

IN4MATX 133: User Interface Software

Lecture 1:
Introduction & History

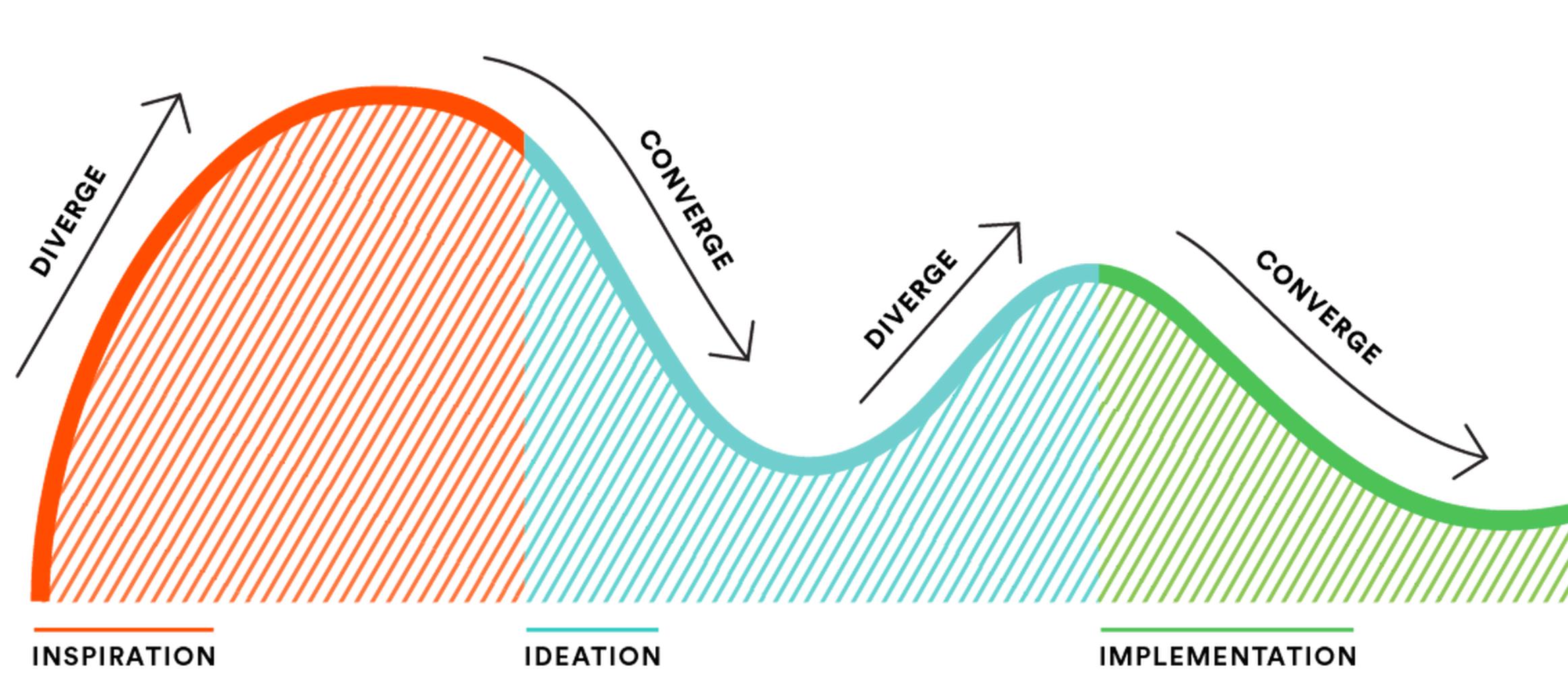
Professor Daniel A. Epstein
TA Goda Addanki
TA Seolha Lee

**I'm thrilled that you have
decided to take this class!**

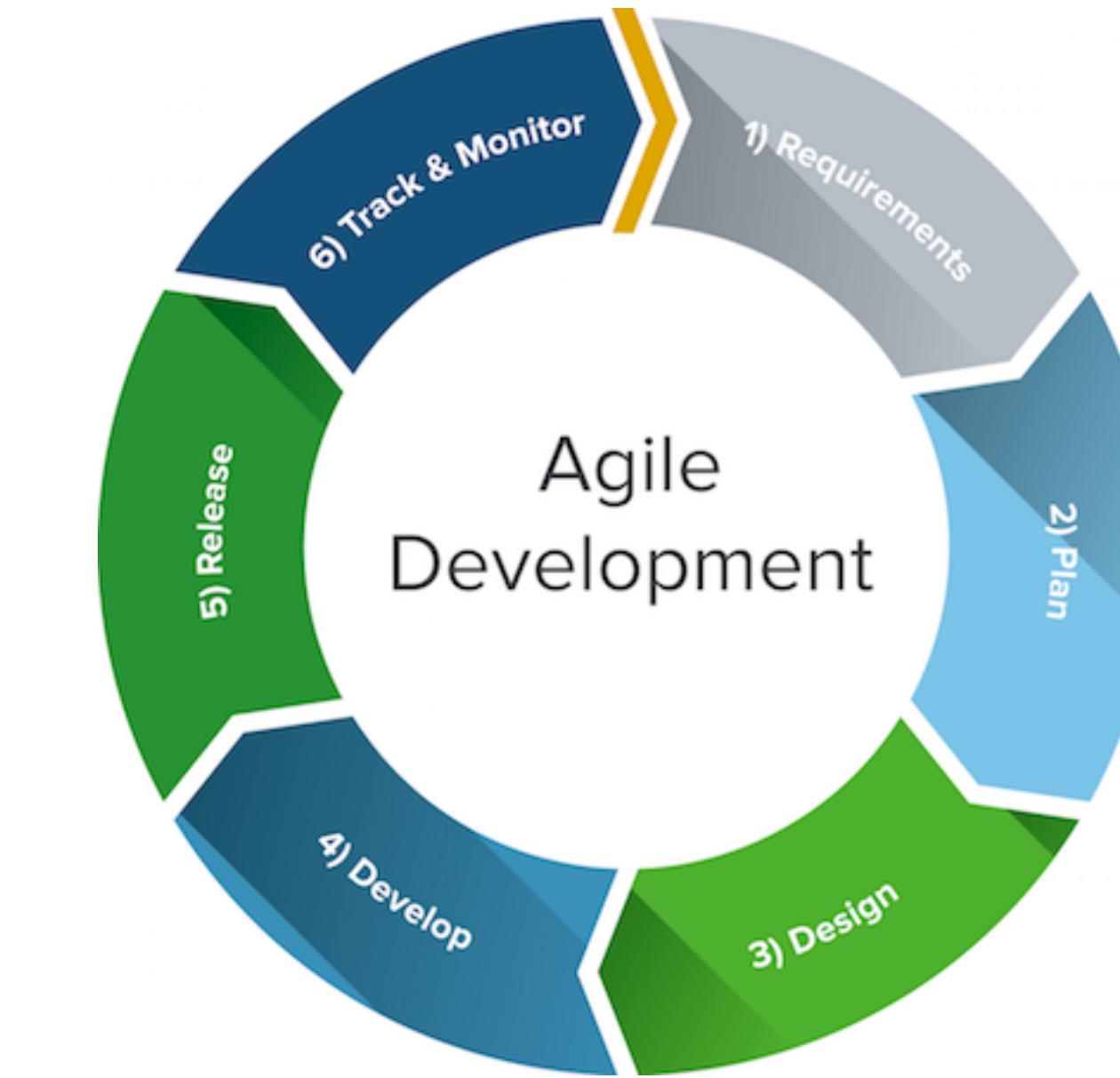
**Learning under COVID times
is extremely challenging.**

**But we aim to make this class
worth your while.**

Product design process

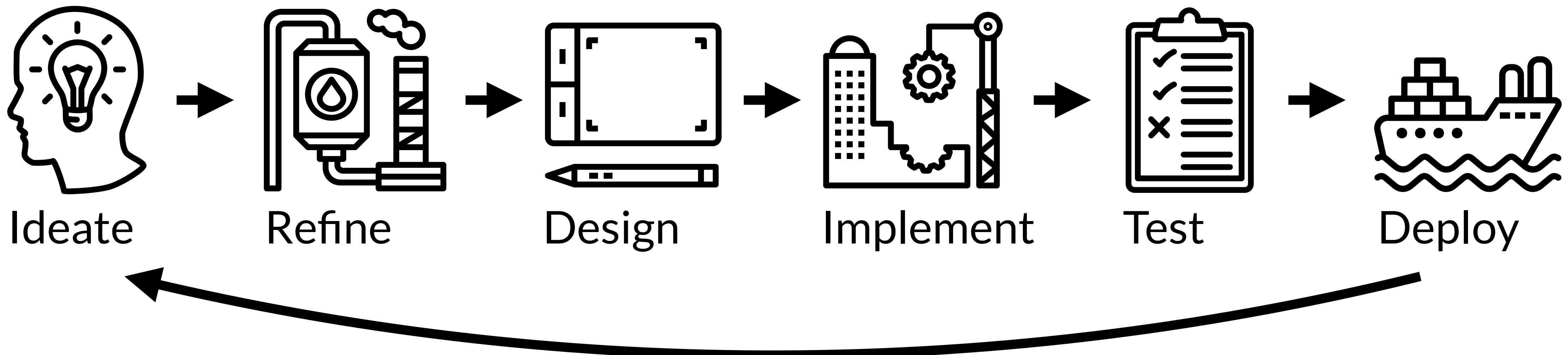


Human-Centered Design, IDEO



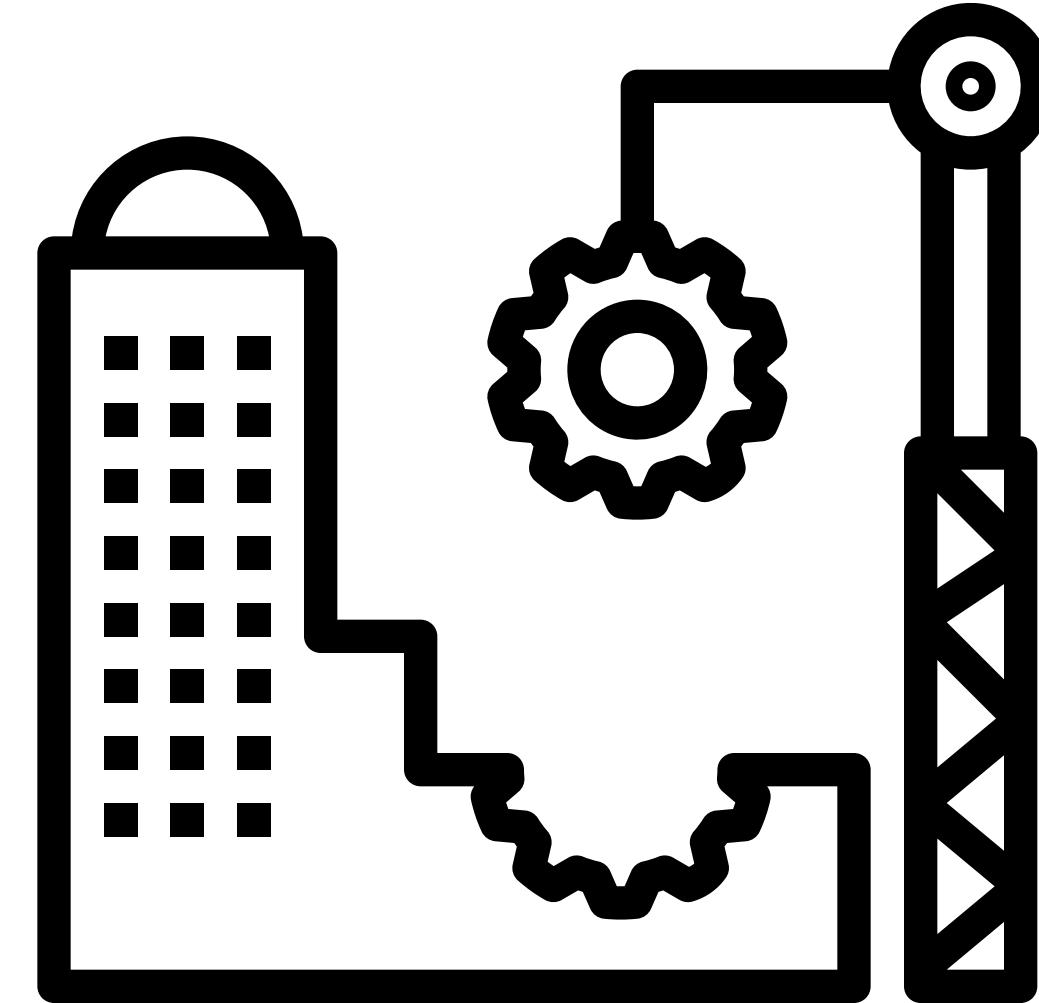
Agile Development, Agile Manifesto

Product design process, simplified



User interface implementation

- Has the power to turn ideas into reality
- Often dictates design decisions and timelines, for better or for worse
- Either you will be implementing, or you will need to communicate with your colleagues who are



What is interface implementation today?

Often HTML, CSS, and JavaScript

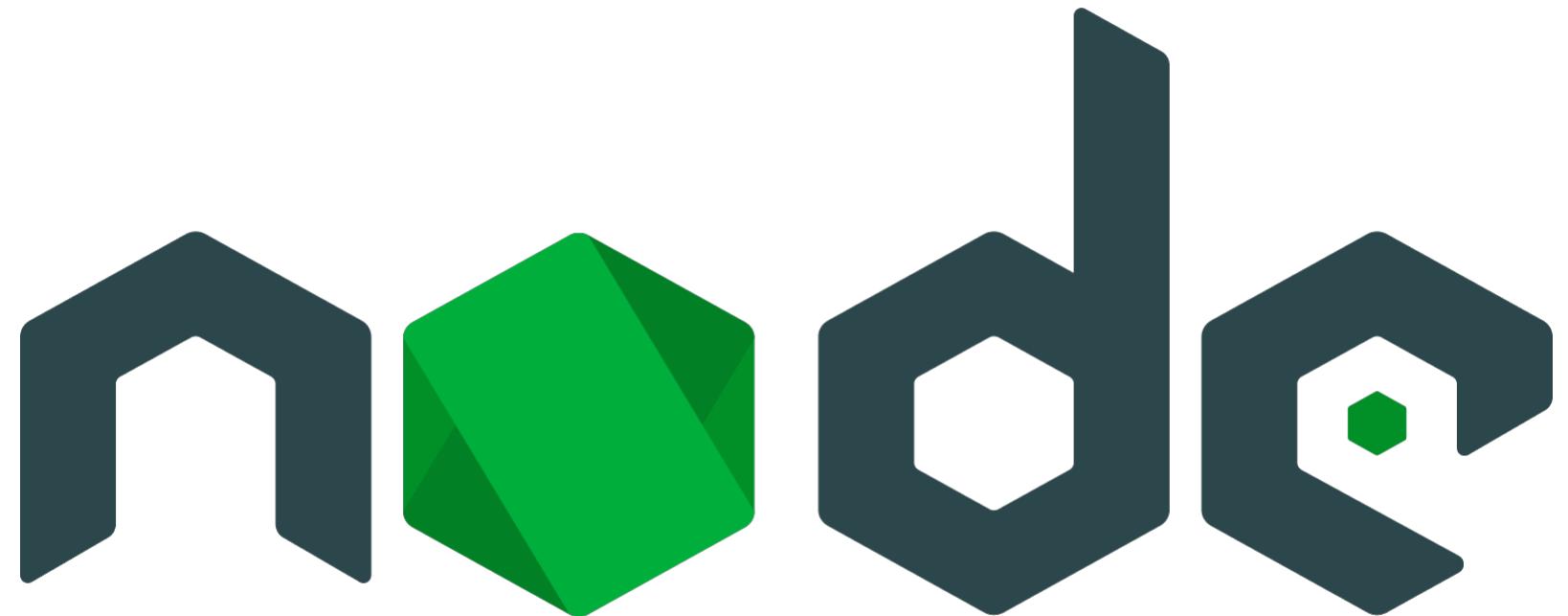


ANGULARJS
by Google



BACKBONE.JS

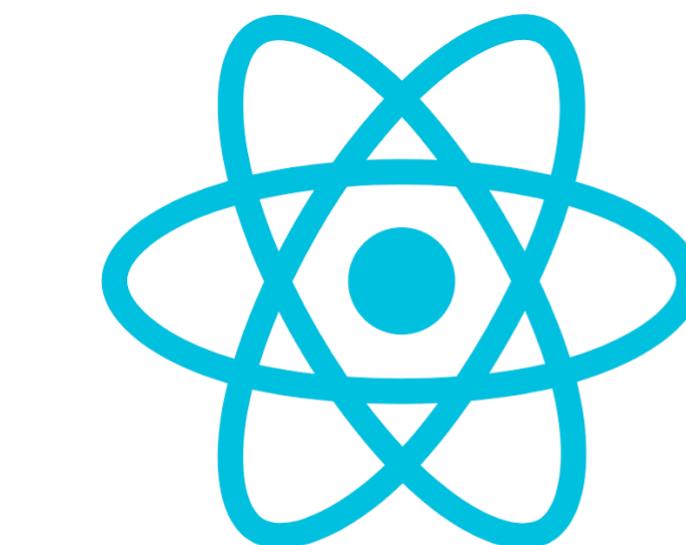
METEOR



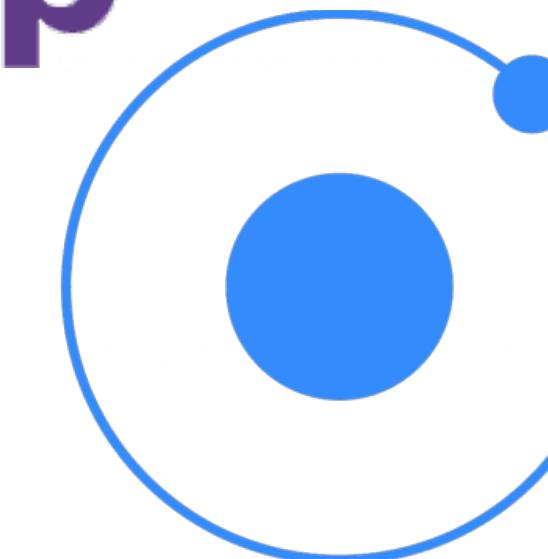
Bootstrap



Vue.js



React JS



ionic

ember

**There are lot of languages
and development frameworks.**

Why do most people use web tools?

