.

Non escludete gli appassionati del pinguino, perché loro si impegnano per un mondo informatico equo. Poverini!

Usare strumenti e librerie libere

Anche se non state sviluppando un gioco per GNU/Linux, potete attingere lo stesso alla sua filosofia: usate programmi e librerie libere.

Le licenze GPL e MIT (o Expat) aderiscono alle quattro libertà fondamentali del software. Quindi i programmi pubblicati con tali licenze sono da ritenersi software libero.

Utilizzando strumenti liberi contribuirete allo sviluppo di mezzi alternativi equi, scollegati dal profitto, e che potrebbero influenzare l'intera industria informatica, come nel caso di FFmpeg.

Usare abitualmente GNU/Linux

Questo sistema operativo libero va provato! Usate una qualunque delle distribuzioni di Linux almeno una volta a settimana. Va bene pure uno dei BSD.

Non fatevi spaventare da ciò che è diverso. Appreziate la forza della differenza radicale del paradigma informatico.

Si fa meglio con poco. Fate vostro questo motto.

Publiccare il gioco per GNU/Linux

Publiccare giochi per GNU/Linux non è difficile: anche i motori proprietari più diffusi supportano Linux, tra cui Unity e Unreal.

Sebbene un motore chiuso non sia un passo verso la sovversione del capitalismo videoludico, pubblicare il gioco per GNU/Linux vi farà imbattere nella sua logica di libertà e aumenterà l'interesse generale per questo sistema.

Attenzione a non fare il contrario: non implementate restrizioni degne delle peggiori console moderne su un sistema operativo libero.

Non scoraggiatevi se state lavorando con strumenti chiusi. Pubblicate il vostro lavoro per GNU/Linux e migliorate la prossima volta. Potete imparare un modo diverso per un mondo migliore.

Liberare il codice sorgente del gioco

Publicarne il codice sorgente con una licenza libera è tra i migliori modi per scollegare i videogiochi dal profitto.

La scelta della licenza dipende dalle circostanze. Le più famose sono la GPL e la MIT (o Expat).

Permetterete agli altri d'imparare dal vostro lavoro, studiando un prodotto vero e finito.

Pure il gioco ne trarrà beneficio, poiché potrà funzionare per sempre.

Donerete la libertà di effettuare delle modifiche per creare qualcosa di nuovo e di ancora più strepitoso!

Liberare il codice sorgente del gioco

Per innescare il potenziale sovversivo dei videogiochi, dobbiamo imparare l'azione più sovversiva nel capitalismo: la condivisione.

Condividendo il codice indebolirete il diritto d'autore, la proprietà privata digitale e il potere del privilegio.

Se temete d'intaccare le vendite, attendete qualche anno prima di rilasciare il codice sorgente.

Esistono tanti progetti che campano di donazioni.

Potreste rilasciare soltanto la logica del gioco e non i dati, come fu fatto per DOOM.

Siate generosi: liberate il codice, i dati, la grafica, la musica, tutto!

Usare motori grafici liberi

Se usate un motore grafico chiuso, per esempio Unity, servirà a poco rilasciare il codice sorgente del gioco.

Altri potranno apprendere dal vostro lavoro e modificarlo, però verrà meno la potenziale immortalità del titolo e l'indebolimento del potere del privilegio.

Questi fattori saranno assoggettati ai proprietari del motore grafico. Rischiereste di rafforzare il potere del privilegio, aumentando la popolarità di uno strumento chiuso, a discapito di alternative libere ed eque.

Godot è un motore multiplatforma libero, generico, in 2D e 3D.

<<https://godotengine.org/>>

Siamo a 2/3

Scoperto il software libero e adottata la sua filosofia, abbiamo ottenuto la libertà. Ora ci troviamo dinnanzi alla sfida di resistere alle forze avverse.

La libertà è più faticosa della sua assenza, perché richiede una costante modifica del comportamento per tutelare le nuove minoranze.

Cos'è che vogliamo evitare soprattutto? Un abuso di potere.

Per evitare l'abuso di potere, dobbiamo dividerlo, decentralizzarlo.

Perciò per mantenere la libertà dei videogiochi, dobbiamo avere a cuore la decentralizzazione dei sistemi.

Col minimalismo possiamo ottenere un risultato pratico.

Usare motori grafici liberi e specifici

Godot è un traguardo fantastico, ma non dobbiamo sognare che diventi lo standard, poiché perderemmo la forza della diversità, che è un pilastro della decentralizzazione.

Un motore generico è più oneroso da mantenere senza una struttura di tipo industriale, che vogliamo evitare.

Cercate un gioco di software libero dello stesso genere che avete in mente e modificatelo per le vostre esigenze.

Scegliete un motore piccolo, che sia facile da mantenere al massimo da tre persone. Meglio se voi stessi rientrate in questo numero.

Sviluppare su GNU/Linux o altri sistemi liberi

Siate il cambiamento che volete vedere negli altri!

Dobbiamo cambiare radicalmente le nostre abitudini informatiche, se vogliamo un futuro con risorse distribuite equamente.

