

Sviluppo di videogiochi in Linux



Chi sono?

- Studente di informatica
 - 1° anno, **UNIME**
- Appassionato di informatica e videogiochi
- Programmatore autodidatta



Di cosa parliamo?

- La mia esperienza con **GNU/Linux**
 - Progetti e risultati
- **GNU/Linux** come ambiente di sviluppo
- Sviluppo **open-source**
 - Creare videogiochi – **perché?**
 - Programmazione *sociale*
 - Opportunità di guadagno
- **Live coding:** *clone di Breakout*

I miei progetti

Open Hexagon

- Gioco arcade gratuito, open-source, multi-piattaforma
 - Nato come un clone di “*Super Hexagon*”
 - Fanbase di circa 2000 giocatori (*Facebook*)
 - **7500** risultati per “Open Hexagon” su YouTube
 - **C++11**, Lua, JSON

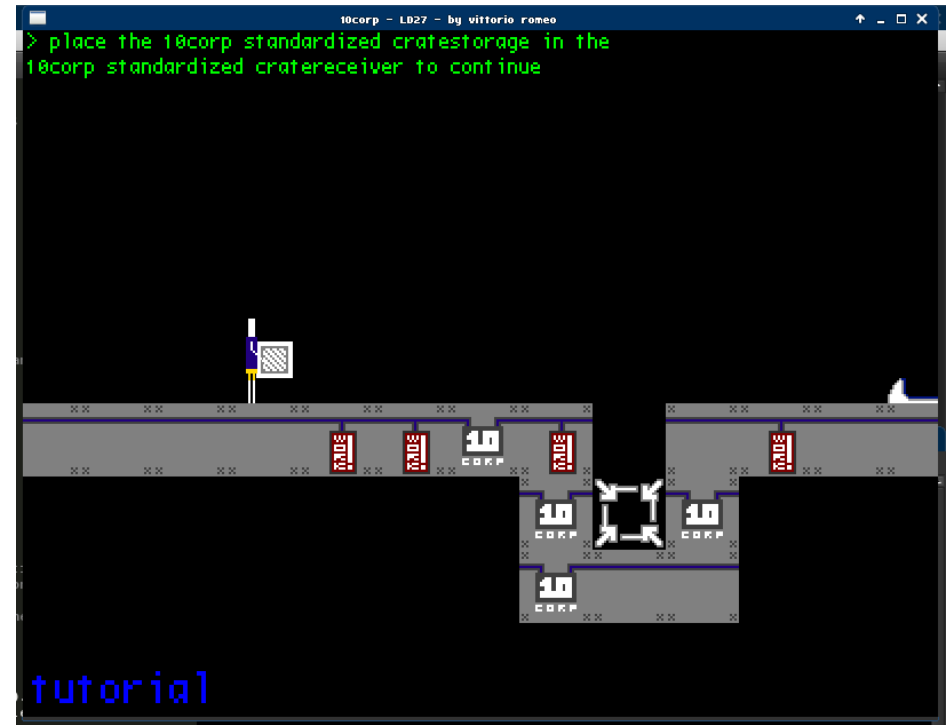


TIME: 0.483
OFFICIAL MODE
SWAP ENABLED
SPECIAL: NONE



10corp

- Gioco puzzle-platformer gratuito, open-source, multi-piattaforma
 - Creato per la competizione “*Ludum Dare 27*”
 - C++11, JSON