Today's goals

By the end of today, you should be able to...

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and know how to find policies
- Describe upcoming course tasks

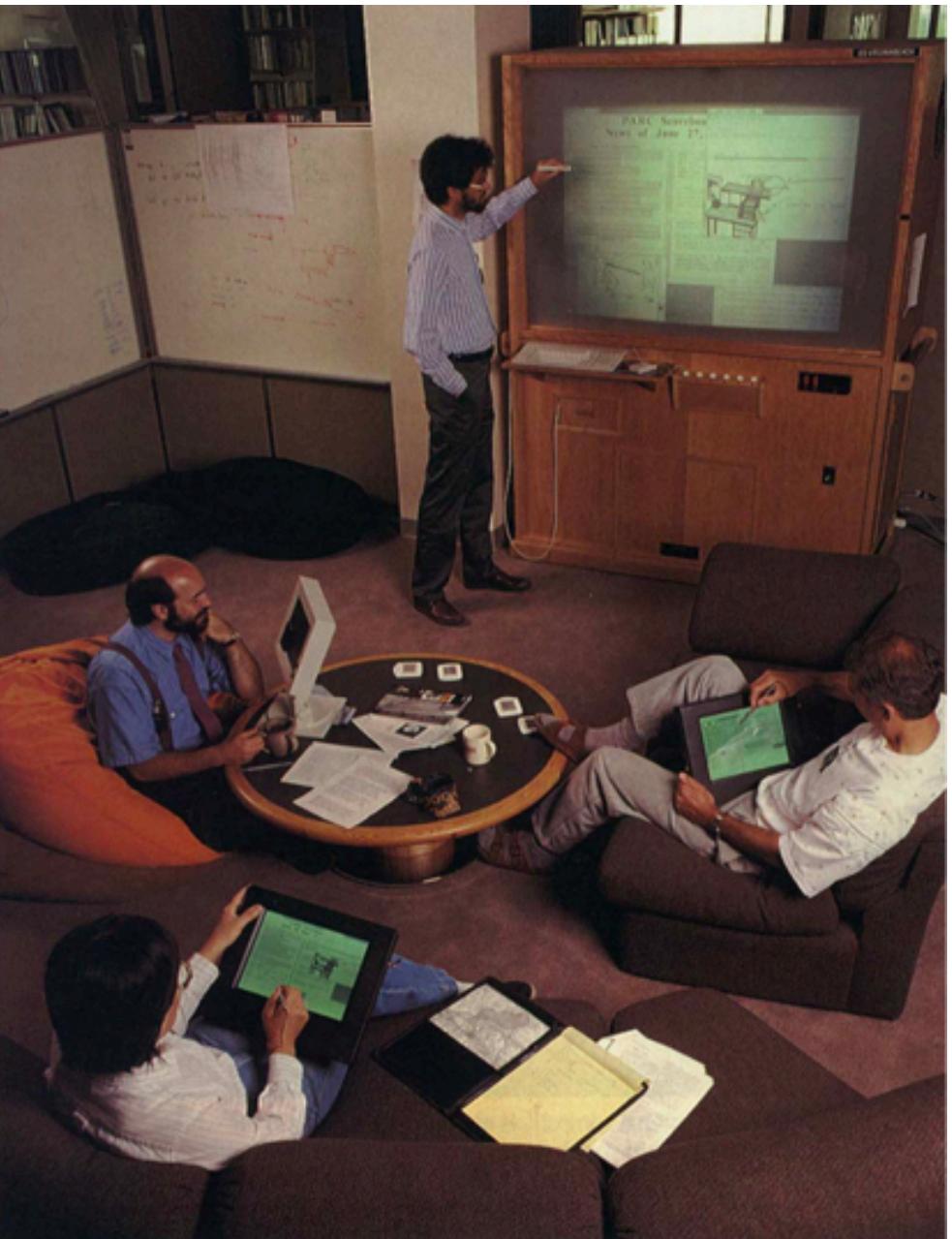
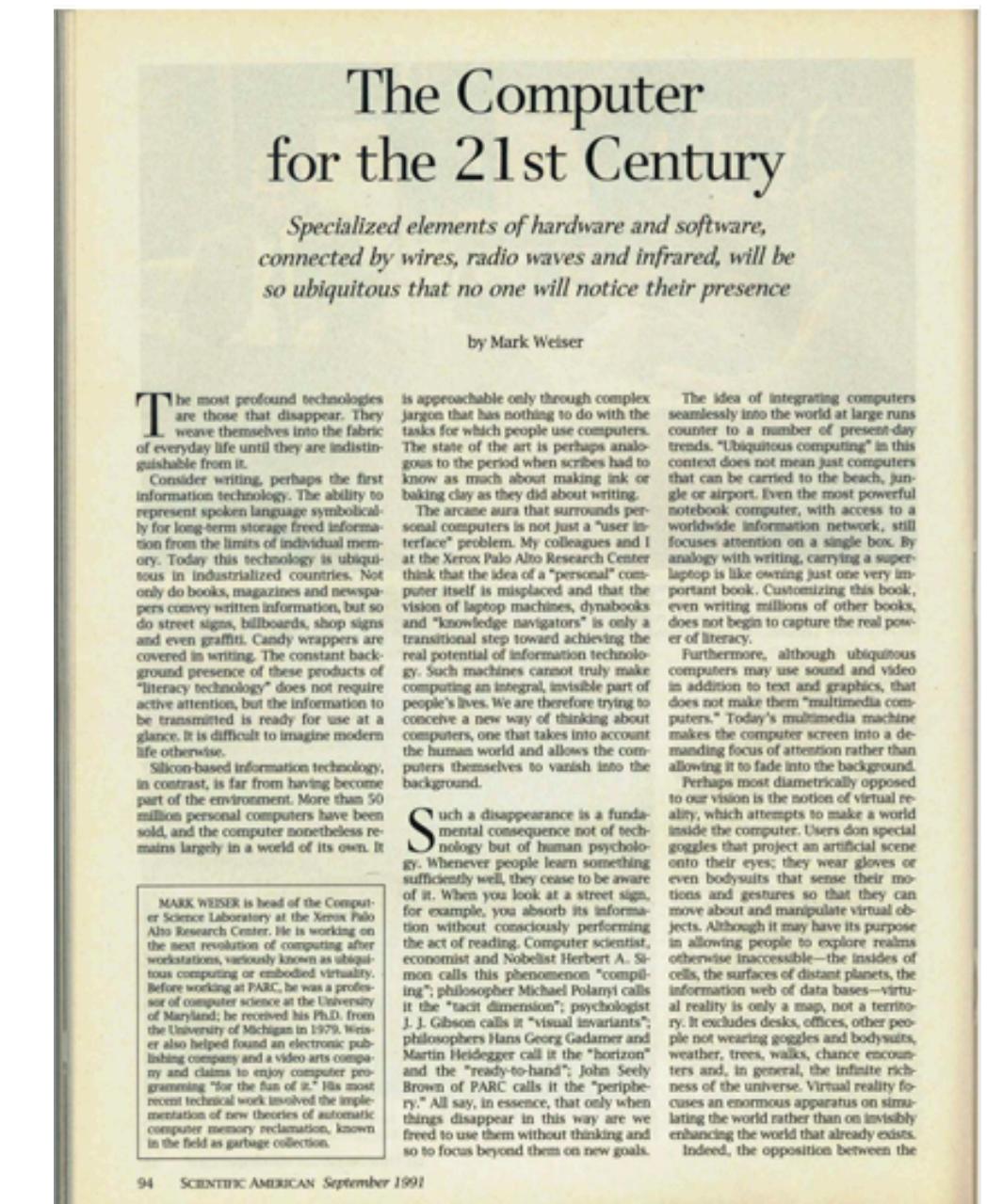
Today's goals

By the end of today, you should be able to...

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and know how to find policies
- Describe upcoming course tasks

The Computer for the 21st Century

- By Mark Weiser, Chief Scientist, Xerox Parc
- Published in Scientific American, 1991
- Coined “Ubiquitous Computing”
 - Reflective and speculative



- <https://dl.acm.org/citation.cfm?id=329126>

Three waves of computing



Mainframe
computing



Personal
computing



Ubiquitous
computing

Three waves of computing



Mainframe
computing



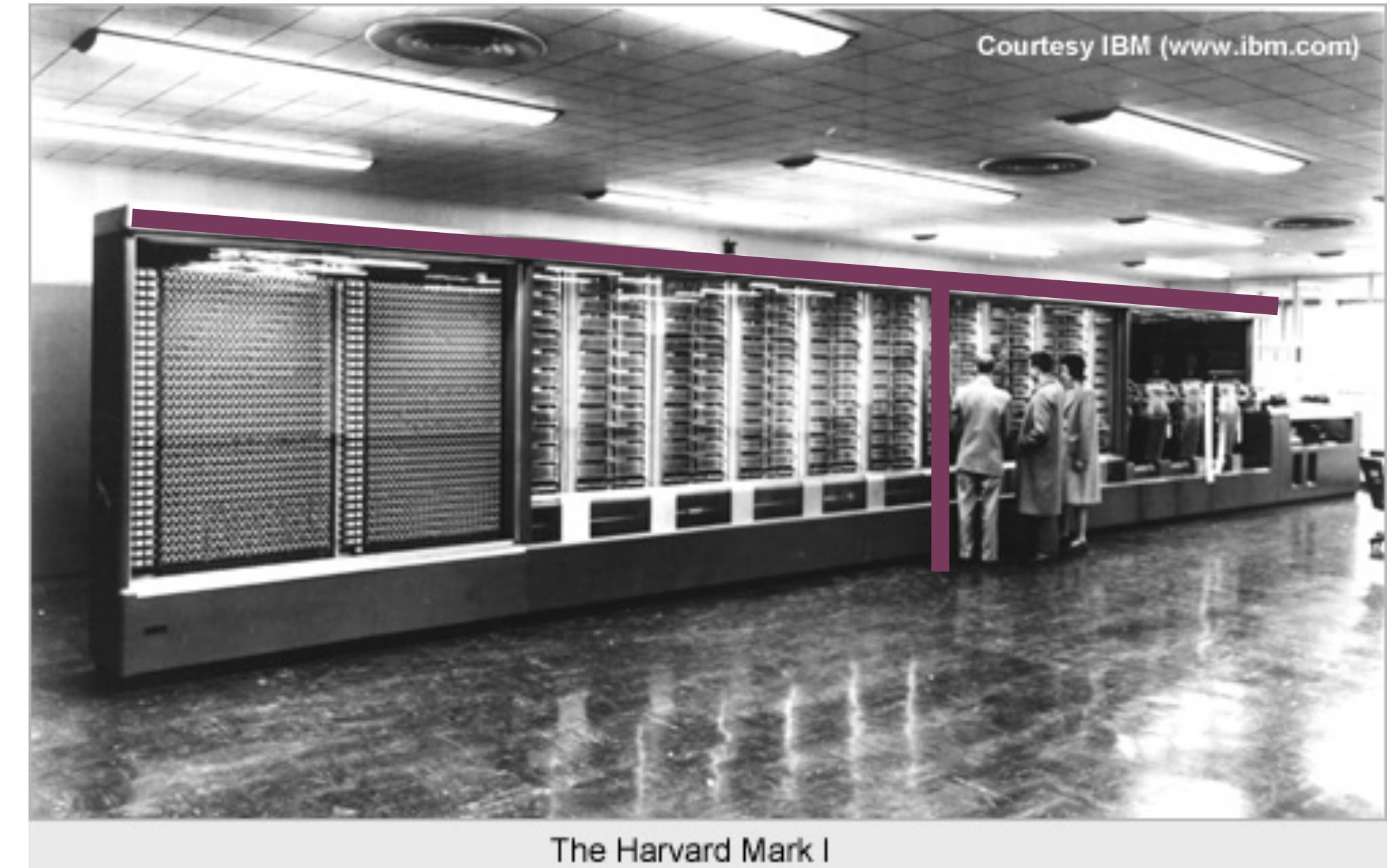
Personal
computing



Ubiquitous
computing

First wave: mainframe computing

- Harvard Mark I
- Large (55 feet wide, 8 feet high, 5 tons)
- Expensive (enclosure alone was \$50,000 in 1945!)
- Used to calculate implosion during the Manhattan Project

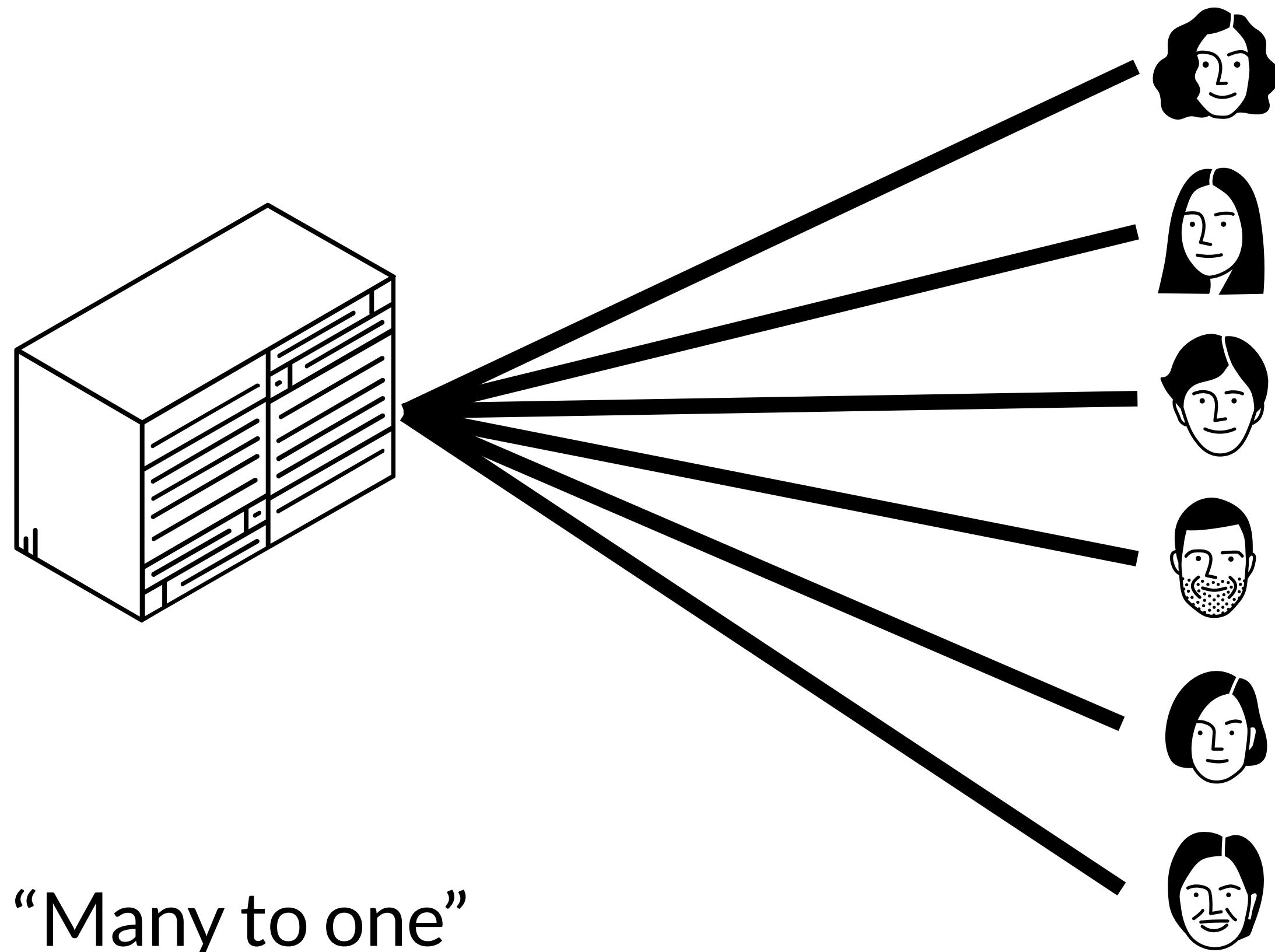


First wave: mainframe computing

- Batch processing
 - Write your program on punch cards
 - Wait your turn for the computer
 - Run program, hope it works
 - If it doesn't, you'll have to fix it and wait for your next turn
 - Efficient use of resources, but poor interactivity

