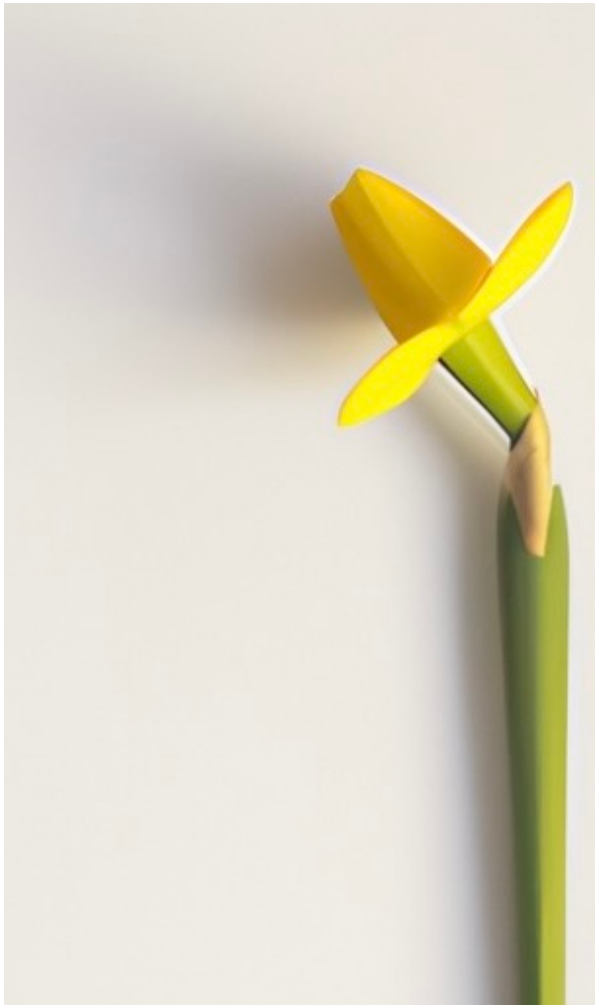
Computers Are A New Medium



[Why Are We Still Using 2D Notations For The New 4D Medium?]

Computers are 4D - a new medium for expression

The previous medium for expression is 2D

2D Medium

- printing press
- paper
- papyrus
- clay tables
- etc.

Advantages

- medium already exists, tried and tested
- inexpensive
- optimized over time
  - scrolls -> books
  - clay tablets -> papyrus -> paper
  - cave walls -> canvas

Disadvantages

- mathematical notation is a workaround to accommodate limitations of the restricted medium
- paper deteriorates over time - doesn't persist through eons
- slow evolution
  - latest advancement is the use of graphite and rubber in lieu of opaque inks
- physics exploration has been constrained by mathematics notation, and, ignores multi-dimensional reality
- Richard Feynman
  - abandoned 2D math notation and used diagrams to explore a specific phenomenon
- Ilya Polyani
  - Nobel Laureate
  - "Order Out Of Chaos"
    - claims that functional-only thinking set physics back by 100 years

4D Medium

x/y/z/t

Advantages

new ways of exploring problem spaces

Disadvantages

- deterioration
  - If power goes out, data lost
  - rapid advancement of storage media, e.g. difficult to read data from only 50 years ago
- Not yet well understood
- Needs invention of new notations

Circling Around The Flame

- Flash
- YouTube
- T.V.
- Film / Video
- streaming
  - Netflix
  - etc.
- Timeline editing for video, and, audio
- Visual REPLs
  - “live coding”
  - Worry Dream

Obsolete Ideas

- desktop metaphor
- text-only programming languages
  - Based on 1950s hardware constraints
  - Based on grids of non-overlapping cells of fixed-size bitmaps
  - Text-based programming languages encourage synchronous expression, whereas Reality is asynchronous
- Filing cabinet metaphor, files
- Documents as static sheets of paper

Multiple Notations

- humans cannot deal with too many dimensions at once
- Physicists learn to use “simplifying assumptions”, i.e. multiple notations to express multiple aspects of the *same* phenomenon
- Mechanical Engineers learn to make drawings of physical objects from at least 3 views
  - Multiple static views were required by reliance on paper (top view, front view, side view)
  - “3D” visualization changes the toolchain
- UNIX pipelines allowed composition of multiple programming languages into single applications
  - Pipelines are conflated with heavy-weight concepts such as “operating systems”, hence, overlooked
  - processes are just closures implemented in a heavy-handed manner (see Greenspun’s 10th Rule)
- MMU hardware
  - Needed to protect apps from one another
    - Game cartridges: different way to achieve the same result, with less software bloat
    - MMUs not needed within a single app
      - a *bug* is just a *bug,* the goal is 0 bugs.
  - Mutual multitasking
    - Overlooked due to conflation with app-vs.-app protection, instead of subroutine-vs-subroutine needs
  - End-users don’t need MMUs
    - Except when using bug-ridden software
    - Developers want hardware assist during development

Programming Simplicity

https://en.wikipedia.org/wiki/Greenspun's_tenth_rule

https://en.wikipedia.org/wiki/Feynman_diagram

https://www.penguinrandomhouse.ca/books/643445/order-out-of-chaos-by-ilya-prigogine-and-isabelle-stengers/9781786631008

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*

X *(Twitter) @paul_tarvydas*

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