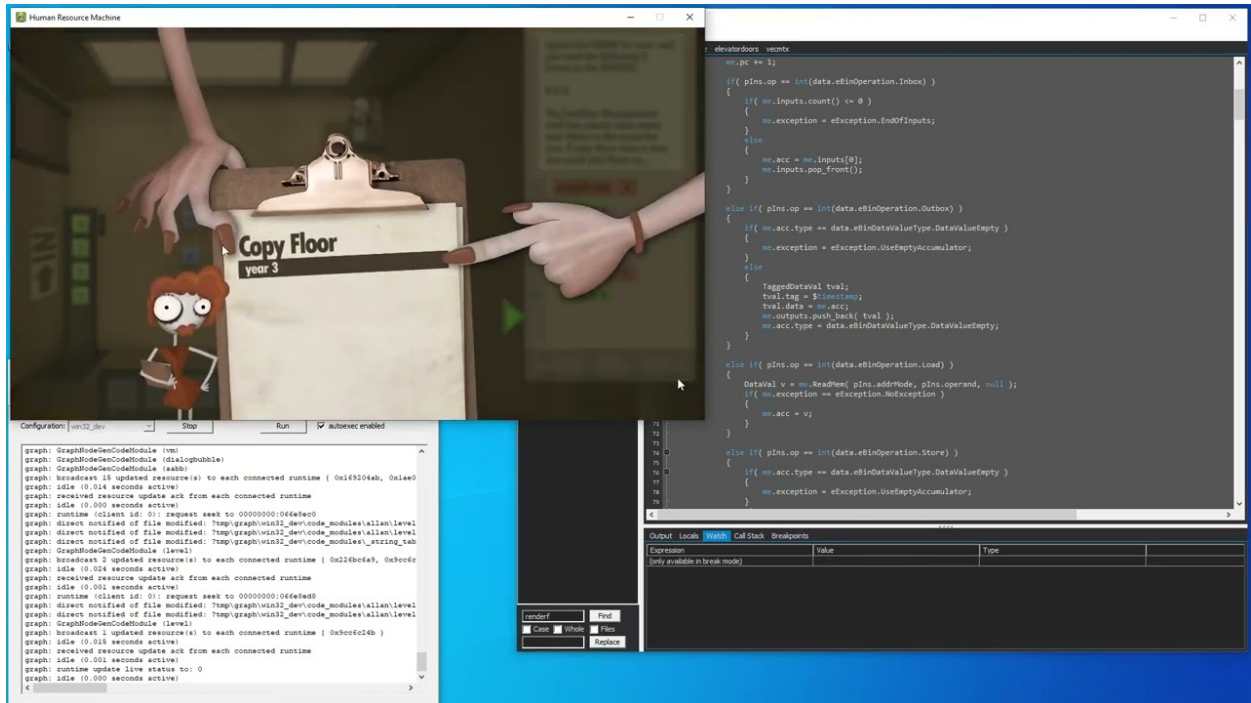


# 2024-04-08 Thinking About

Some thoughts with regard to the Tomorrow Corporation Tech Demo ...