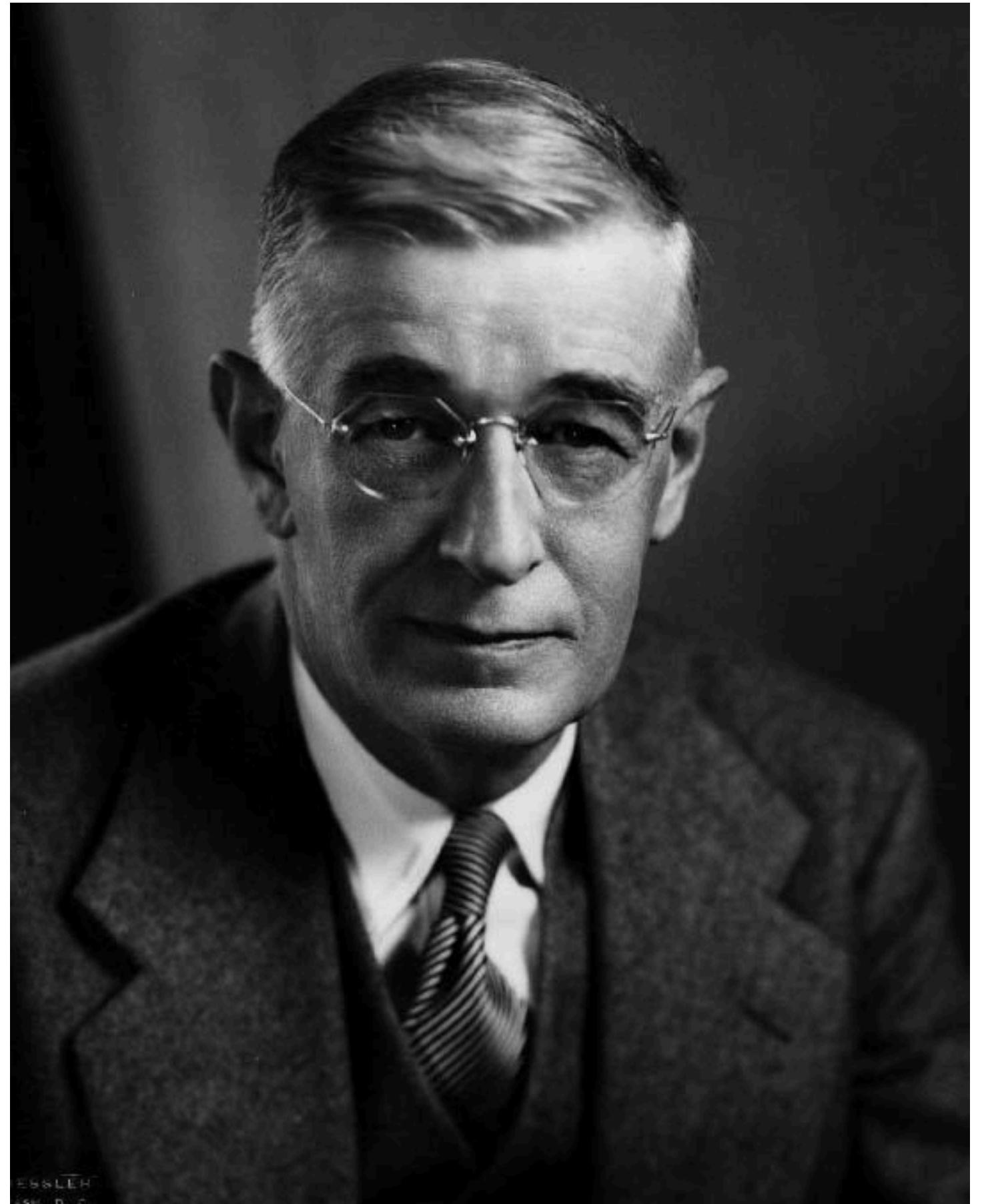


First wave: mainframe computing



Vannevar Bush

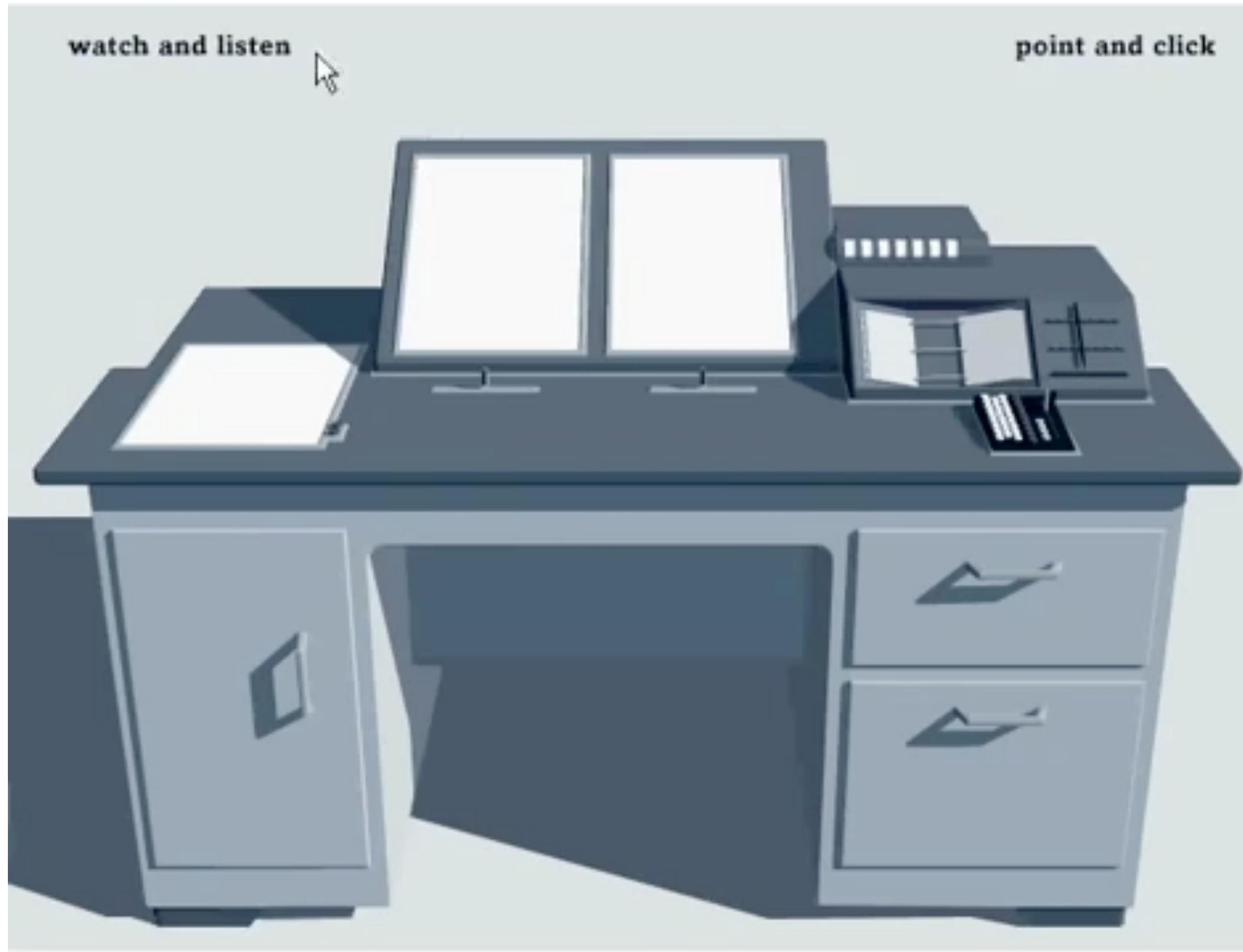
- Faculty at MIT
- Oversaw National Defense Research Committee, which led the Manhattan Project
- Post-war, helped define mission of the National Science Foundation
 - Federal government funds universities
 - Universities do basic scientific research
 - Research helps economy and defense



As We May Think

- Published in Atlantic Monthly, 1945
- [http://www.theatlantic.com/magazine/print/1945/07/as-we-may-think/
3881/](http://www.theatlantic.com/magazine/print/1945/07/as-we-may-think/3881/)
- In part, set out to define a post-war scientific research agenda
 - Speculative, not reflective

Memex (1945 speculative design)



<https://www.youtube.com/watch?v=c539cK58ees>

(video from 1995 animation presented at SIGIR, not from 1945)

Memex (1945 speculative design)

- Linking information across devices and sources
 - Hypertext, the foundation of the web
- Pen-based annotation of primary sources

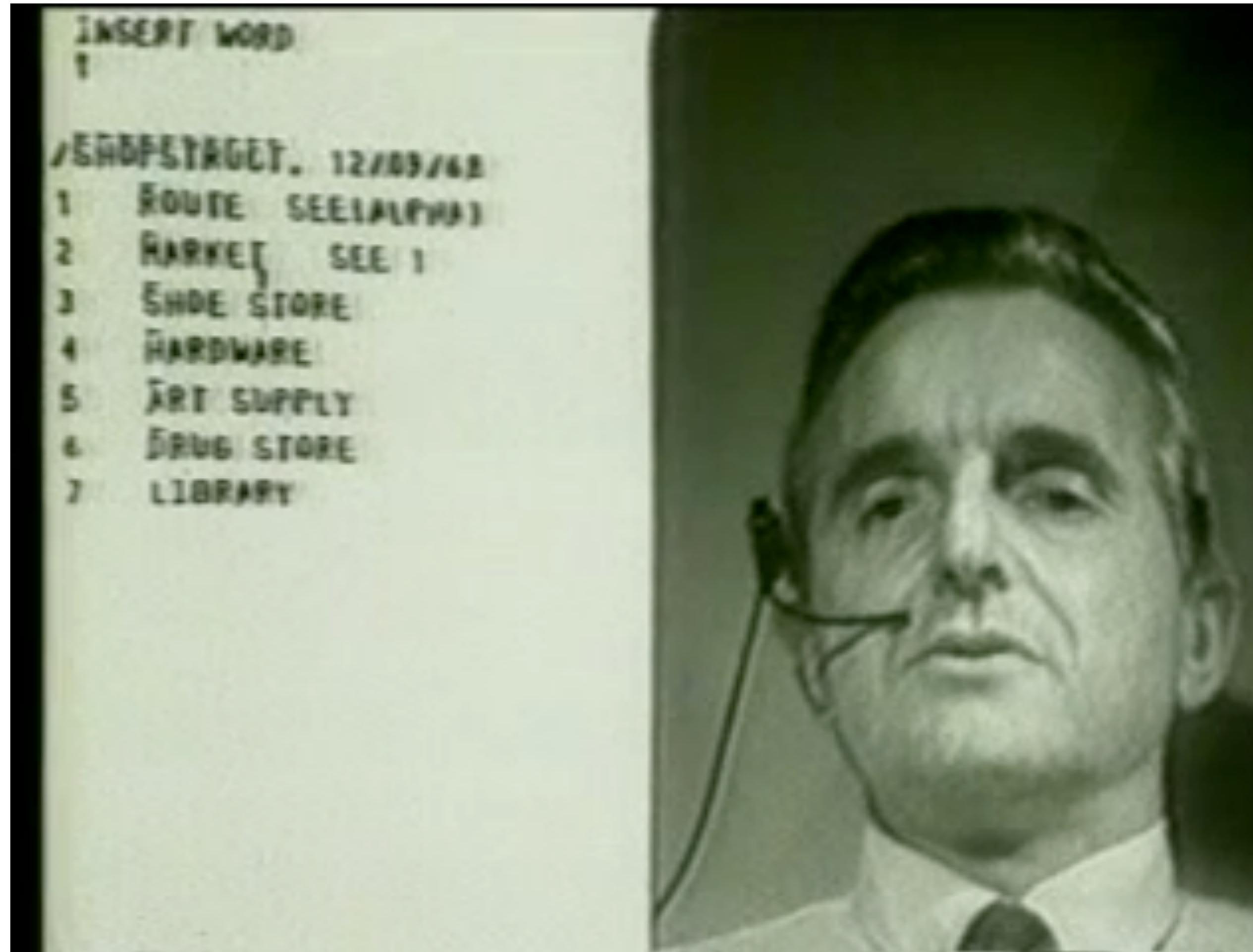


Command-Line interfaces (1960's)

- Originally used in a terminal connected to a mainframe
 - Was eventually integrated into personal computing (in Unix, etc.)
- A person could change execution based on output
- Enabled real-time debugging



Doug Engelbart's NLS (1968)



The image is a composite of two photographs. On the left, a screenshot of the NLS (Augment) interface is displayed. The screen shows a menu with the following options:

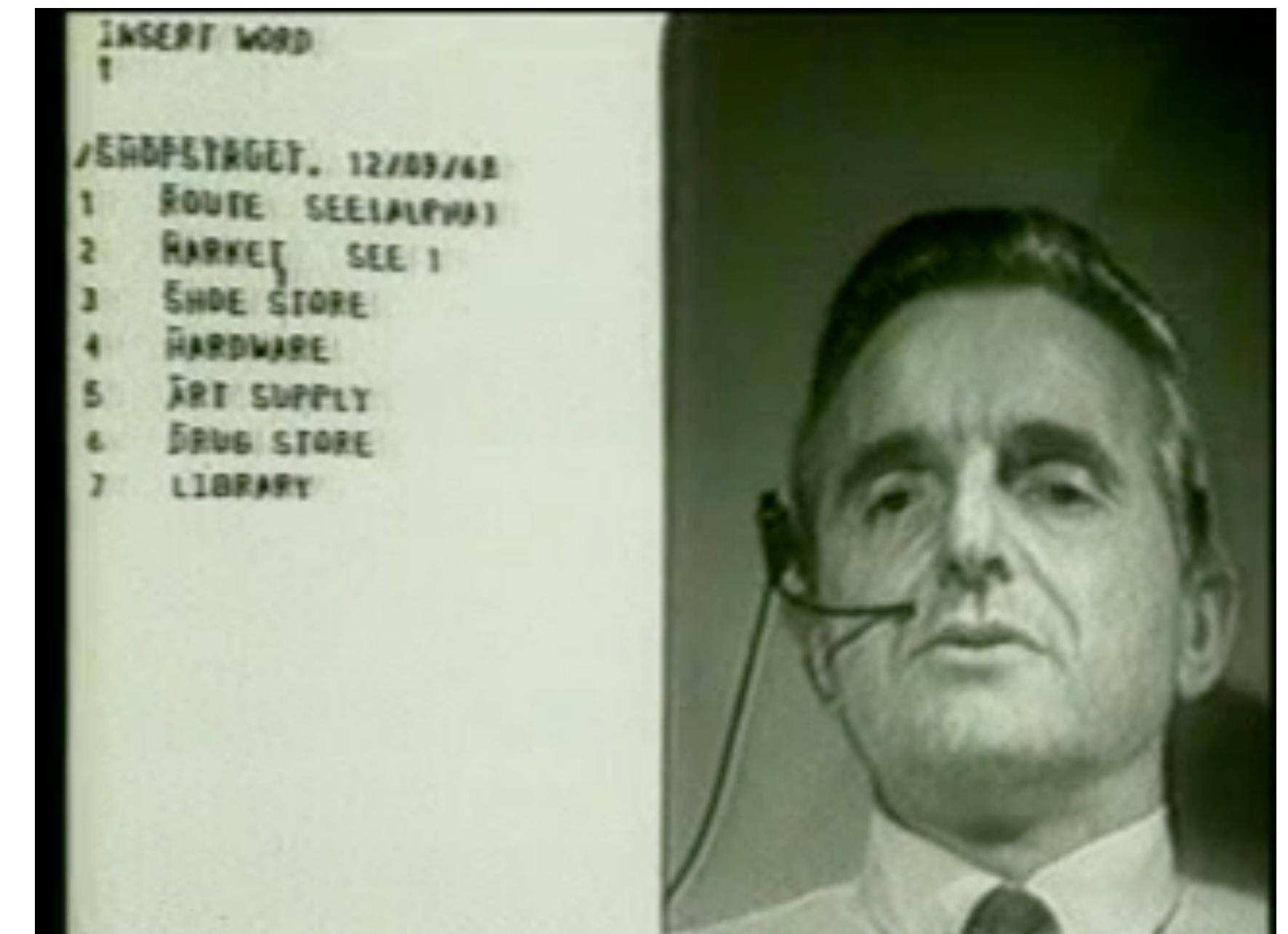
- 1 INSERT WORD
- 2
- 3 /SHOPSTREET. 12/03/68
- 4 ROUTE SEE ALPHABET
- 5 MARKER SEE 1
- 6 SHOE STORE
- 7 HARDWARE
- 8 ART SUPPLY
- 9 DRUG STORE
- 10 LIBRARY

On the right, a black and white portrait photograph of Doug Engelbart is shown. He is wearing a suit and tie, and has a microphone attached to his shirt.