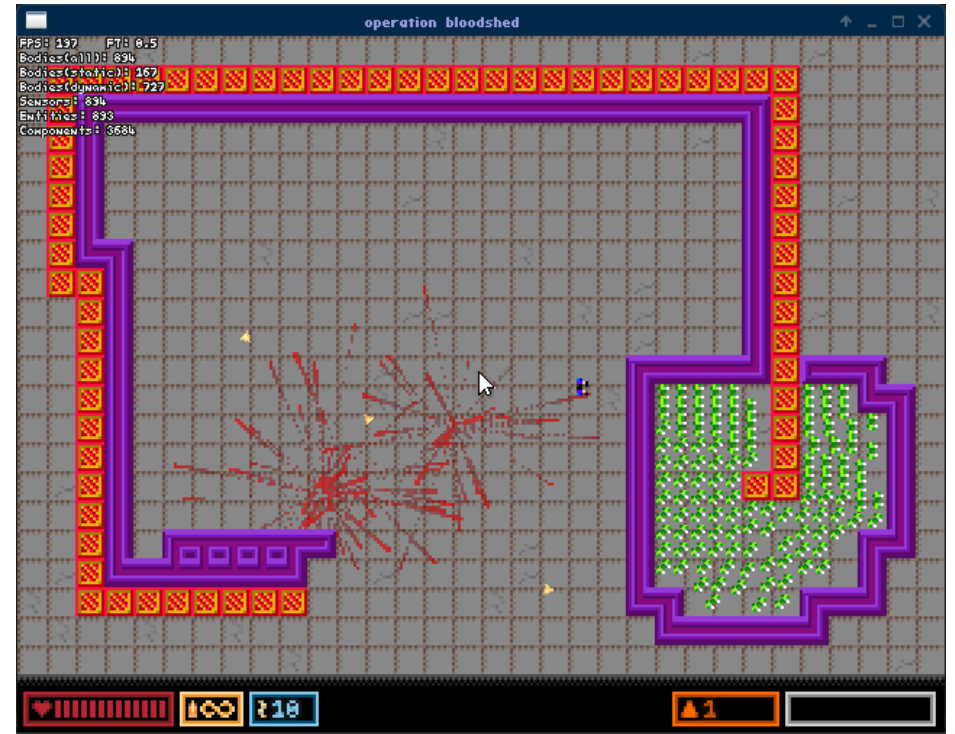
> this is your final task for today
good luck, worker



9.15001

operation bloodshed

- Gioco azione/arcade retro
 - In fase di lavorazione



Z: 0
Level NY: 0/1
C/W: save/load Z/R: cycle 2
A/S: cycle id Q/W: cycle param
LShift: pick
N/N: copy/paste params

id(1)
open(0)



Perché programmare videogiochi?

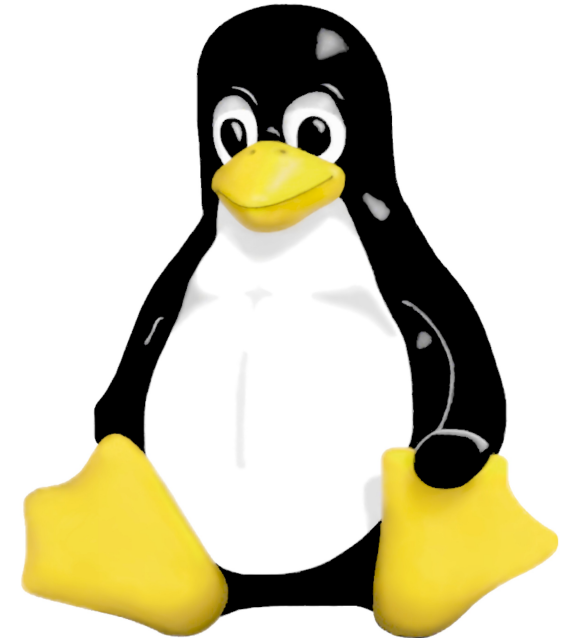
Gamedev: esperienza a 360°

- Un videogioco ha bisogno di:
 - Design
 - Codice
 - Grafica
 - Musica
 - Sonoro
 - Storia
 - Community
 - ...



Come realizzare tutto ciò che serve?

