Do you even emulate… (Super Mario) bro?

Panayiotis “Pan” Talianos
@panintended:fosdem.org
February 6th 2021 @ FOSDEM
About the speaker

- Electronics, DIY
- Swimming
- Software developer in telecomms
Talk details

• *Why* writing an emulator is awesome
• Brain dump → objective arguments
• Why the “why”?  
• Target audience:  
  – Solid foundation in programming, CS fundamentals  
  – Curious about a computer’s inner workings  
  – Interest in retrocomputing/gaming console history
The journey, vol.1: Why this talk?

- Interest in console emulators → oddly specific
- 1st Emudev track @FOSDEM!
- But *why* are emulators so cool?
- Find an answer! (and inner peace?)
- Resonates with others? → this talk!
The journey, vol.2: Down to basics

Facts:

- I like computers (I'm sure you weren't expecting this...)
  - Video games (NES, PlayStation, N64)
  - Early 2000s → first PC → emulators
- Curious about how things work on the inside
- “How did this come to be?” (i.e. the story behind it)
  - Sony/Nintendo gaming console partnership [2]
The journey, vol.3: year of the dragon CHIP-8

- Early 2000s load savestate... flashback
  - Emudev people == Gandalfs
  - But how to... emulate them? (heheh)
  - Do you even program, bro?
- Mid-2010s light bulb moment: “How about now?”
- Chip-8 interpreter → tons of fun
The verdict, vol.1: you have fun

- Emulator development ≈ beating a video game
  - Core functionality ≈ main story
    - Well-specified hardware & software ≈ “begin” / “end”
    - System specification ≈ the rules of the game
  - Extra features ≈ side missions
    - Fast-forward
    - Savestates
    - Go back in time!
    - Improved graphics, etc.
The verdict, vol.1: you have fun

- Computing done backwards
- Build a computer, no hardware required
- Once done, the software just works™ *
- Relatable context (video games)

* a.k.a. “When I run my code for the first time” [3]
The verdict, vol.1: you have fun

• End result: self-explanatory, relatable (not just for you!)
• Show it off, expect people to actually... “get it” :
• Bonus: you don’t have to write the game!
The verdict, vol.2: you learn

• Technical
  – Registers, memory, interrupts, assembly, etc.
  – Not university coursework!

• History
  – Crash Bandicoot (PlayStation hacks) [5]
  – Resident Evil 2 (fitted onto 64 megabytes!) [6]
The verdict, vol.3: you appreciate

• Best practices
  – Efficient code
  – Maintainable code
  – *Use the debugger* (you will write one, too!)

• The luxuries we have today
  – Tools, libraries, SDKs, high-level languages
  – Day-1 patches
    • Gran Turismo 2 NTSC-J (could not get past 98.2%!) [7]
The verdict, vol.3: you appreciate

- FOSS/emudev/homebrew communities
  - Academia
    - Technical & historical context
  - Computer history preservation
    - Through documentation/emulation
    - People still write Game Boy games!

[8]
An awesome journey

- Fun, educational
- Humbling, inspiring
- A better engineer
Y u no emudev??

• It doesn’t *have* to be C/C++/Rust *

• Online communities
  – Reddit, Discord, FOSS projects, blogs, etc. [9]
  – Guidance, tons of resources

• That’s all, have fun!

* but set your expectations accordingly!
See you at FOSDEM 2022!

Feedback  (much appreciated, thanks!)
https://submission.fosdem.org/feedback/11536

References
[1] GIF from "The Room" (by Wiseau-Films)
[6] Resident Evil 2 on N64: https://www.youtube.com/watch?v=BaX5YUZ5FLk
[8] µCity (Game Boy homebrew): https://github.com/AntonioND/ucity
[9] Emudev communities
   https://www.reddit.com/r/EmuDev/
   https://discord.me/emudev

Contact: @panintended:fosdem.org