<http://www.douengelbart.org/firsts/1968-demo-interactive.html>

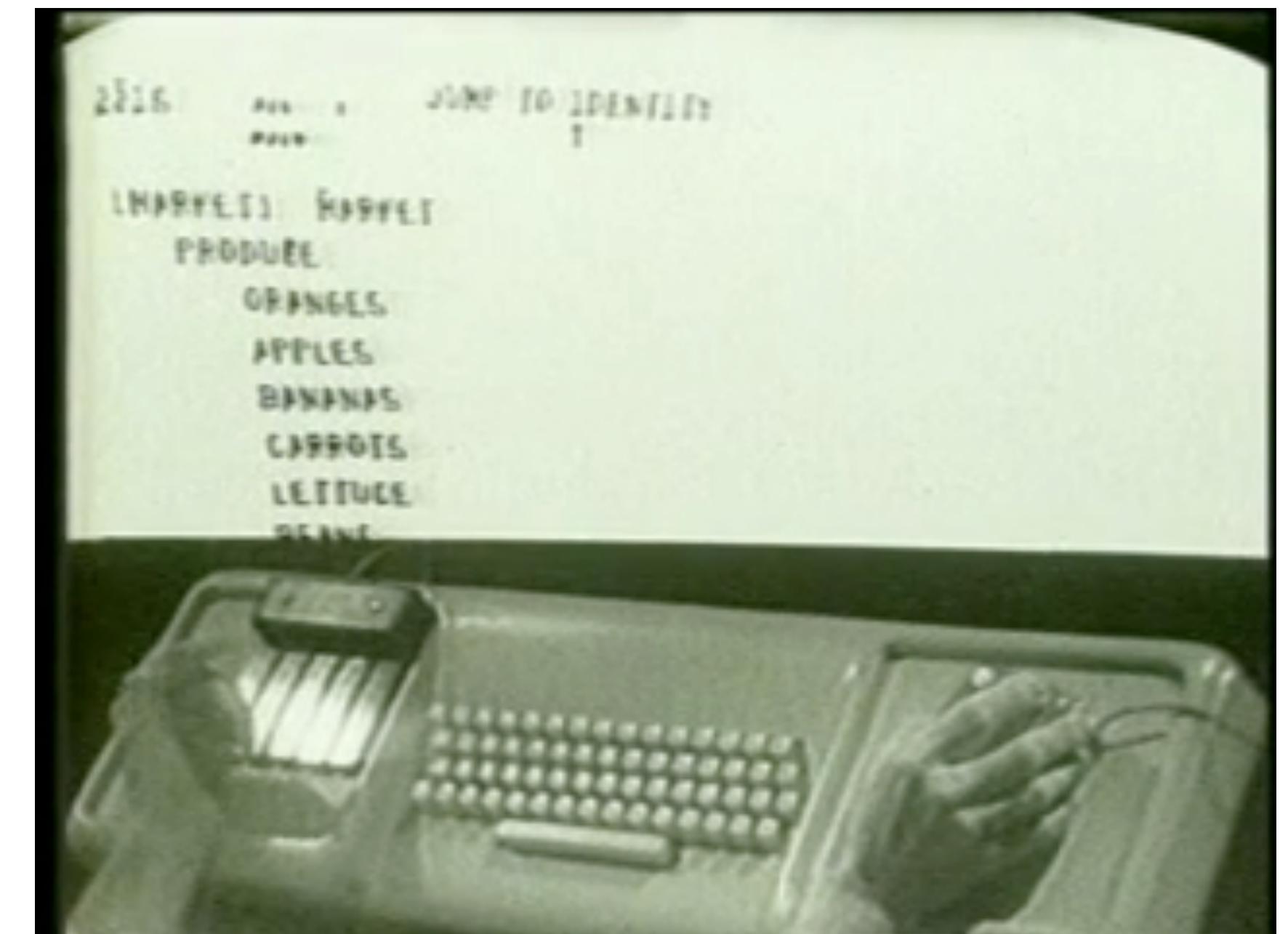
Doug Engelbart's NLS (1968)

- First working hypertext system
- Invention of the mouse
- Simple graphics
(earlier systems had this,
but used in a full system here)



Doug Engelbart's NLS (1968)

- It introduced other ideas as well
 - A chording keyboard
 - Remote collaboration
- Some people thought he “faked it”
- Others thought it was irrelevant because “the terminal can do the same”
- Won Turing Award in 1997



Three waves of computing



Mainframe
computing



Personal
computing



Ubiquitous
computing

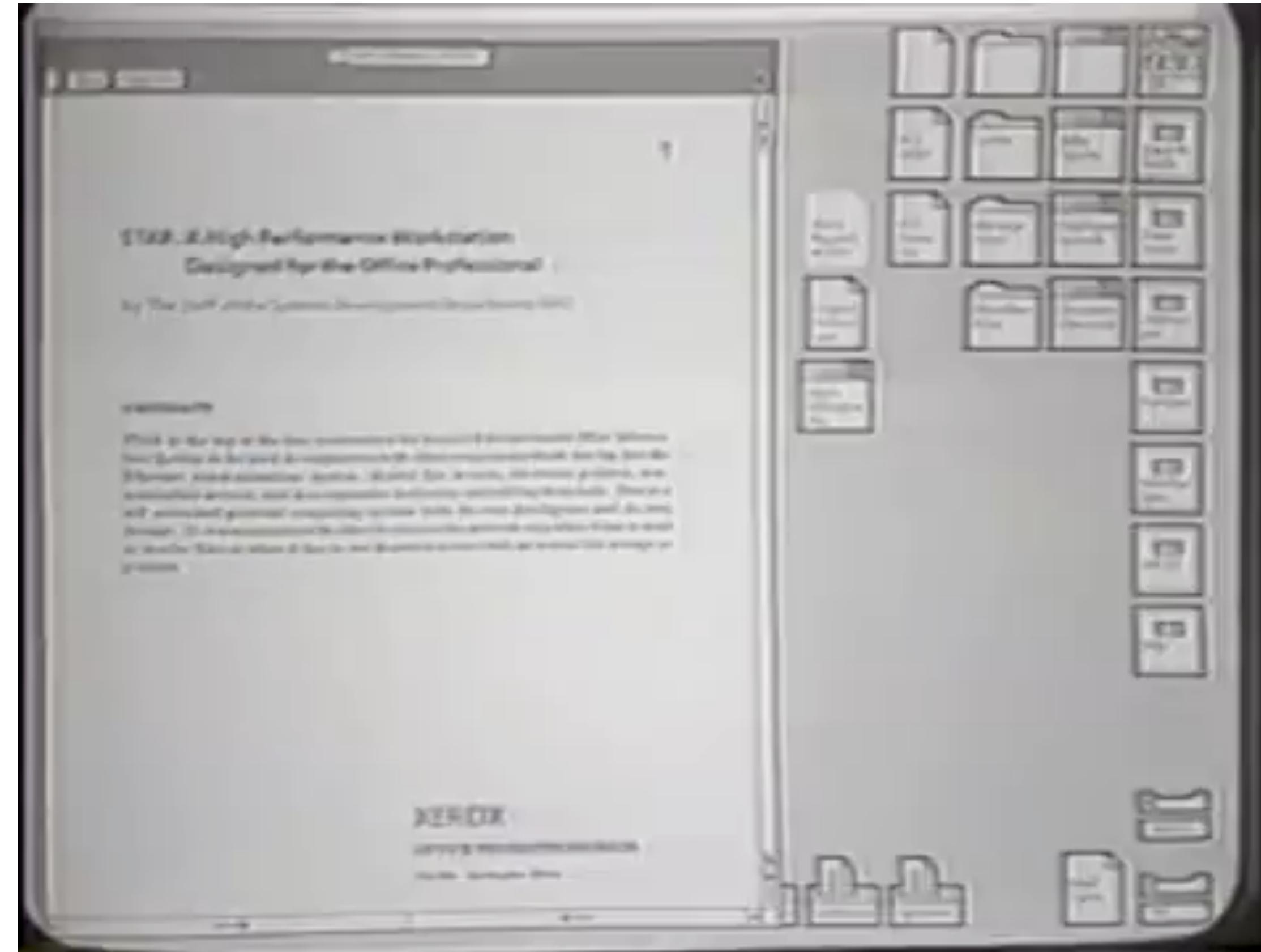
Second wave: personal computing

- First introduced by Xerox
- Xerox Alto, 1973
 - Mouse
 - Chording keyboard
- Xerox Star, 1981
- Xerox models
 - were commercially unsuccessful
- Still expensive, too few applications



Second wave: personal computing

Xerox Star (1981)

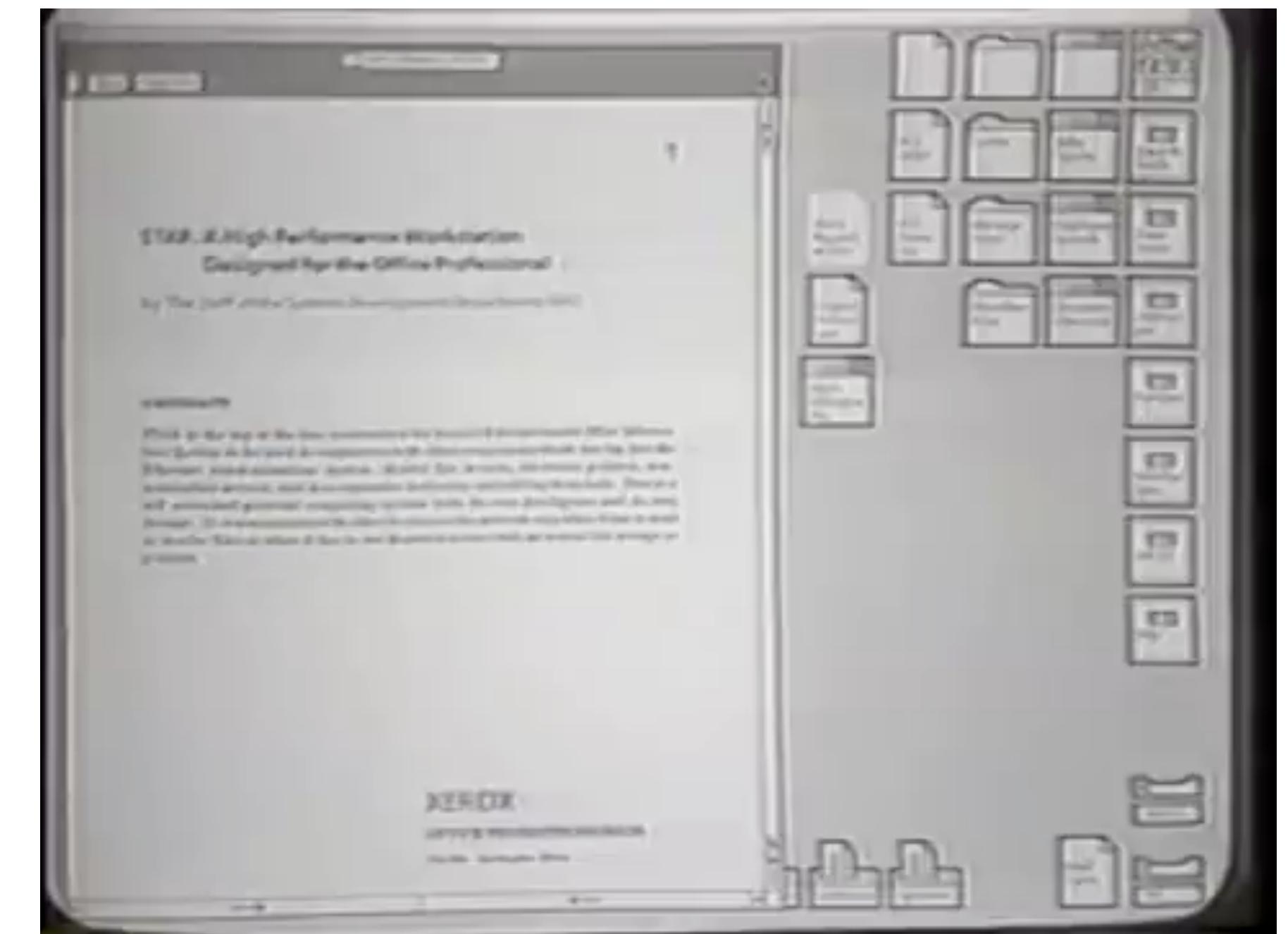


<https://www.youtube.com/watch?v=ODZBL80JPqw>

Second wave: personal computing

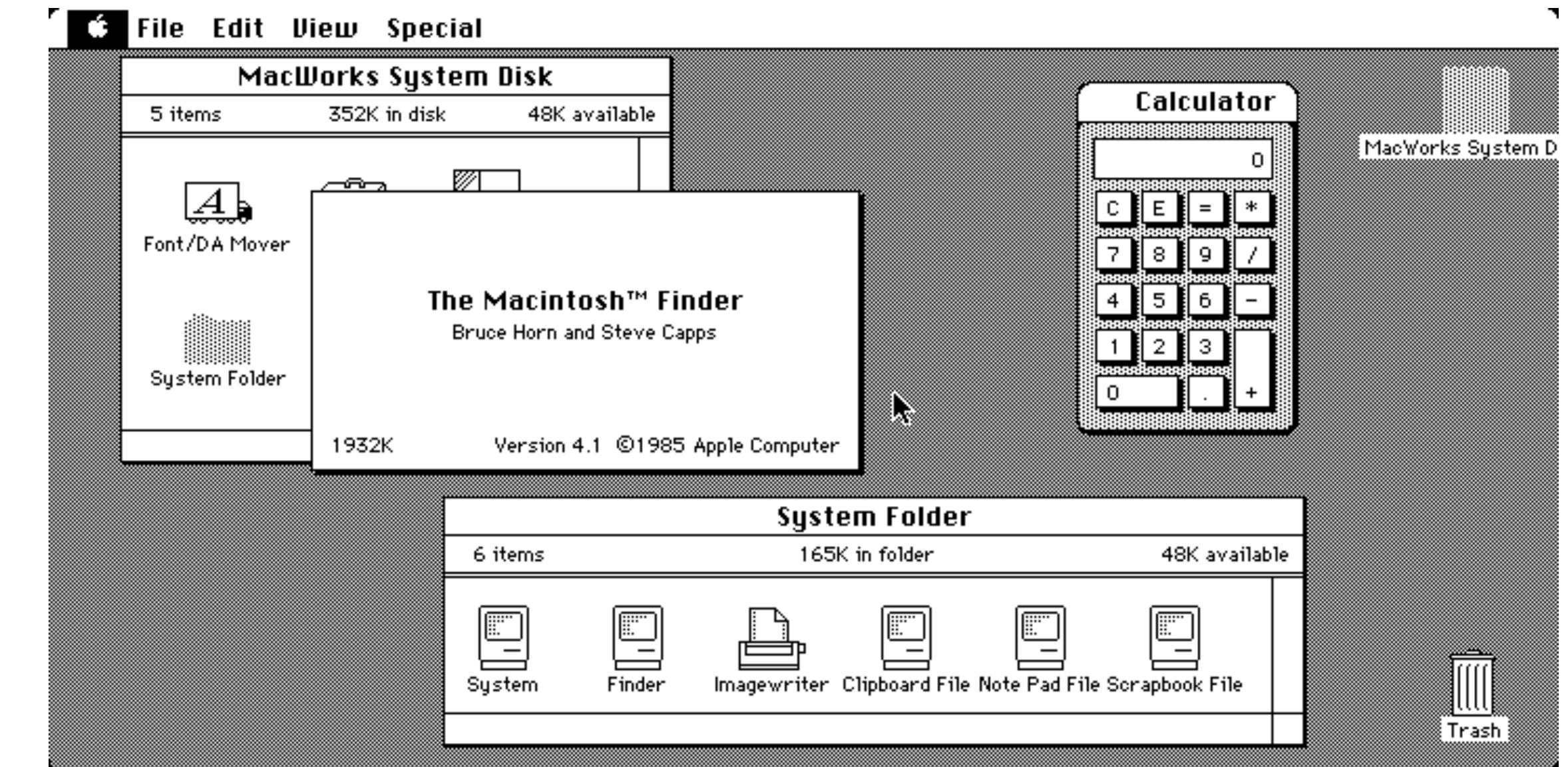
Xerox Star (1981)

- Software running in windows
- Desktop with icons for navigating between files and programs
- Super slow!



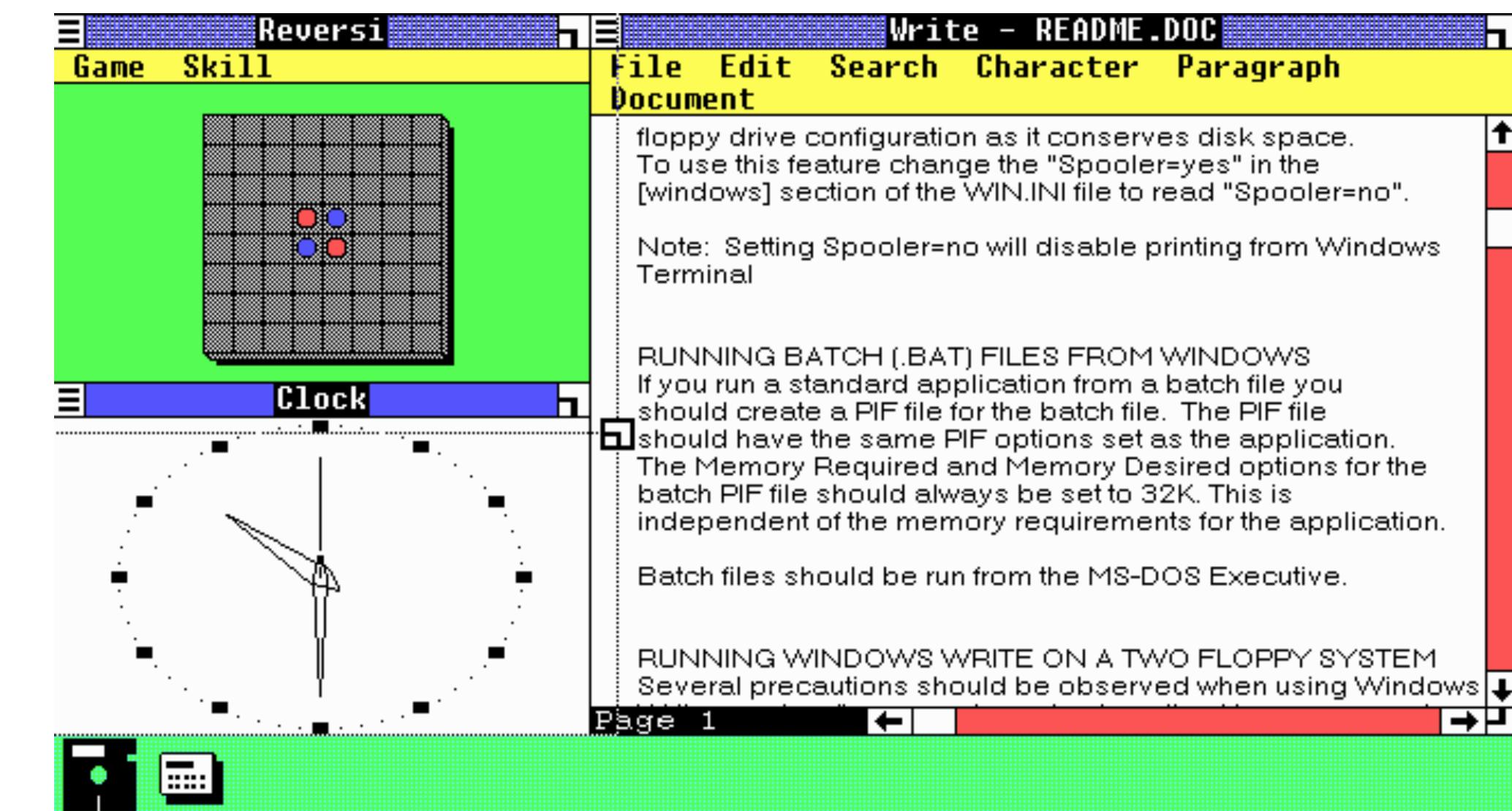
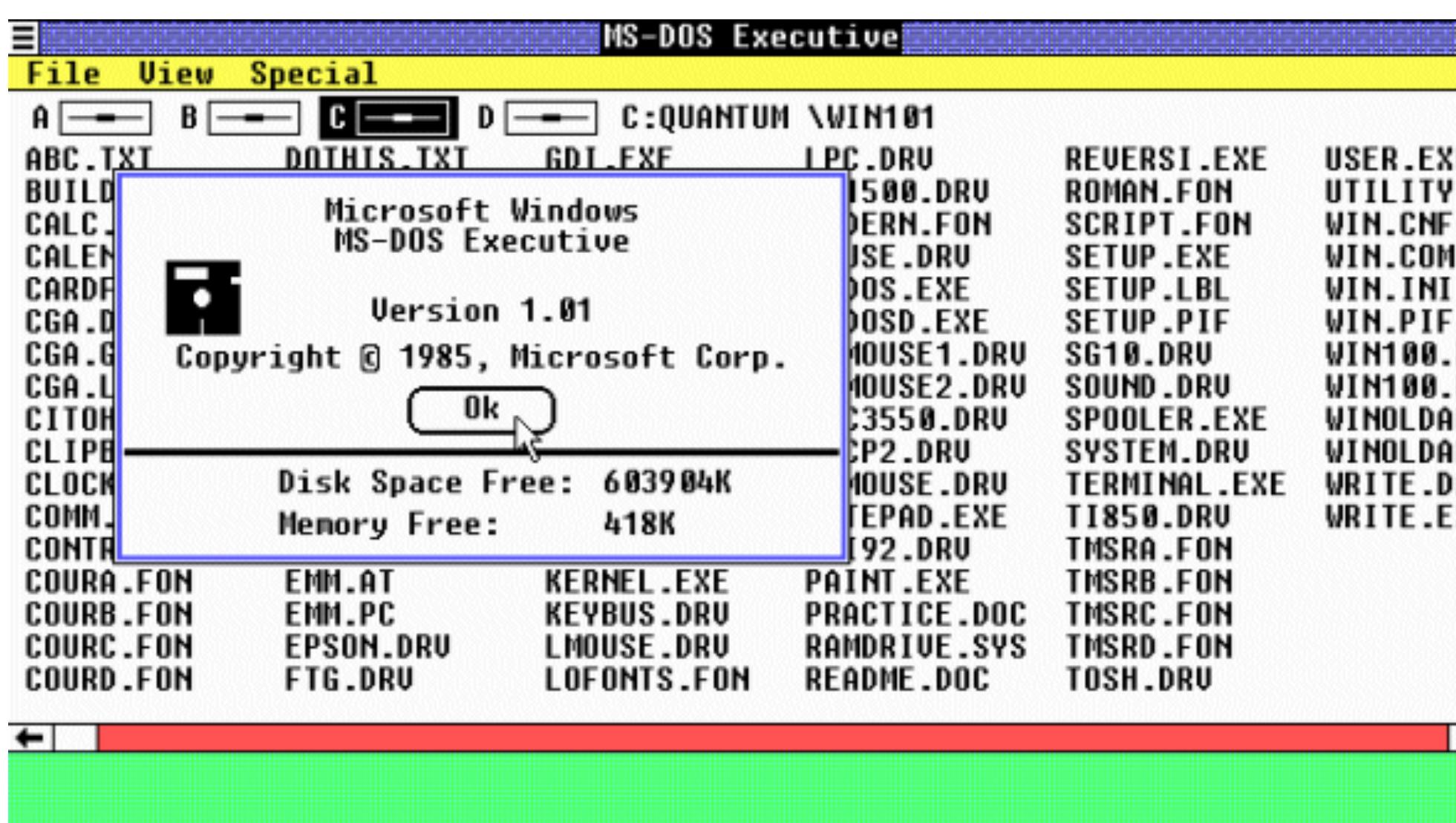
Second wave: personal computing