I sistemi attuali non sono degli strumenti, ma delle scatole nere le cui potenzialità sono limitate artificialmente, in funzione di un profitto gigantesco di pochi.

Sviluppare per Linux o BSD vi renderà un programmatore migliore, perché potrete studiare il funzionamento di ogni programma sul vostro sistema e perché la sua natura modulare vi spronerà a non legarvi a un unico paradigma. Sarà più facile scrivere codice multipiattaforma, pure per dispositivi "esotici".

Scrivere il proprio motore grafico

Con il vostro motore grafico potrete cucirvi il processo di lavorazione su misura! E mantenerlo semplice e piccolo, decentralizzato.

È un ottimo esercizio, attraverso il quale acquisirete numerose competenze trasversali.

I videogiochi si fanno programmando, non pensate di poterla fare franca senza mettere mano al codice: o vi illudete o state delegando.

Non esiste conoscenza che non sia potere.

Non dimenticate di pubblicare il codice con una licenza libera!

Scrivere codice facile da portare e da leggere

Per diffondere il messaggio del gioco sovversivo, fate in modo che giri su più dispositivi possibili.

Non basta un motore multipiattaforma, dovete strutturare il codice affinché la trasposizione da un sistema operativo all'altro sia indolore.

Sottolineo l'importanza, quasi scontata, di un codice facile da leggere, in particolare per la diffusione della conoscenza.

Scrivere codice facile da portare e da leggere

Ecco alcuni consigli, in ordine di onerosità crescente:

1. **Provare il gioco sul computer meno potente che avete.**

Non snobbate i PC "vecchi", dite NO allo spreco di componenti elettroniche!

2. **Diminuire le linee di codice.**

È più facile comprendere file brevi. Lo stesso dicasi della portabilità.

3. **Diminuire il numero dei file.**

È più facile comprendere un programma se non è sparpagliato in migliaia di file. Lo stesso dicasi della portabilità.

4. **Evitare la virgola mobile.**

Un sacco di processori si affaticano con la virgola mobile, soprattutto quelli dei dispositivi esotici e vecchi, come le console classiche. Una logica a numeri interi è pure più facile da collaudare.

Scrivere codice facile da portare e da leggere

5. Evitare l'accelerazione hardware.

La GPU non dev'essere un requisito. I vecchi portatili e i dispositivi mobili esotici non dispongono di scheda grafica. Le schede video non parlano tutte la stessa lingua. Un render software vi farà risparmiare batteria. Per giochi indipendenti in due dimensioni, non avete bisogno della scheda grafica. Usatela al massimo per l'ingrandimento della risoluzione o per dei filtri.

6. Evitare implementazioni dipendenti da architetture a 64 bit.

Uno dei motivi per il passaggio a 64 bit è per aumentare la capacità della RAM oltre i 4 GB. Il software libero e GNU/Linux girano benissimo con soli 4 GB di RAM. Compilate e distribuite il gioco anche per 32 bit. Non escludete tanti processori vecchi che il progresso ha dismesso prematuramente. Un processore a 32 bit consuma meno di uno a 64.

Scrivere codice facile da portare e da leggere

7. **Separare la logica dall'implementazione.**

Seguite il funzionamento del vostro codice, non quello del sistema per cui sviluppate. Create delle funzioni involucro, di cui scriverete l'implementazione a seconda della piattaforma di compilazione. Per compilare il vostro gioco su un altro sistema, vi basterà scrivere le funzioni che si appoggiano alle API.

8. **Non dare per scontato una determinata periferica audio.**

Se il gioco funziona anche senza audio, è meglio sia per la portabilità che per l'accessibilità. Non tutte le periferiche audio sono uguali, in alcuni casi i giocatori potrebbero sentire i suoni in maniera diversa. Fate in modo che l'audio non sia mai l'unico vettore di informazioni.

9. **Non dare per scontato l'accesso al file system.**

Questo consiglio può sembrare anacronistico coi computer di oggi, però cambia con quelli vecchi o le console. Si applica sia al caricamento dei dati, più impegnativo, che per il salvataggio. Vi ricordate le password?

Scrivere codice facile da portare e da leggere

10. **Non dare per scontato una determinata risoluzione e un rapporto.**
Questo è più arduo, quasi impossibile se disegnate gli sfondi per una certa inquadratura, come nelle avventure grafiche. Ma nei giochi in cui l'immagine viene composta da caselle, potete supportare diverse risoluzioni e rapporti.
11. **Pensare il gioco per risoluzioni basse.**
Usate come riferimento principale la risoluzione più bassa che avete scelto di supportare e poi, se serve, aggiungete dettaglio. A volte basta un 640x480 ingrandito con un'interpolazione lineare o un 1280x720.
12. **Non dare per scontata un'uscita video a pixel.**
Forse estremo, non per questo meno valido. Tanti giochi funzionano alla grande con interfacce testuali. Si pensi alle avventure testuali, ai dungeon crawler, ai roguelike, ai giochi di commercio e agli strategici simulativi.

