Live Programming and Designing of Dynamic Web Applications

Towards a totally RAD Development Experience

Linus Hagemann at FrOSCon 2023
Linus Hagemann

@linushagemann@digitalcourage.social

linushagemann.de
How do we write Software?
Sketchpad (1963) by Ivan Sutherland

direct manipulation

direct feedback

© 1964 MIT - Lincoln Laboratory, edited via rhizome.org
Sketchpad (1963) by Ivan Sutherland

direct manipulation

direct feedback
Smalltalk (1972)  
Xerox PARC  
(OOP)  
GUIs  
self contained  
fully live
Smalltalk (1972)
Xerox PARC
(OOP)
GUIs
self contained
fully live

© Collin Donnell
Squeak/Smalltalk

A gradient fill style is a fill which interpolates smoothly between two colors.

Marcel Taeumel, CC BY 4.0 via Wikimedia Commons
Creators need an immediate connection to what they create... So much of creation is discovery, and you can't discover anything if you can't see what you are doing.

Bret Victor
Inventing on Principle
SOFTWARE-BASED TOOLS ARE TRAPPED IN TINY RECTANGLES.

For years, I’ve been designing tools.

Tools for people making software...

... electronics, music, animation, mathematical systems...

Things with complex behavior.

My focus has always been — how can creators see that behavior?

How can they see what the thing they’re building is actually doing?

And what are powerful ways of seeing so they can understand what it’s doing?

These have generally been software-based tools.

What “software-based” means today is that these tools are trapped inside a tiny rectangle that sits on your desk.

To do your work, you sit at your desk and you stare at this tiny rectangle. And this frustrates me.

Bret Victor with Drawings by David Hellman
Dynamicland
HyperCard by Bill Atkinson
Version 1.2.2 ©1987–88 Apple Computer

Product Manager: Chris Espinosa, Mike Holm
Program Design: Bill Atkinson, Dan Winkler, Ted Kaehler, Bill Fernandez, Carol Kaehler, Adam Paal
HyperTalk Language: Dan Winkler
Printing: Adam Paal
Sound: Ted Kaehler, Mark Lentczner
Help System: Carol Kaehler
User Guide: Scot Kamins
Graphic Art: Kristee Kreitman, Marge Boots
Advice & Support: Mary Sincitico, Bob Goodenough

© https://winworldpc.com
Scratch
Flash

Welcome to Flash 5

Flash is the industry standard for interactive vector graphics and animations for the Web.

If you're new to Flash, move the pointer over a workspace element, or double-click to find out what it does.

Next, follow the included interactive lessons to start learning right away. Choose Help = Lessons and begin with Lesson 1: Creating. Other lessons cover symbols, layers, text, artwork, buttons, and sound. Click here in order to build on important concepts.

When you finish the lessons, you can learn about more advanced features in the Flash 5 Tutorial. That tutorial is available in the Flash 5 book, "Using Flash!". Flash Help includes comprehensive information on all features. To open Flash Help, choose Help = Using Flash. You can look up topics in the index, or begin with Flash Basics.

lively.next
Flash
Lazarus team and J. W. Dietrich, LGPL, via Wikimedia Commons

Lazarus (Delphi)
Lively Kernel (2006)

Dan Ingalls
- self contained
- fully live
- direct manipulation driven
● morphic
● live evaluation & inspection
● all time is runtime
How does that work?
lively.server

lively.lang   lively.source-transform   lively.ast

DOM

Browser
WENN DAS AUTO ÜBERNIMMT
Autonomes Fahren wird durch technologische Neuerungen immer interessanter für Autohersteller und natürlich auch ihre Kunden. Dabei geht es stets auch um die Frage, wie der Autopilot in verschiedenen Situationen reagiert. In dieser kleinen Demo simulieren wir, wie sich ein autonom gesteuertes Auto verhält, wenn es auf langsamer oder schnellere Fahrzeuge trifft, wann es überholt und bremst.

© typeshift
Rapid Application Development
Rapid Application Development
1. Workflows for Designers and Coders with the same Artifact

2. Collaboration (git)
1. Workflows for Designers and Coders with the same Artifact

2. Collaboration (git)

- generate declarative component definitions from direct manipulation operations
- provide means to attach behavior
● reconciliation
● components & parts
● ViewModels & bindings
What next?
Rights for used images lie with the attributed sources.

All emojis designed by OpenMoji – the open-source emoji and icon project. License: CC BY-SA 4.0.

Otherwise this slide deck is licensed under CC BY 4.0. © Linus Hagemann, 2023.