Macintosh (1984)



Second wave: personal computing

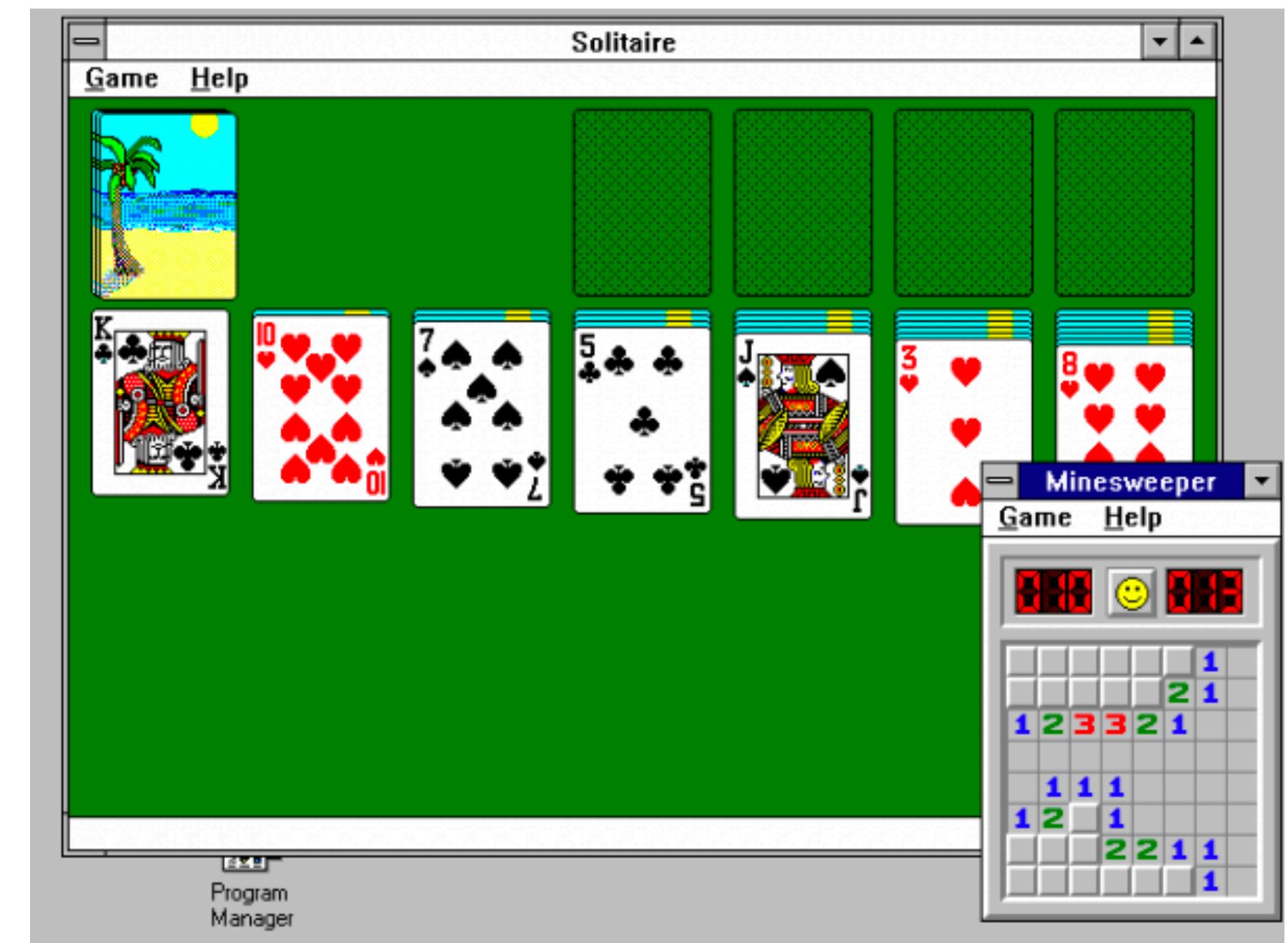
Windows 1.0 (1985)



Second wave: personal computing

Windows 3.0 & 3.1 (1990 & 1992)

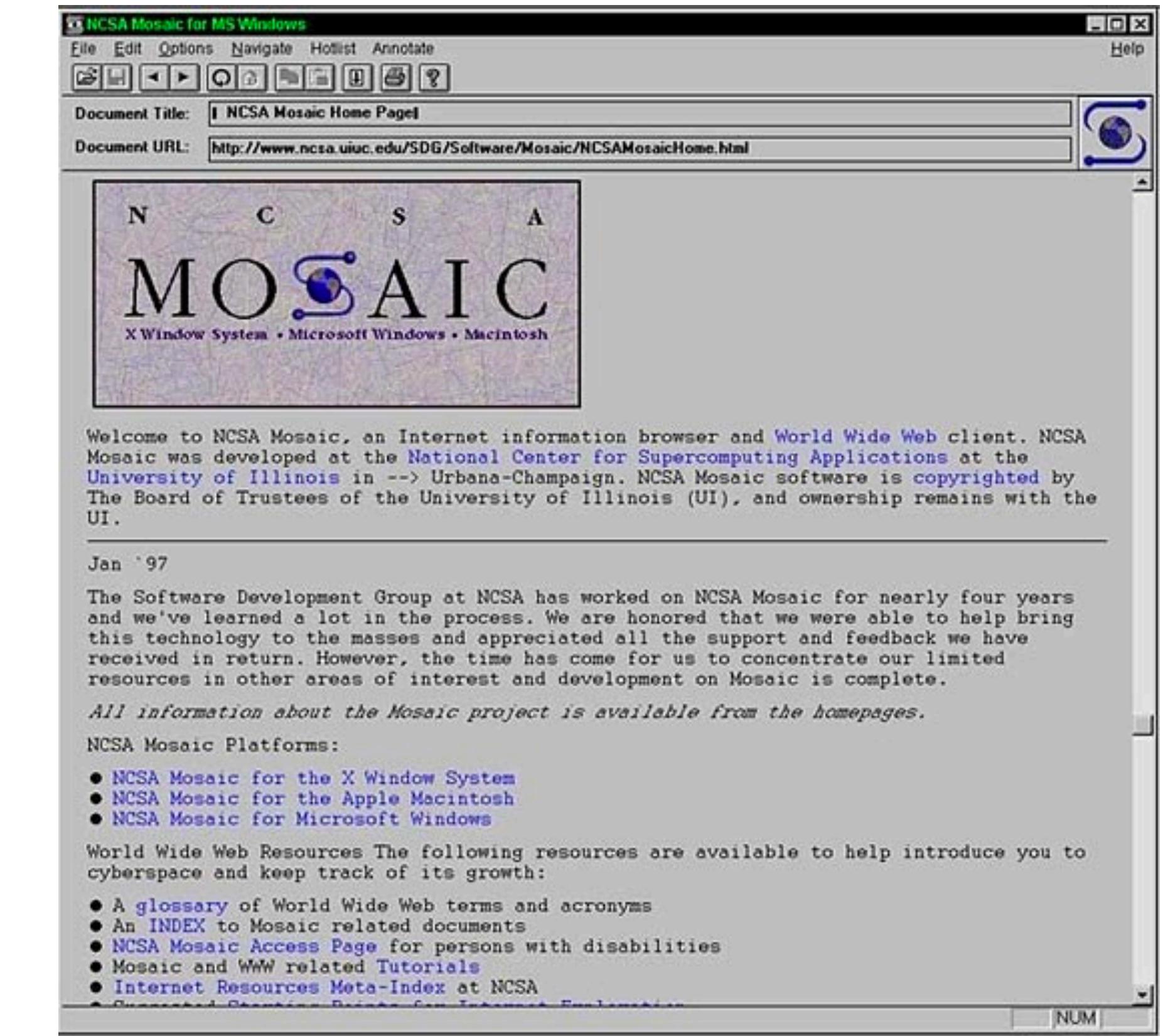
- Windowing became primary
- Added games: Solitaire, Minesweeper, and FreeCell!
 - These were a trick to teach mouse skills



Second wave: personal computing

Mosaic Web Browser (1993)

- Originally for Unix systems, later ported to Mac and Windows
- “First” graphical web browser
- Microsoft IE came in 1995
- Apple didn’t make a browser until Safari in 2003



Second wave: personal computing



“One to one”

Three waves of computing



Mainframe
computing



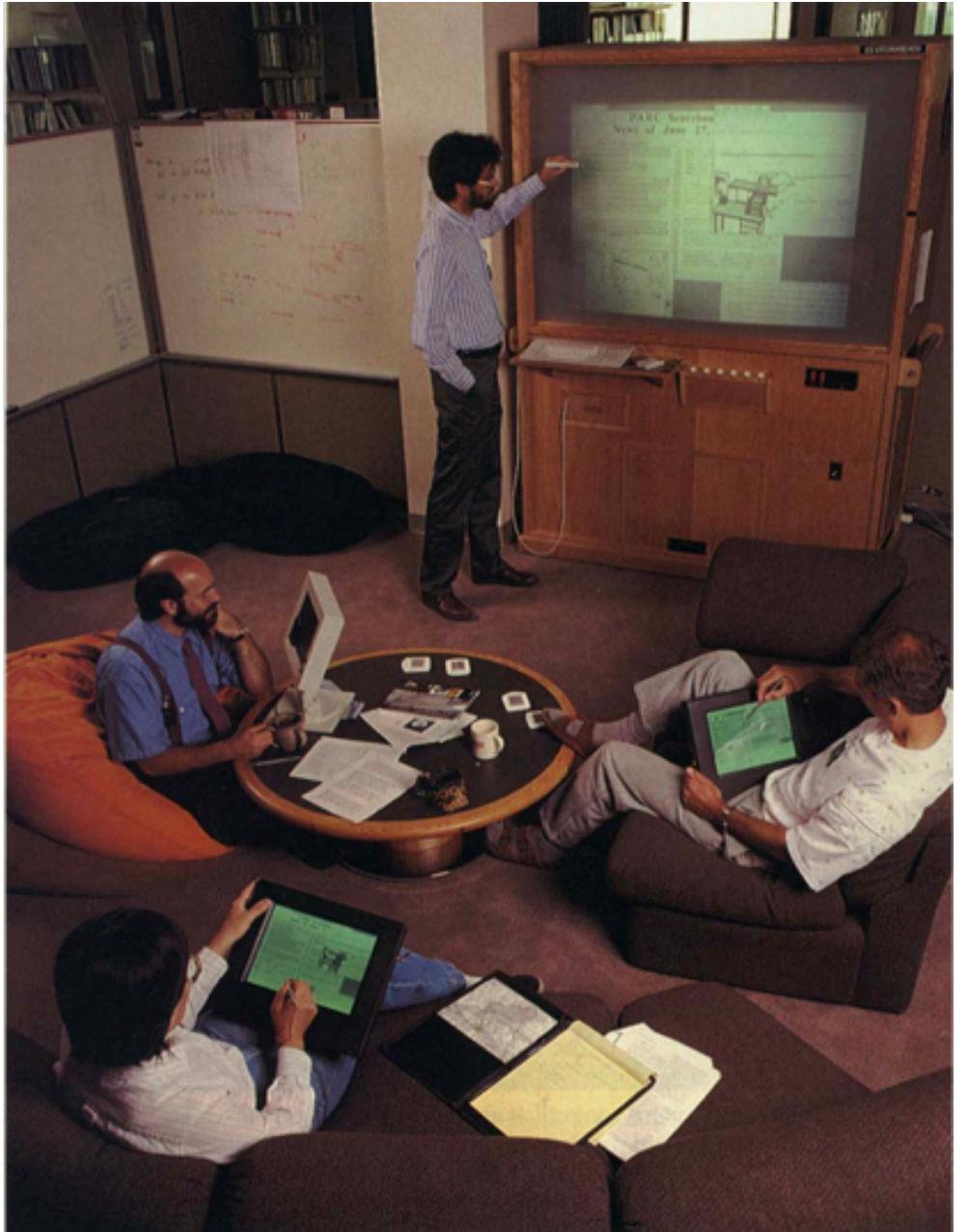
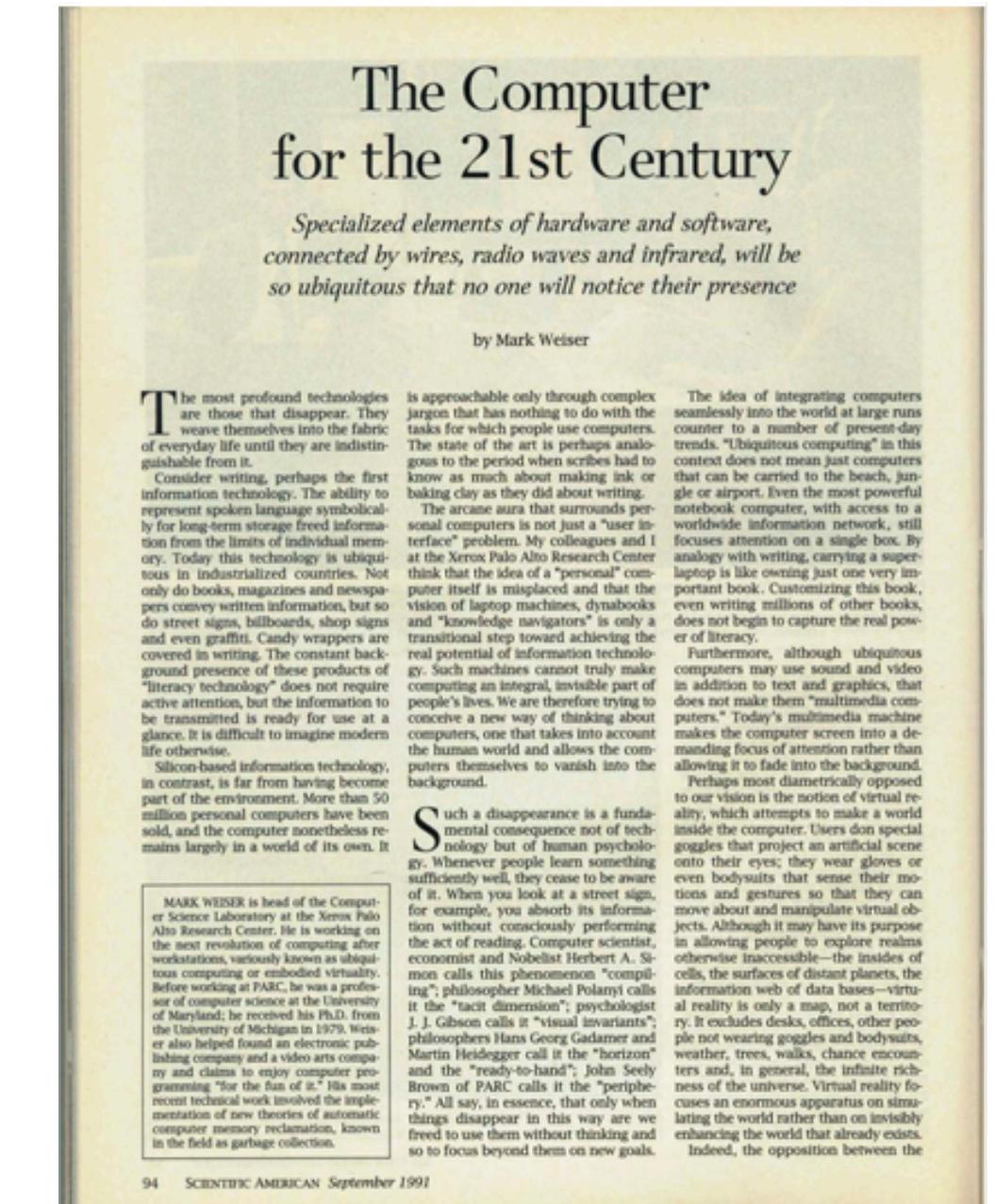
Personal
computing