Scrivere codice facile da portare e da leggere

Non solo non avete bisogno della scheda grafica, potete fare a meno della grafica a sprite!

Adottate un approccio modulare, separate la logica dall'implementazione, prevedete diversi casi d'uso e cercate di favorire la facilità d'implementazione rispetto a quella d'interfaccia, tranne quando tale scelta cozza con la leggibilità o il buon senso.

Usare linguaggi di programmazione il più indipendenti possibile

Un linguaggio di programmazione indipendente non dev'essere controllato da unico ente, sia nelle funzionalità che nella possibilità pratica di creare compilatori.

Ricordatevi dell'importanza di controllare i mezzi di produzione per liberare i lavoratori! È d'obbligo usare strumenti che non si ritorcano contro la gente comune, plasmati e limitati da una ditta enorme per inseguire il suo profitto.

Consiglio vivamente C e un po' meno C++.

Escludo C# e Java, in mano a Microsoft e Oracle. Neanche Rust e Go sono da tenersi in seria considerazione.

Per compiti più leggeri, magari di automazione, potreste impiegare il linguaggio della shell, Lua o Python.

Vi basta C, shell, le utilità base di GNU, sed e awk.

Consigli per C

Aderite alla revisione dell'89/90 (ANSI) o a quella del '99.

Sono le versioni di C più diffuse, con un grande numero di compilatori per i sistemi più disparati.

Provate compilatori diversi da GCC o Visual Studio.

Su GNU/Linux e BSD, evitate Autoconf a favore di semplici Makefile con configurazioni differenti scritte come commenti.

Siete arrivati in cima alla vetta!

Rispettate i videogiocatori distribuendo i giochi senza barriere economiche o tecnico-pratiche.

Usate programmi liberi, su sistemi operativi liberi.

Pubblicate il vostro lavoro con una licenza libera, per farlo leggere e modificare da chiunque ne abbia voglia.

Rimanete in pochi, con un forte sentimento d'indipendenza.

Coltivate uno spirito minimalista per soppesare le decisioni da prendere. Se in dubbio, prediligete soluzioni vecchie e rodate alle novità di tendenza, a meno che esse non vi permettano maggiore flessibilità nel lungo termine.

Come per ogni linea guida che si rispetti, non abbiate paura d'infrangere quanto detto prima, se dovete agire a discapito del buon senso.

Bibliografia

The C Programming Language (1988)
di Brian W. Kernighan e Dennis M. Ritchie

Programming Linux Games (2001)
di John R. Hall

The Unix Programming Environment (1984)
di Brian W. Kernighan e Rob Pike

The Debian Administrator's Handbook (2003-2022)
di Raphaël Hertzog e Roland Mas

Open source, software libero e altre libertà (2018)
di Carlo Piana

Sitografia

Elenchi di giochi liberi

<https://libregamewiki.org/Main_Page>

<<https://osgameclones.com/>>

Come scegliere una licenza libera

<<http://www.gnu.org/licenses/license-recommendations.it.html>>

Strumenti per scrivere un motore grafico

<<http://wiki.libsdl.org/SDL2/FrontPage>>

<<http://www.tastyfish.cz/#projects>>

Risorse grafiche e audio libere e gratuite

<<https://opengameart.org/>>

<<https://freesound.org/>>

Elenchi di programmi liberi e minimalisti

<https://directory.fsf.org/wiki/Main_Page>

<https://wiki.installgentoo.com/wiki/List_of_recommended_GNU/Linux_software>

<<https://github.com/naav97/guides/blob/master/ALTERNATIVES.md>>

<<http://suckless.org/rocks/>>

Ludografia

Anarch <<https://drummyfish.gitlab.io/anarch/>>

MicroTD <<https://drummyfish.itch.io/microtd>>

DOOM <<https://github.com/id-Software/DOOM>>

Simon Tatham's Portable Puzzle Collection

<<https://www.chiark.greenend.org.uk/~sgtatham/puzzles/>>

Catacombs of the damned! <<https://github.com/jhhoward/Arduboy3D>>

Arduventure <<https://github.com/Team-ARG-Museum/ID-46-Arduventure>>

1 on 1: Racer <<https://fmanga.itch.io/1-on-1-racer>>

Dungeon Crawl Stone Soup <<http://crawl.develz.org/>>

GNU Chess <<http://www.gnu.org/software/chess/>>

.kkrieger <<https://it.wikipedia.org/wiki/.kkrieger>>

Ludografia

C-Dogs SDL <<http://cxong.github.io/cdogs-sdl/>>

Dave Gnukem <<https://djoffe.com/gnukem/>>

Triplane Classic <<https://triplane.sourceforge.net/>>

Contatti

Diapositive <<http://choice.tiepi.it/~matteobin/scritti/software-libero-e-minimalismo-per-sviluppo-indipendente.html#diapositive>>



Email matteobin@tiepi.it

IRC <<ircs://irc.oftc.net:6697/zonawarpa>> (matteobin)