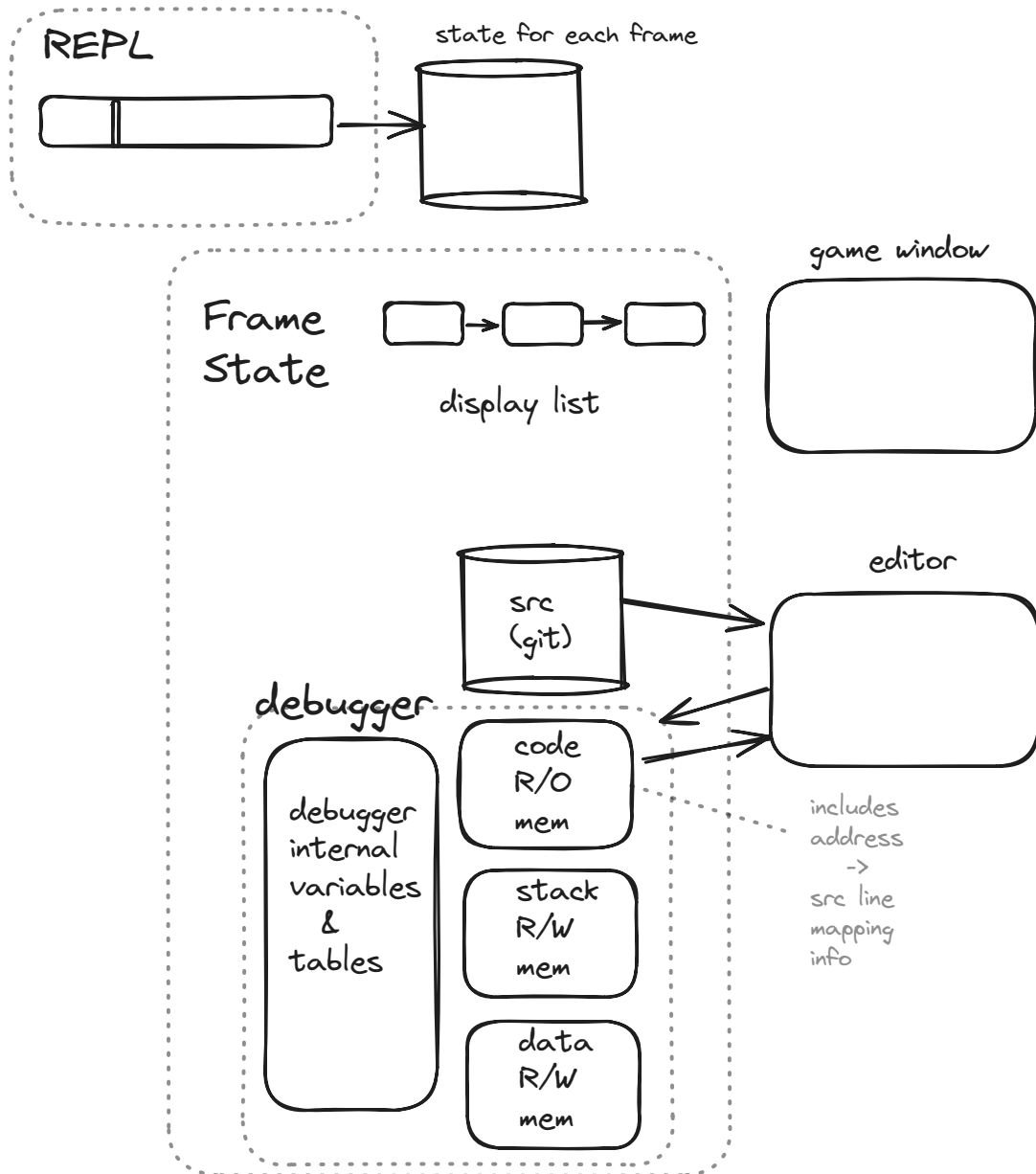


If I think only of an MVI (Minimum Viable Implementation - ignore all concerns for "efficiency"), I see something fairly simple:

Basically, the REPL (the slider) clears the IDE state, peels one frame-state out of the frame-state array (database, whatever) and loads that state into the IDE, re-rendering the game window and the editor window. Re-loading the source code can use git-like technology. Re-loading the display list and the debugger state means pasting into whatever. If you convert the binary data to text, you can use git here, too.

If you own the code for the debugger, the game display engine and the editor engine, then it shouldn't be a problem to ensure that they all use data that is wholly contained in some convenient state location(s) that can be serialized and

deserialized to a database. Modern dev machines are fast and memory-endowed, so there is no reason to optimize this stuff until proven guilty.



## Appendix - See Also

### **See Also**

**References** <https://guitarvydas.github.io/2024/01/06/References.html>

**Blog** <https://guitarvydas.github.io/>

**Blog** <https://publish.obsidian.md/programmingsimplicity>

**Videos** <https://www.youtube.com/@programmingsimplicity2980>

[see playlist “programming simplicity”]

**Discord** <https://discord.gg/Jjx62ypR> (Everyone welcome to join)

**X (Twitter)** @paul\_tarvydas

**More writing (WIP):** <https://leanpub.com/u/paul-tarvydas>