Ubiquitous
computing

Third wave: ubiquitous computing

- Weiser speculated people would interact with three types of computers
 - Tabs: inch-scale devices, like post-its
 - Pads: foot-scale devices, like paper
 - Boards: yard-scale devices, like whiteboards
- Speculated devices would have shared ownership



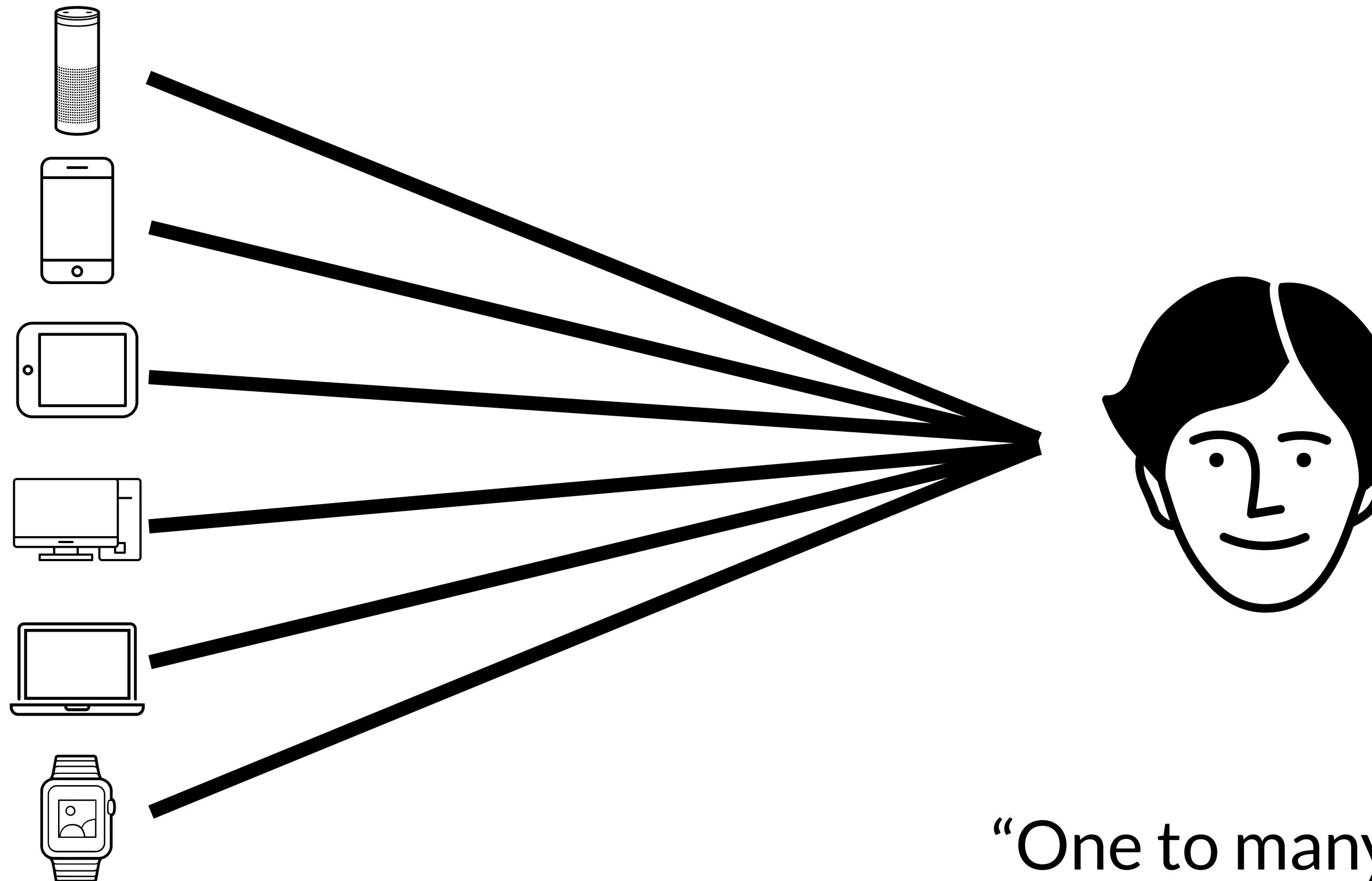
Third wave: ubiquitous computing



Third wave: ubiquitous computing

- Lines up with what we use today, for the most part
 - Tabs = phones and watches
 - Pads = tablets and laptops
 - Boards = interactive projectors? smart TVs? augmented reality?
- Still a strong sense of device ownership

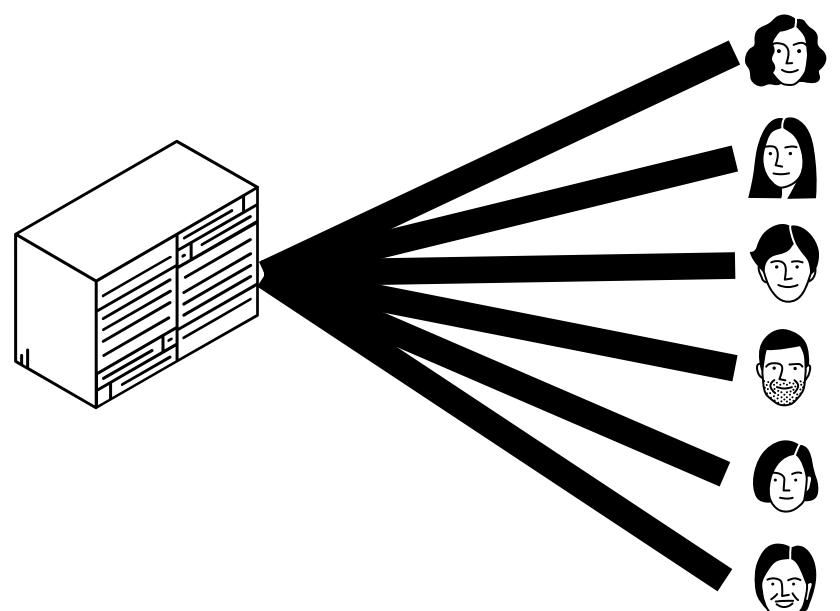
Third wave: ubiquitous computing



Three waves of computing



Mainframe
computing



“Many to one”



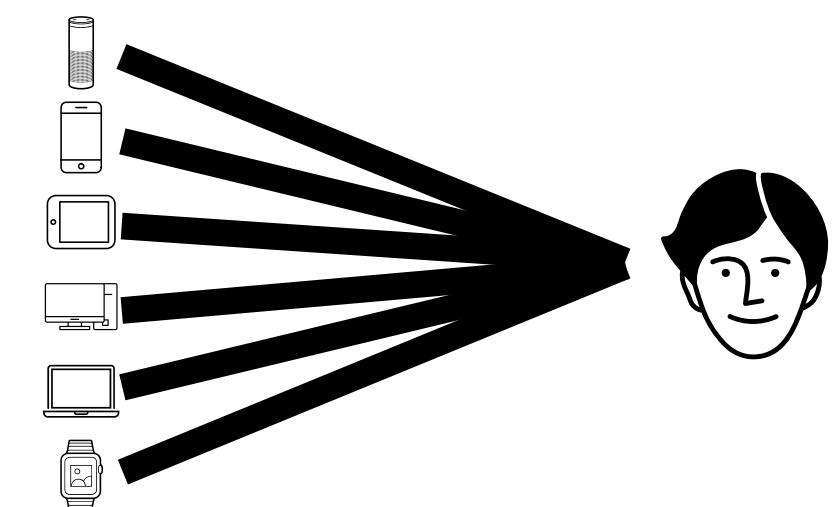
Personal
computing



“One to one”



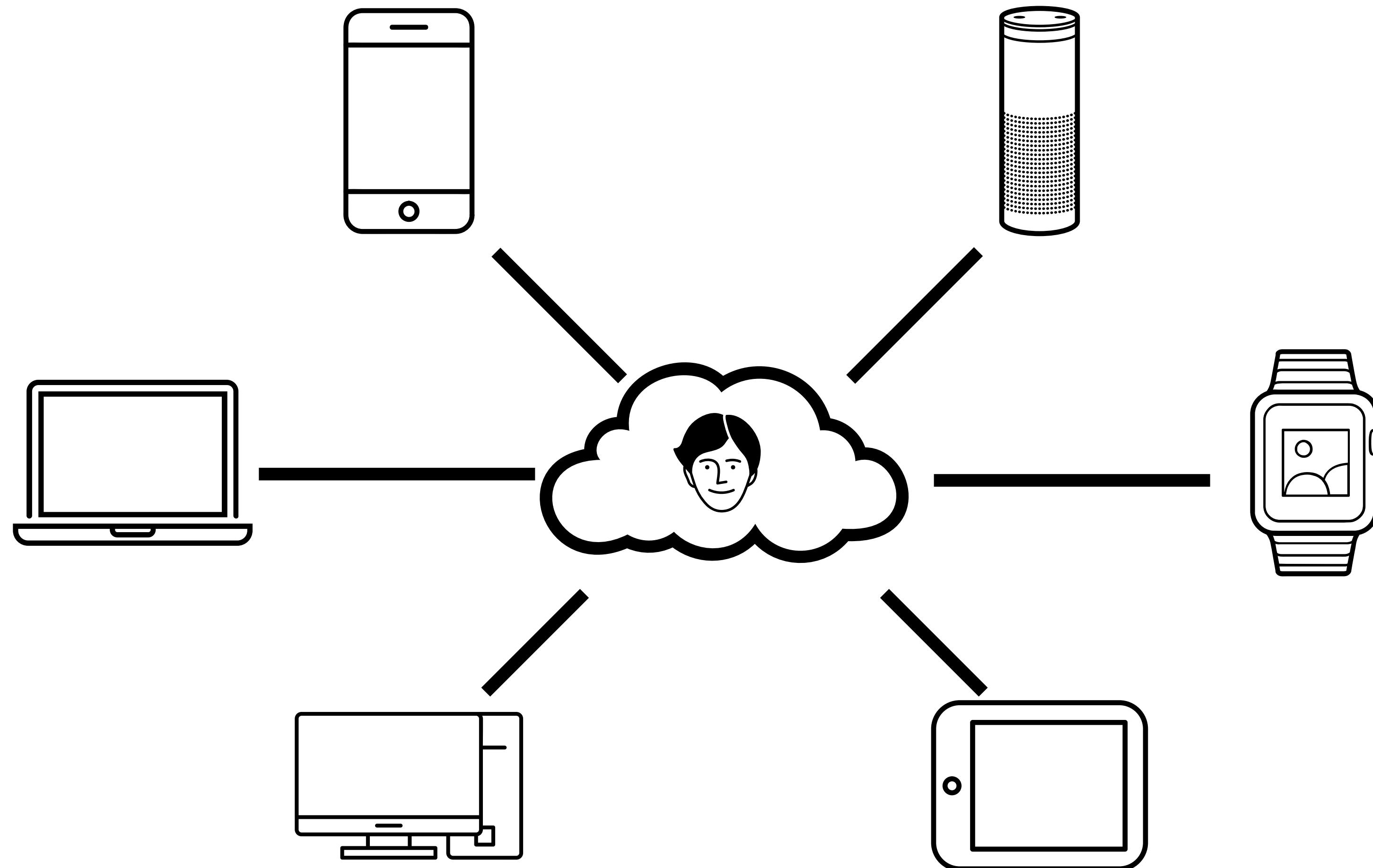
Ubiquitous
computing



“One to many”

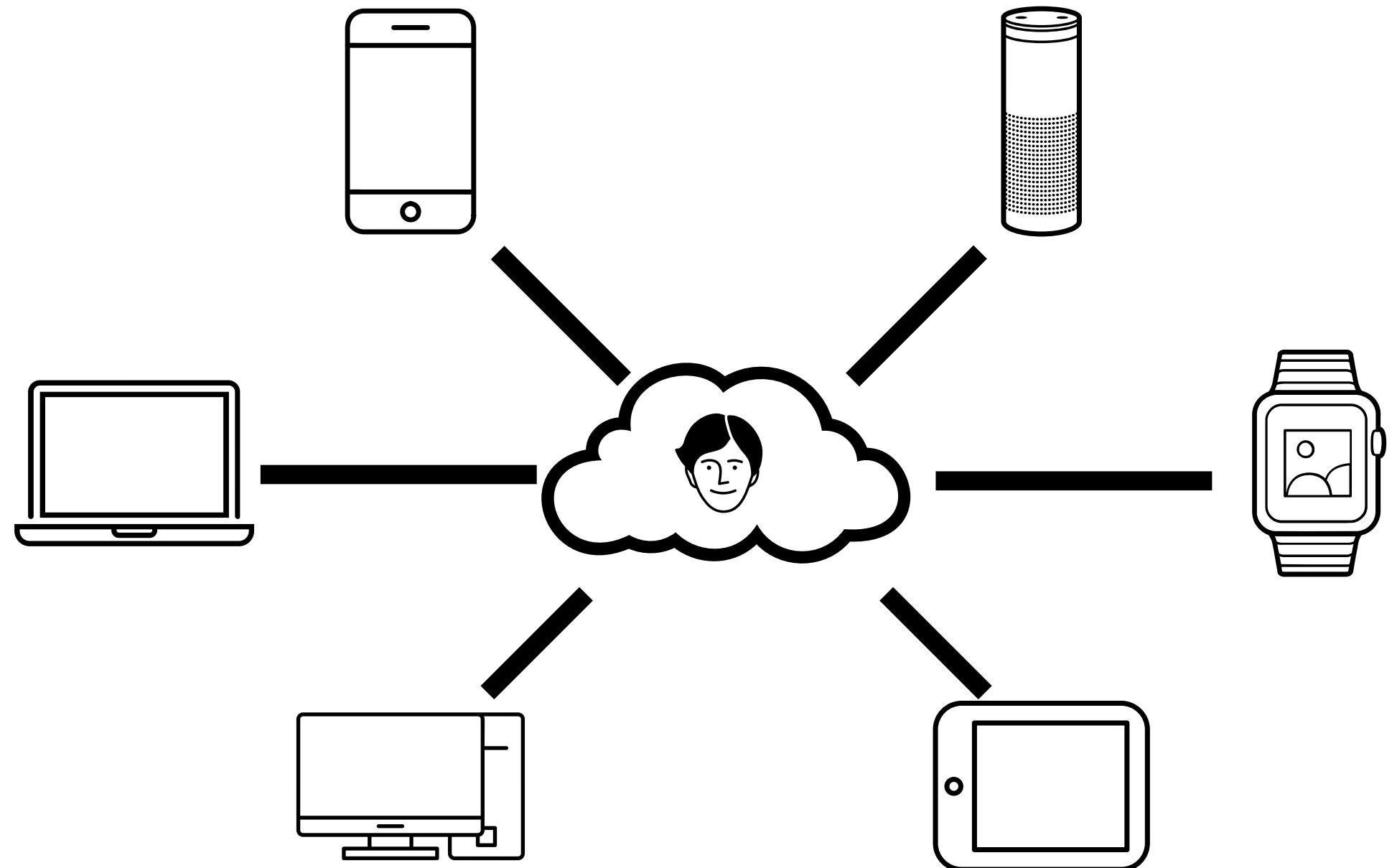
Think & Share:
**Why are web tools now the standard
for interface development?**

One person, many devices, synced over the cloud



One person, many devices, synced over the cloud

- Use HTTP requests to send data to the cloud and receive data from it
 - JavaScript provided early tools to do this
- Render that data with HTML
- Style it with CSS



**Ubiquitous computing is, in large part,
why web tools are the current standard
for interface development**

Web tools as the standard

- Nearly every platform needs to communicate with a cloud system
- Most need a web browser so people can access sites
- Shared programming language and development environment enables efficient work
- Developers can write once, deploy to many platforms
 - Hopefully customize style and functionality to the device
- Other reasons?

Today's goals

By the end of today, you should be able to...