GNU/Linux + strumenti open-source



Sviluppo con GNU/Linux

- Controllo **totale**
 - **Nessuna** limitazione
- Esperienza **educativa**
 - Capire **veramente** come funzionano le cose
- Collaborazione **open-source**
 - **Innumerevoli** risorse
 - Se qualcosa manca... **si contribuisce**
- **Nessuna** spesa richiesta
 - Tutto disponibile **gratuitamente**
- Massima **efficienza**
- Target **universali**
- **Grande community**



Cosa ci guadagno?

Guadagni economici

- **Modello Freemium**
 - Evitare il **pay-to-win**
- **Donazioni**
 - Incentivare con **reward**
- **Dual-licensing**
 - Esempi: **Qt** e **OpenOffice**
- **Kickstarting**
- Opportunità lavorative
- Codice **gratis**, risorse a **pagamento**



Google wallet



indiegogo



Guadagni personali

- **Esperienza**
 - Lavorare su **GNU/Linux**
 - Scrivere codice **efficiente** e **portabile**
 - Gestire e creare **risorse**
 - **Distribuzione** e **networking**
- **Soddisfazione**
 - Realizzare un **gioco completo**
 - Ottenere **feedback**
- **Community**
 - **Fan-base** intorno ai propri progetti
 - **Target** esistente per progetti futuri



Guadagni universali

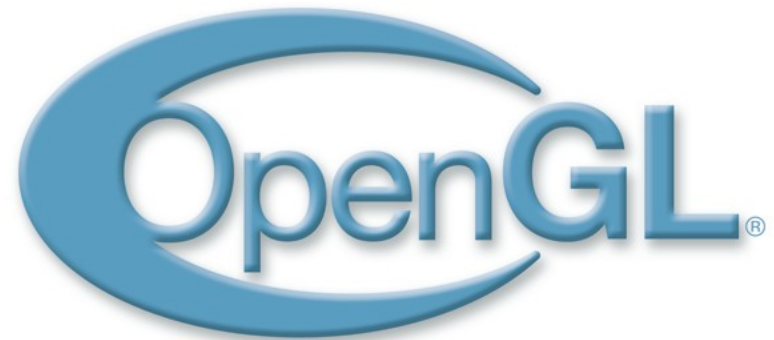
- **Librerie, snippet**
 - Qualsiasi contribuzione **open-source** può tornare utile!
- **Documentazione, Q/A**
 - StackOverflow, ...
 - Creazione **postmortem**, **tutorial**, **articoli**
- Un'esperienza **nativa** videoludica **in più** per **GNU/Linux**
 - Più giochi = più utenti



Da dove inizio?

Open-source: infinite possibilità

- C & SDL
- C++ & SFML2
- Python & PyGame
- Lua & Love2D
- Java & LibGDX
- JavaScript & WebGL
- ...



Live coding: clone di **Breakout**

Live coding: Breakout

- 160~ linee di codice
 - C++11
 - Nuovo standard
 - Più moderno
 - Più veloce
 - Più conciso
 - SFML2
 - Wrapper C++ per:
 - OpenGL
 - Libsndfile
 - GLFW



Partecipa al *live coding*!

Compilatori C++11

- **GCC**

- Versione 4.6.0: compila con ``-std=c++0x``
- Versione 4.7.2: compila con ``-std=c++11``

- **Clang**

- Versione 2.8: compila con ``-std=c++0x``
- Versione 3.0: compila con ``-std=c++11``



Libreria SFML2

- File binari
 - <http://sfml-dev.org/download/sfml/2.1/>
 - **Ubuntu:** apt-add-repository
ppa:sonkun/sfml-development
 - **Arch Linux:** pacman -S sfml
- Sorgenti
 - <https://github.com/LaurentGomila/SFML>
 - Richiede cmake per la compilazione
- Oppure...



Dati livecoding

<http://vittorioromeo.info/Downloads/datiLC.zip>

oppure

<http://www.mediafire.com/?55q89mv5yarsgci>



<http://vittorioromeo.info>



Domande? Commenti?
vittorio.romeo@outlook.com

Grazie per l'attenzione!