

---

---

# How to accidentally start working on Video Sequence Editor

— Aras Pranckevičius —  
Blender Conference 2024

---

---

**Only a small part of this will be about VSE!**  
**The rest is random rambles about code**

# How do you start contributing to Blender?

- Load a house model into Blender and export to OBJ
- “Hmm feels like it should be faster”
- Start profiling what’s going on
- Oh! There’s GSoC happening *right now* with new OBJ I/O, 10x faster!
- Notice that it could be *another 10x* faster
- Do that, while also stealing GSoC project landing from Howard

...and then things go downhill from there

# Wait, “accidentally” start working on VSE?!

- Be in Amsterdam for unrelated reasons
  - Drop by Blender HQ just to see it
  - Get locked up in a room by Francesco & Sergey: “VSE or else!”
- 
- I know *exactly nothing* about video or editing so... *perfect!*
  - I didn't even know that Blender *has* a video editor :)

# A year of accidental VSE work

- 4.1: faster timeline/effects/scopes, improved filtering, better looking scopes
  - 4.2: rounded corners, text outline/shadows, some speedups
  - 4.3: faster thumbnails, faster modifiers, other speedups
  - 4.4: EXR proxies, maybe 10/12 bit videos, ...
  - ...
- 
- Check out [“VSE Workshop: August 2024”](#) on code.blender.org

# Getting into new codebase

## Large codebases often are...

- Fairly old (Blender: some parts 30yo)
  - Little or no documentation
  - Grew and evolved organically
  - There are places that no one remembers what/why/how
- 
- In VSE, I had a dozen findings of *“who did this and when? Oh, Ton 15 years ago in ‘add sequencer’ commit”*

# Initial reaction can be “aaargh what is this?!”

- Most of the time there *is* a reason, it is just not apparent
- Very tempting to say “this sucks, let’s rewrite”
  - Very dangerous too
  - You *probably* do not have enough knowledge why things are this way
- Maybe it would be better to do it differently
  - And maybe even the *original authors* would agree!
  - ...or maybe not. Depends!



# Assume authors are not stupid

If something looks strange/wrong/weird:

- 30%: there is a good reason for it, you just don't know it (yet)
- 30%: there *was* a good reason for it, *back then*
- 30%: there is no good reason indeed, everyone agrees it should be fixed
- 10%: Ph'nglui mglw'nafh Cthulhu R'lyeh wgah'naglhtaiön  
run and don't look back!

# Useful tools and workflows

- Tests!
  - Good test coverage is great! Allows change/refactor/optimize/improve with less fear
  - How about adding new tests *before* starting to change something
- Debugger, duh
- Source control: git blame, history
  - Who changed this, when and why? Can you ask them?
  - Is this strange piece because they were fixing something? Make sure to not regress that
    - ...or even better, ensure test coverage
- Profiler!
  - Many surprises can be found, do not assume *anything*
  - Superluminal on Windows is great, Xcode Instruments on Mac, *dunno about Linux*

# Learning a new area (VSE)

A year ago, I didn't even know that VSE *exists*

- What's "a strip"?
- What's an "effect stack"?
- What's an "I'm buff"? I am not, but *ImBuf* might be.
- How is *ffmpeg* used here? ("*in a very messy way*" is the answer)
- What's a proxy, and are they edible?
- How do caches work?

You *just* kinda learn these concepts as you run into them, while working on the code.

# Dangers of learning a new area

- You do not have *deep expertise* in workflows and tools
  - “Make this existing thing work faster” is very doable
  - “Is this whole system even *good?*” is harder to answer though
- Need to talk with other people who *are* experts!

# Making and shipping changes

- Blender's current process feels quite sane!
  - Just Enough process that it does not turn into complete chaos
  - But not so much that it would slow down / annoy / demotivate people
- It helps if PR description contains *all* the relevant information
  - What problem is it solving?
  - For performance: before / after measurements. "Should be faster" is not enough!
  - For anything UI/UX related: screenshots or gifs/videos help a lot!
  - If large enough change: technical explanations of system(s), algorithms, "why this" etc.
  - If PR has large changes during review, edit description to match latest reality

# Making and shipping changes

- Different aspects might be easier/harder to ship, and that's ok
  - Performance improvement: hard to object to them!
    - Unless it makes the code *much worse*
  - Some UX / Design change: oh boy
    - Everyone has an opinion, at least 5 of them are conflicting :/
- I like this quote from [John Kiril Swenson](#) (GSoC'24):

## Takeaways

- The Blender codebase is really not as scary as I had anticipated.

**...and that's it!**

Meanwhile, VSE will continue getting some improvements