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and policies
- Describe upcoming course tasks

Course Overview

- Course staff introductions
- Administravia
- Topics covered
- A0 (due Friday)

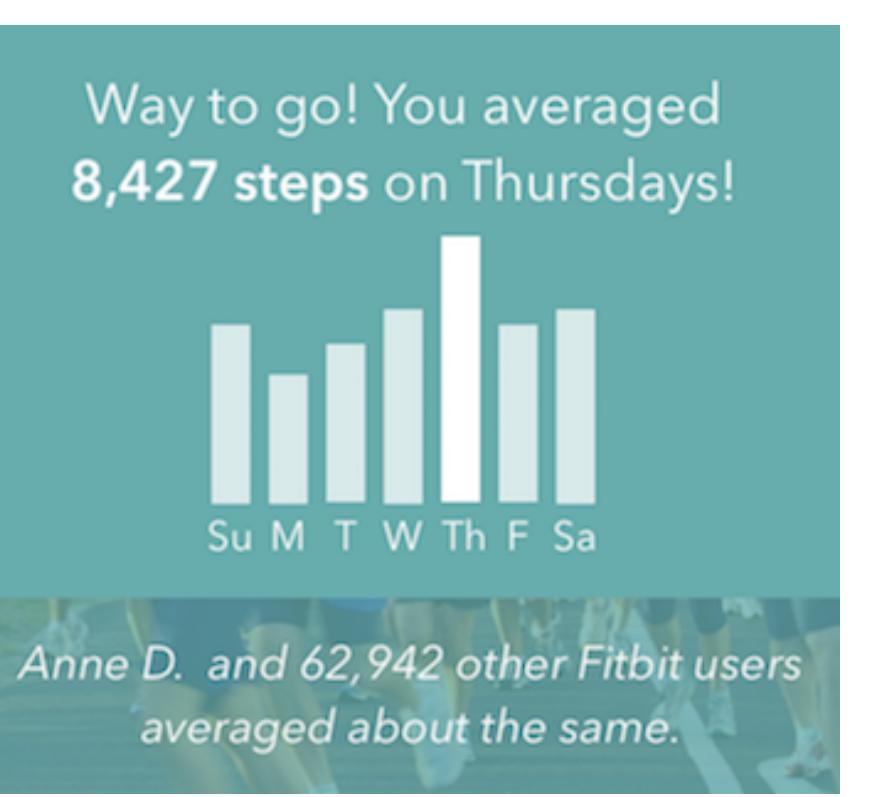
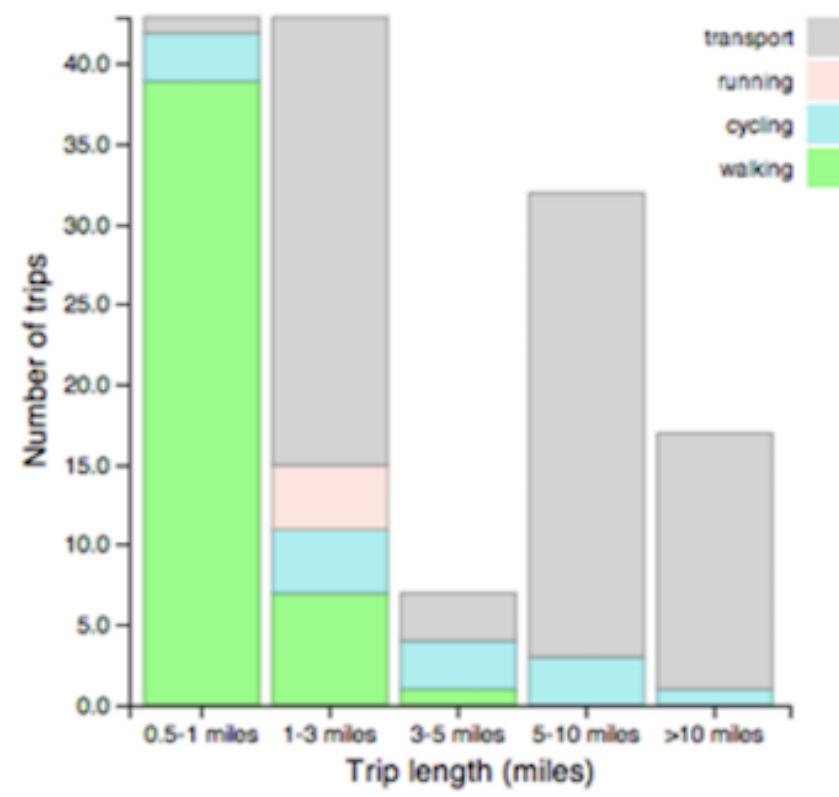
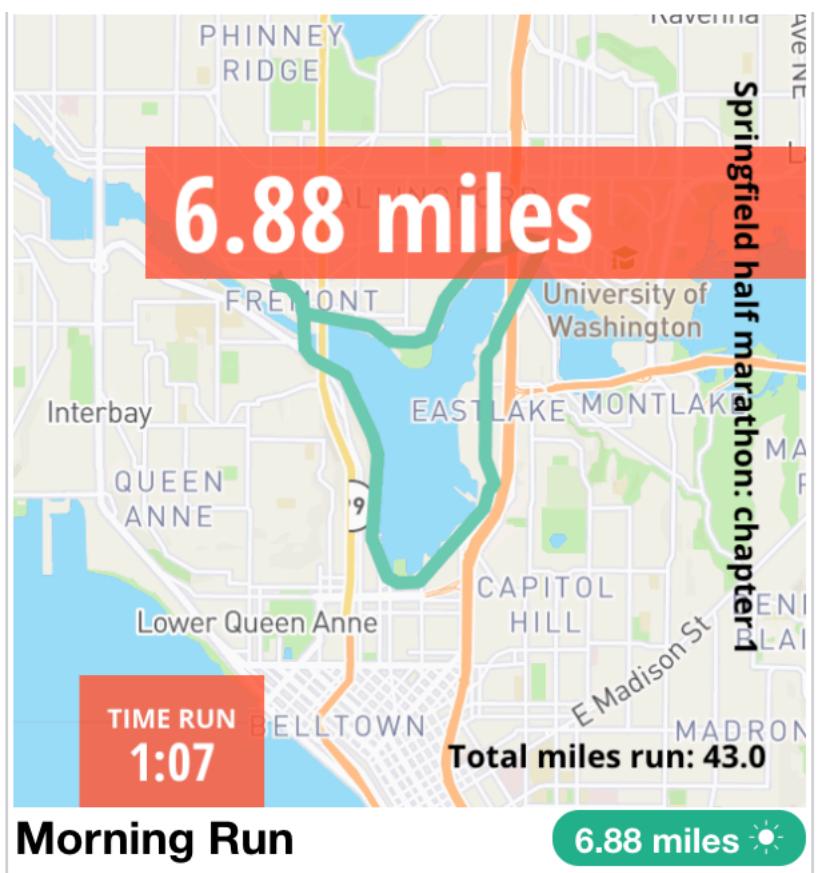
Who we are

Professor Daniel A. Epstein (he/his/him)

- Ph.D. Computer Science & Engineering,
University of Washington 2018
- B.S. Computer Science,
University of Virginia 2012
- Joined UCI Informatics in 2018
- Internships at Microsoft & Adobe,
collaborations with Snap(chat)



Who we are



Who we are

Goda Addanki (she/her/hers)

- 1st year MS in Computer Science
- Worked at Qualcomm as an embedded protocol developer for 5 years
- Interested in networks and systems
- Likes dancing and reading non-fiction



Who we are

Seolha Lee (she/her/hers)

- 1st year PhD in Informatics
- M.S. City and Regional Planning,
Georgia Tech & B.S. Civil
Engineering, Seoul National
University (Korea)
- Researches data-driven urban
governance & social justice
- Swimmer & plant lover



The syllabus

- Explains due dates/times, assignment policies, exam goals, etc.
- It probably answers your question
 - Please check it before you ask us

Syllabus

Policies and background for IN4MATX 133, Winter 2022 Quarter. All syllabus content is subject to change, particularly prior to the start of the quarter. In particular, if there are phrasings which do not make sense for the virtual nature of the quarter, please let Professor Epstein know and he will correct or clarify.

Course Description

From the [catalog](#):

Introduction to human-computer interaction programming. Emphasis on current tools, standards, methodologies for implementing effective interaction designs. Widget toolkits, Web interface programming, geo-spatial and map interfaces, mobile phone interfaces.

In practice, every instructor takes a slightly different bend to these topics. I focus on covering modern technologies for web and mobile development and how device capabilities can impact interface design and software architecture.

All students must have taken [I&C SCI 45J](#) or [I&C SCI 45C](#) with a C or higher. This course will also assume students have mastered material in prior courses.

Learning Objectives

At the end of this course, students should be able to:

- Program web applications in HTML, CSS, and JavaScript which are sensitive to screen size and a person's abilities
- Leverage external data sources and APIs via asynchronous HTTP requests
- Develop hybrid mobile applications which take advantage of on-device utilities
- Describe affordances of different screen modalities and input techniques
- Choose an appropriate web or mobile development framework for a given task
- Implement and articulate best practices for authentication, storage, and communication in web and mobile

Course Format

- Fully remote for the first two weeks
 - We hope to offer lectures and discussions in-person starting week 3, but who knows
 - Synchronous delivery, but recorded
- All office hours will be online to avoid close contact in smaller spaces
 - May offer in-person hours later in the quarter, but there will always be an online option

Staying in touch

- Web: <http://inf133-wi22.depstein.net/>
- Email us: informatics-133-staff@uci.edu
- Slack: <https://uci-inf-133-wi22.slack.com/>
 - Information will go out to Slack first!
 - For the most part, Canvas will only be used for submission and grades

Staying in touch

- Office hours: on calendar
 - Tuesdays (Epstein), TAs TBD
- Submission: clone starter code from GitHub Classroom
 - Zip up finished assignment, submit on Canvas
 - Unsure about Git or GitHub? Go to discussion on Monday!
- Zoom: Automatically recorded, theoretically...

Communication best practices

- Slack is best for assignment/exam clarification and assistance
 - Please use the public channels to allow your peers to help
 - We will not reply to direct messages; there are too many of you!
- Email is best for personal communication (absences, concerns, regrades)
 - Email the staff list (informatics-133-staff@uci.edu) rather than us individually

Course goals

By the end of this course, you should be able to...

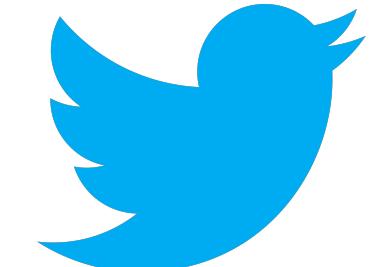
- Build webpages in HTML, CSS, and JavaScript which are sensitive to screen size and a person's abilities
- Build mobile apps in TypeScript frameworks, following design best practices
- Leverage external web APIs (databases, information sources) and device resources (photos, sensors) to lower development burden and enable new capabilities
- For a given design, choose appropriate devices to support and development frameworks to use

Assignments

- A1: Personal web portfolio



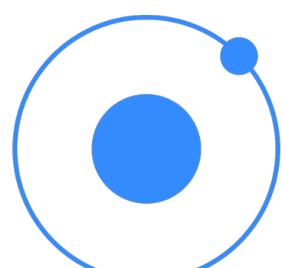
- A2: Programming on the web



- A3: Web frameworks

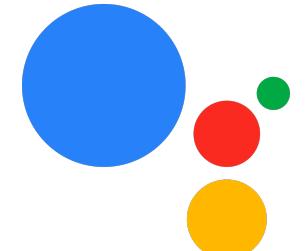
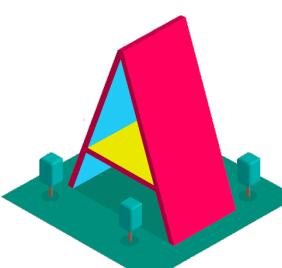


- A4: Mobile development



ionic

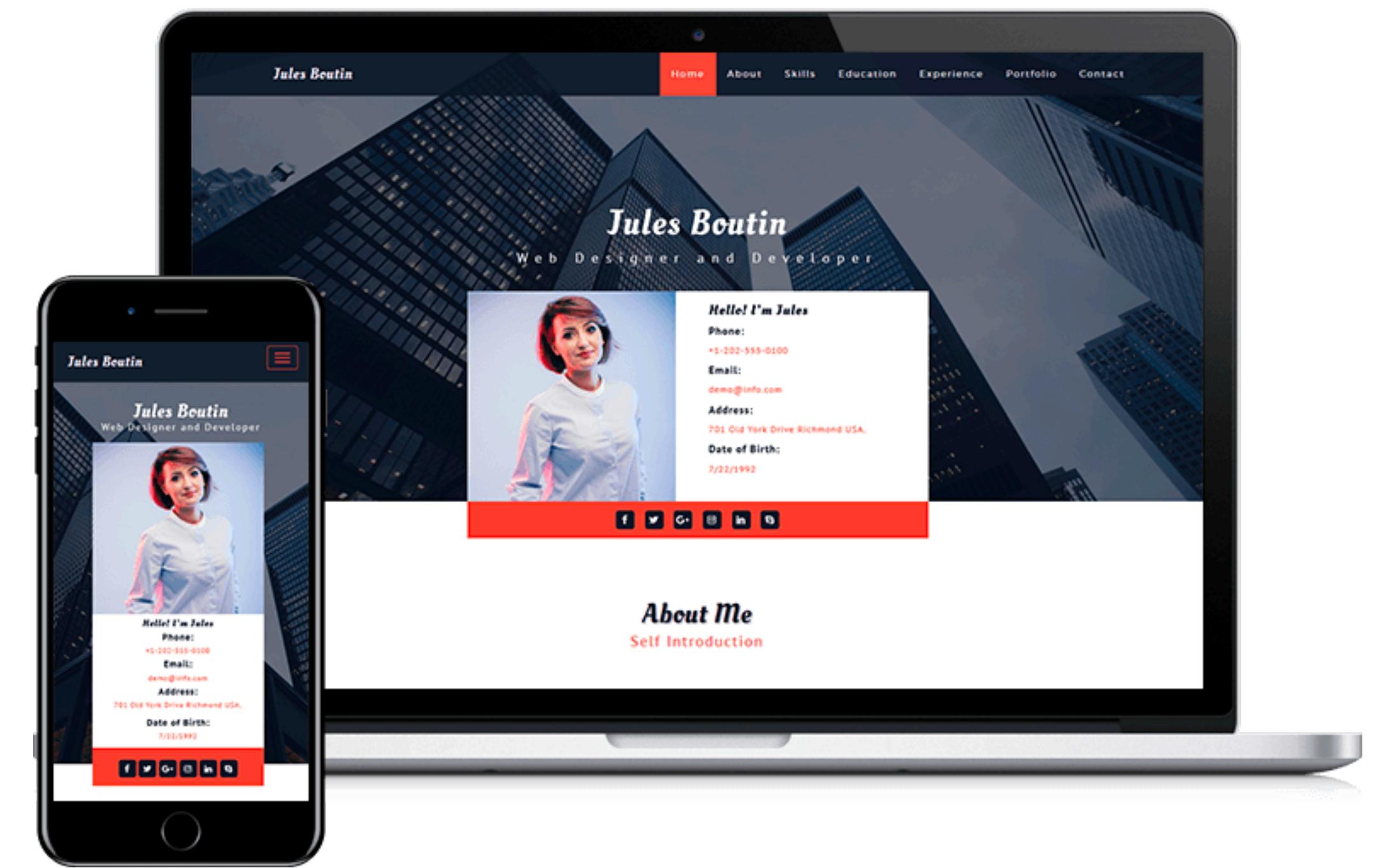
- A5: Beyond Web & Mobile



A1

Responsive Portfolio in HTML and CSS

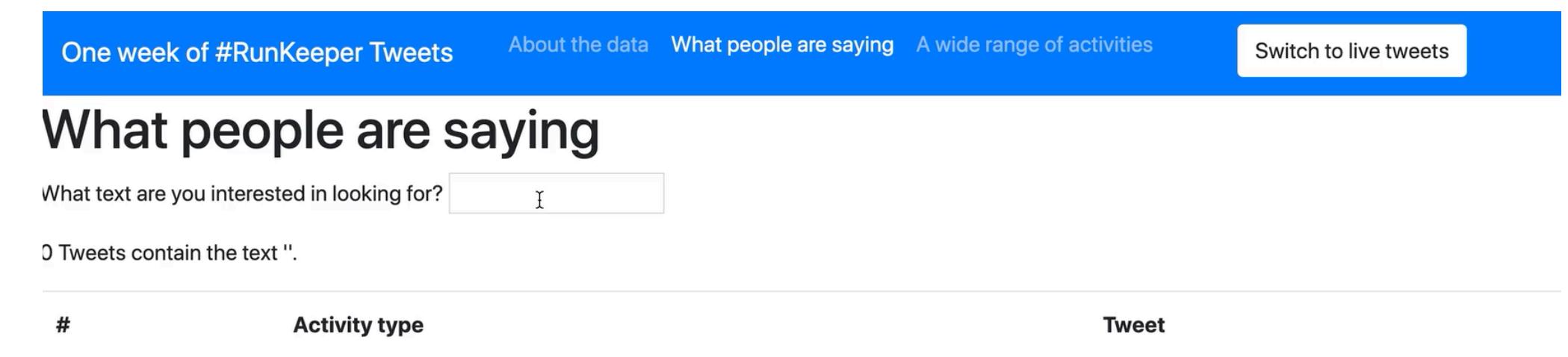
- Learning goal: develop familiarity with HTML and CSS, which form the foundation of all web design
- Apply *responsive* design, or adapt to screen size and orientation



A2

Runkeeper Tweet Report in JavaScript and TypeScript

- Learning goal: become comfortable with JavaScript, a widely-used development language on the web
- Will learn to use JavaScript libraries for visualization and interaction



A3

Spotify Browser in Angular

- Learning goal: develop skills in web frameworks which separate interface from data and interaction (Model-View-Controller)
- Will make an interactive browser of Spotify's library

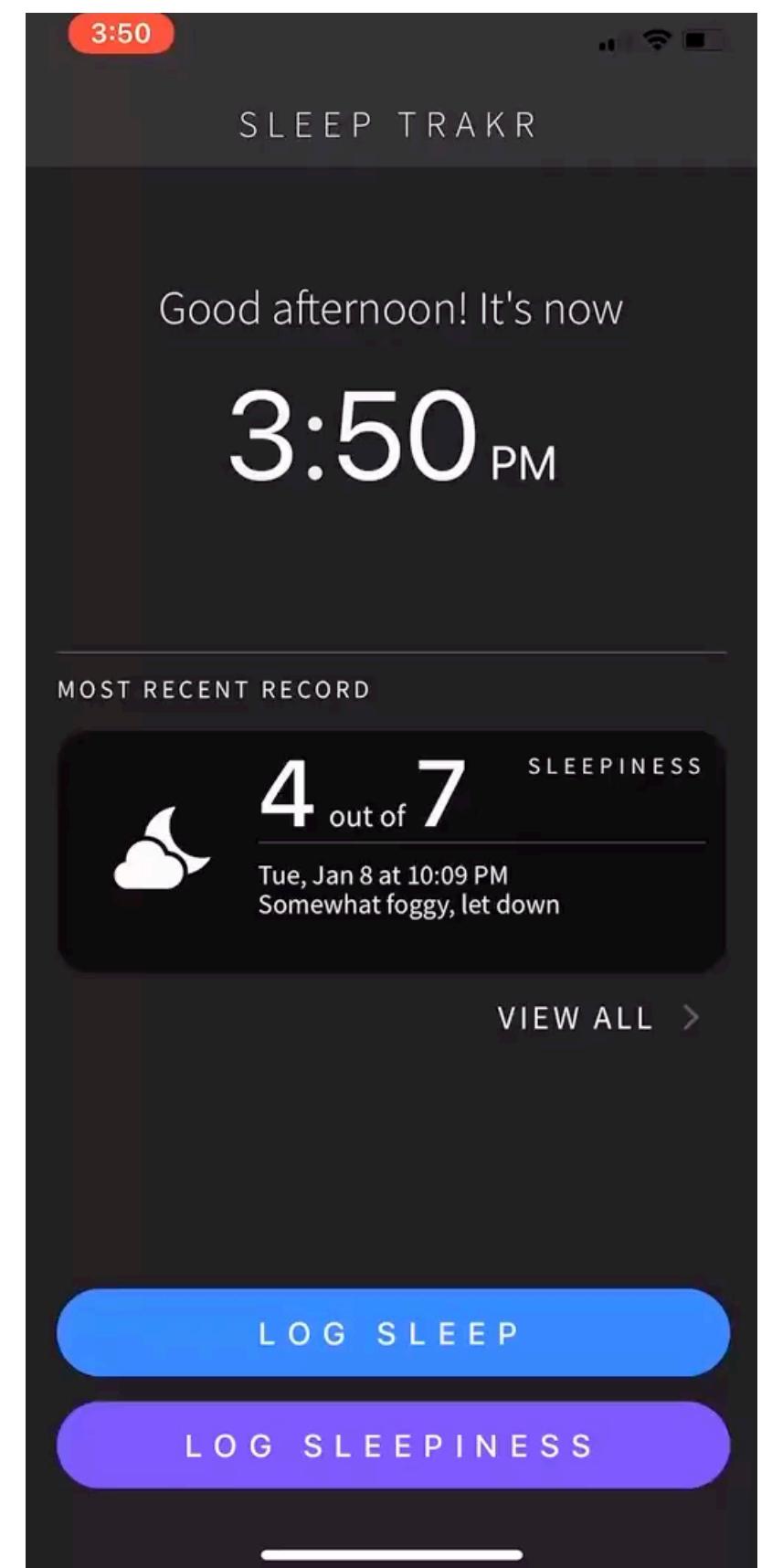
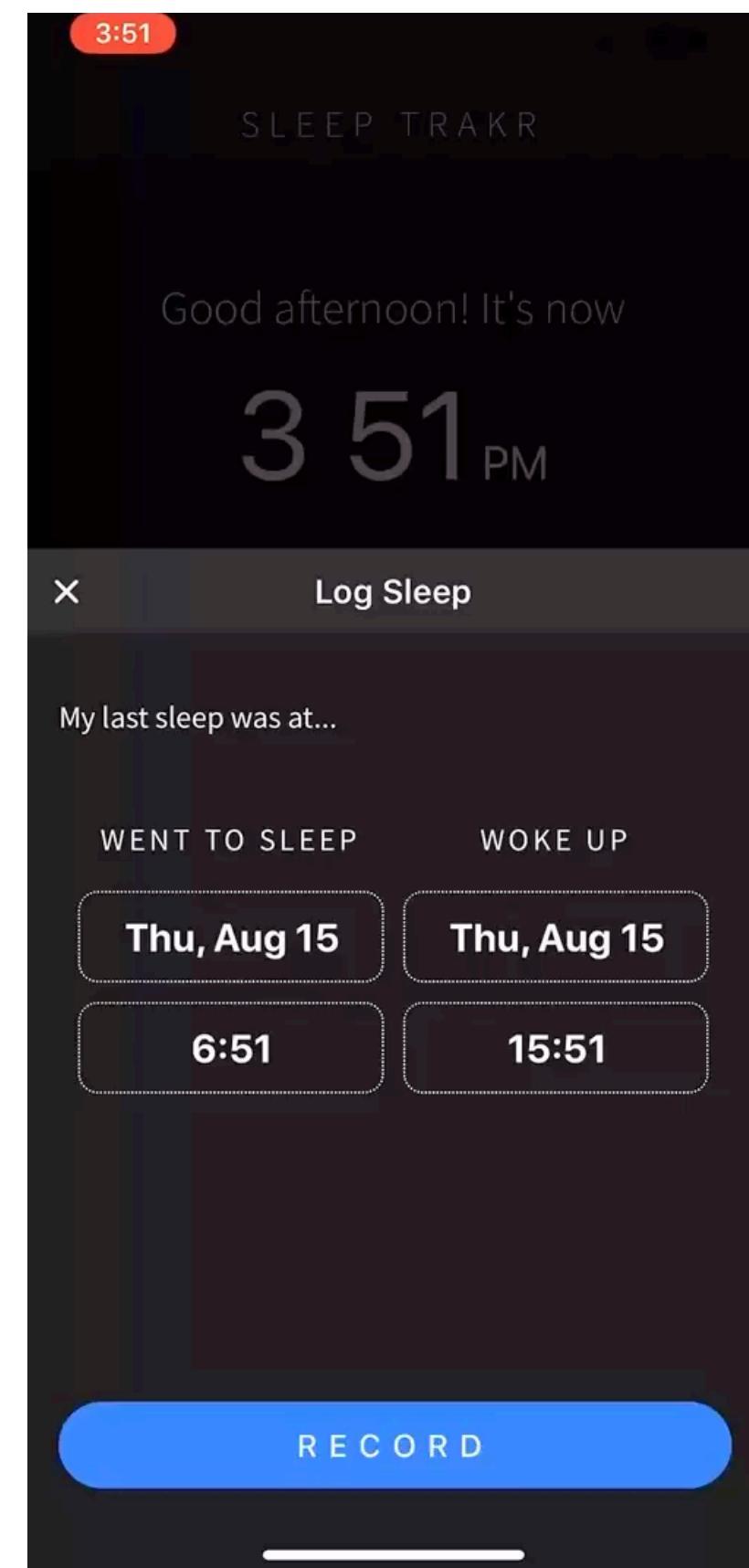
The screenshot shows a dark-themed application interface. At the top left is a "Log in" button. In the center, there's a large circular profile picture of Carly Rae Jepsen. Below the profile picture is a button labeled "Open Carly Rae Jepsen on Spotify". To the right of the profile picture, under the heading "Genres", is a list of musical genres: canadian pop, dance pop, electropop, indie optimism, pop, post-teen pop, and uk pop. To the right of that, under the heading "Carly Rae Jepsen's Top Tracks", is a table showing ten tracks with columns for #, Track, Duration, Artist, and Album. The tracks listed are: 1. Call Me Maybe (3:13, Carly Rae Jepsen, Kiss (Deluxe)), 2. OMG (with Carly Rae Jepsen) (4:36, Gryffin, OMG (with Carly Rae Jepsen)), 3. Good Time (3:26, Owl City, The Midsummer Station), 4. Cut To The Feeling (3:28, Carly Rae Jepsen, Cut To The Feeling), 5. Party For One (3:05, Carly Rae Jepsen, Dedicated), 6. Too Much (3:17, Carly Rae Jepsen, Dedicated), 7. I Really Like You (3:25, Carly Rae Jepsen, Emotion), 8. Now That I Found You (3:20, Carly Rae Jepsen, Dedicated), 9. Julien (4:55, Carly Rae Jepsen, Dedicated), and 10. Run Away With Me (4:11, Carly Rae Jepsen, Emotion). At the bottom of the interface are buttons for "Carly Rae Jepsen's Albums" and "Similar Artists".

#	Track	Duration	Artist	Album
1	Call Me Maybe	3:13	Carly Rae Jepsen	Kiss (Deluxe)
2	OMG (with Carly Rae Jepsen)	4:36	Gryffin	OMG (with Carly Rae Jepsen)
3	Good Time	3:26	Owl City	The Midsummer Station
4	Cut To The Feeling	3:28	Carly Rae Jepsen	Cut To The Feeling
5	Party For One	3:05	Carly Rae Jepsen	Dedicated
6	Too Much	3:17	Carly Rae Jepsen	Dedicated
7	I Really Like You	3:25	Carly Rae Jepsen	Emotion
8	Now That I Found You	3:20	Carly Rae Jepsen	Dedicated
9	Julien	4:55	Carly Rae Jepsen	Dedicated
10	Run Away With Me	4:11	Carly Rae Jepsen	Emotion

A4

Sleep Tracker in Ionic

- Learning goal: learn to leverage UI components in a mobile framework and align with principles of good mobile design
- Will implement an app to log daily sleep



A5

Beyond Web & Mobile

- Learning goal: experience how web & mobile implementation and design skills apply to a new device (VR headset, speaker, watch)
- Will implement a basic app (TBD) on a simulator and write about your experiences

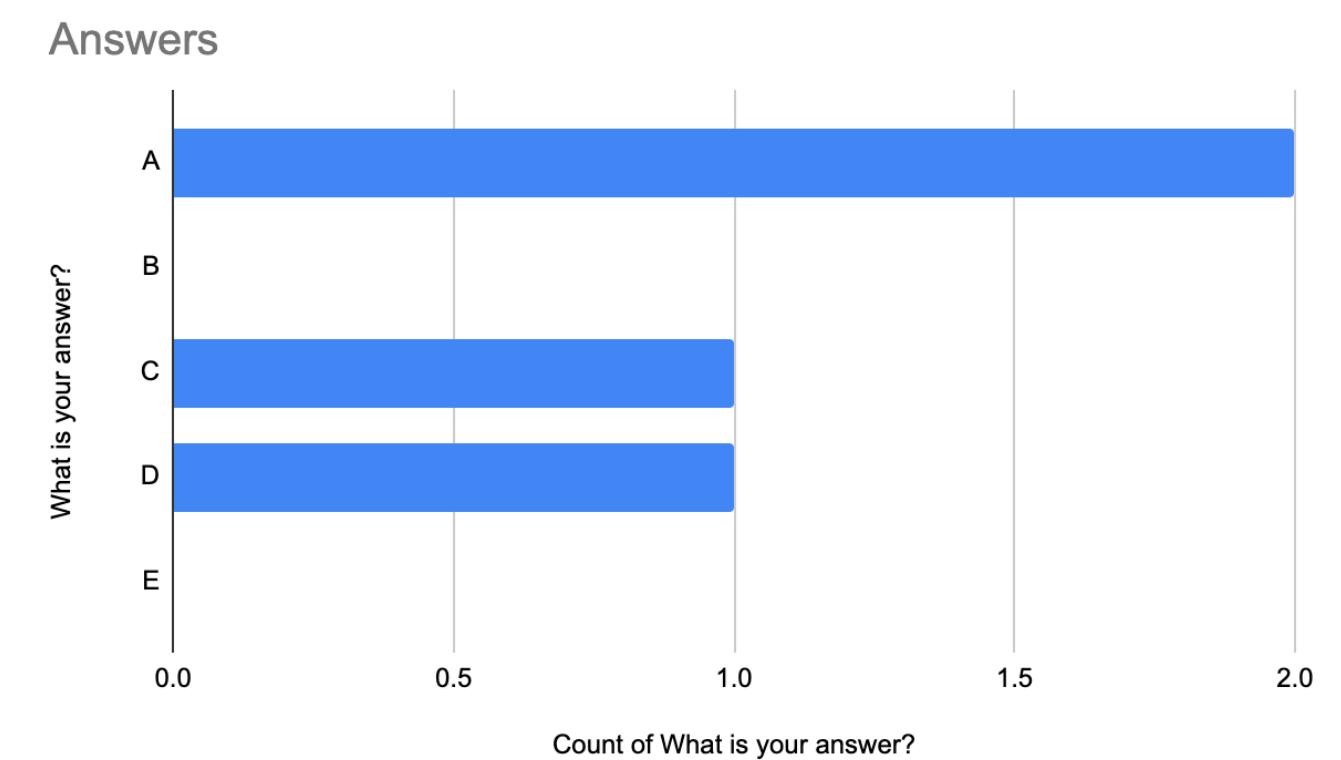


Exams

- Due to COVID-19, we decided not to have exams in this offering

Participation

- Survey question, think-share, discuss
- Opening other websites is a risk
 - Research suggests that students who use electronics during lecture take worse notes and perform worse on exams
 - Not much we can do about that, but aim to be mindful
- Will start Tuesday, January 11th
 - “Warm-up” next class



Participation

- You do not need to synchronously attend & answer survey questions, but we **strongly encourage** doing so if doing so is reasonable
 - If you do not attend synchronously, you have 24 hours to answer the survey question
 - All questions graded on *completion*, not correctness
- Attendance correlates with higher grades, improved learning and retention, and higher education satisfaction
- You can miss 2 responses without penalty

Participation

- Answer each other's questions on Slack!
 - You can also get participation credit this way
 - You can respond faster than we can
 - Often times, you've experienced the same pitfalls

Grading

- Assignments: 90%
 - A0 (1%), A1-A4 (18% each), A5 (17%)
- Participation: 10% (up to 5% extra credit)
 - In-class survey participation (10%), Slack participation (5%)

Discussion sections

- Are optional
- Are a great opportunity to get more depth on topics discussed in lecture
- May occasionally be open office hours
- You can go to either one, or both...
- Friday: introduction to Git and GitHub
 - We won't cover them in lecture, so a great opportunity to learn the basics

Calendar overview

Jan 2	Jan 3 No Discussion	Jan 4 Introduction & History 2:00-3:20 Zoom	Jan 5	Jan 6 HTML & Accessibility 2:00-3:20 Zoom	Jan 7 A0 Due Getting to Know You, Getting to Know Us	Jan 8
Jan 9	Jan 10 Git and GitHub Tutorial (TBD) 2:00-2:50 Zoom 3:00-3:50 Zoom 4:00-4:50 Zoom	Jan 11 CSS 2:00-3:20 Zoom Professor Epstein Office Hours 3:45-5:00 Zoom	Jan 12	Jan 13 Responsive Design & Javascript 1 2:00-3:20 Zoom	Jan 14	Jan 15
Jan 16	Jan 17 Martin Luther King Day	Jan 18 JavaScript 2 2:00-3:20 SSL 270 & Zoom Professor Epstein Office Hours 3:45-5:00 Zoom	Jan 19 A1 Due Responsive Portfolio in HTML and CSS	Jan 20 DOM Manipulation & Package Management 2:00-3:20 SSL 270 & Zoom	Jan 21	Jan 22
Jan 23	Jan 24 Javascript Demo (TBD) 2:00-2:50 SSTR 103 & Zoom 3:00-3:50 SSTR 103 & Zoom 4:00-4:50 SSTR 103 & Zoom	Jan 25 TypeScript & Data Visualization Tools 2:00-3:20 SSL 270 & Zoom Professor Epstein Office Hours 3:45-5:00 Zoom	Jan 26	Jan 27 AJAX, Fetch, & Promises 2:00-3:20 SSL 270 & Zoom	Jan 28	Jan 29

<https://inf133-wi22.depstein.net/calendar>

A0 due Friday!

- Background survey
 - Technologies you're familiar with
- Syllabus quiz
 - Take it until you get 100%

COVID-19 Considerations

- Omicron is a real bummer.
- These are difficult circumstances to learn in as well as teach in
- Moreso than usual, we don't know when your life is being disrupted
 - Please tell us so we can accommodate
- Within reason, we will allow late assignment submissions without penalty if you email us at least 24 hours before the assignment deadline
 - No late submissions for the last assignment

Reflection

- This is an applied course with a lot of programming.
 - About half of the class will cover implementation techniques
 - The other half is theoretical concepts which inform design and development
- We'll teach principles and languages at a high level,
but you'll need to pick up the specifics of APIs, packages, etc. on your own.
- We're happy to help, but we haven't used every aspect of every API

Reflection

You will learn a lot

“A LOT to learn for assignments. I will say that before I took the class, web development was something I had never even tried and had always sworn off doing (seemed too intimidating!), but now feel extremely confident with and have a huge interest in. So props to the professor for completely turning me to a subject that I probably would have ignored for years more!”

Reflection

But it will be hard

“The assignments were somewhat challenging for those people who have never been exposed to JavaScript and app development, but they are definitely helpful for us to understand the technology and skills used in the trend of website/app development”

“A lot of students might think this course as something very challenging (and it is) but it was really helpful in terms of implementing the ideas that we have learned in previous classes.”

Reflection

- We have high expectations
 - We want you to make cool things
- But we also care and will listen
 - Let us know how things are going, ask questions
- Be “all in”
 - If you’re not ready to commit, please drop now
 - Someone else will be happy to take your spot

Today's goals

By the end of today, you should be able to...

- Describe how society got to today's ubiquitous computing
- Hypothesize why web technology has become the de-facto tool for interface development
- Identify your course staff
- Summarize this course's goals and know how to find policies
- Describe upcoming course tasks

IN4MATX 133: User Interface Software

Lecture 1:
Introduction & History

Professor Daniel A. Epstein
TA Goda Addanki
TA Seolha Lee