Programming Usable Interfaces

Vassilis Kostakos
vassilis+pui@cmu.edu
Office: 2.91

Lecture 1: Introduction

Why UIs are important

but difficult to design and implement

Course

- **Classes: Tues/Thurs: 2-4pm**
- **Rooms: Sala 2, Sala 11**

- **Online forum: http://hci.dme.uma.pt/forums**

- **Class attendance/participation *very* important**

Pedagogical goals of class

- Learn to express your ideas computationally
- Learn to communicate with programmers
  - What's hard / easy to prototype
- Practice doing this in a rapid turn around, iterative, framework

What do we cover and do?

- Implementing prototypes of user interfaces
  - Why important
  - How to do it
- User testing of prototypes
- A little bit of principles of how UI systems go together
  - Focused on desktop appl (not web)

Questions?

Introduction to User Interfaces

Who are "Users"

- People who will use a computer system, device, or web site
- As contrasted with "Designers"
  - People who create the system
  - You
- Users != Designers

What is the "User Interface"

- Everything the user encounters
  - Functionality
  - Content
  - Labels
  - Presentation
  - Layout
  - Navigation
  - Speed of response
  - Documentation and help

What is "Usability"

- Learnability
  - Minimal training
- Efficiency
  - Productivity in performing real tasks
- Predictability
- Memorability
  - Little "re-learning" required
- Satisfaction
  - Pleasurable
- Flexibility

Why are Interfaces Important?

- Sit-down-and-use computers and software
  - Don’t read the manuals
- Usability critical to software sales
- HCI-trained people build better interfaces
  - Programmers don’t think like end-users
  - Exposure to different kinds of interfaces, problems

Why Important? Cont'd.

- **Well-defined methods & techniques**
- ***Not just opinions*, luck, domain experiences**
- **Very expensive to *not* do usability engineering**
  - **Interfaces will be re-designed before or after release**
  - **Studies shows that usability engineering saves money (5000x)**

Why Important? Cont'd.

- Novices will be more effective and quicker
- Make experts more efficient
- Reduce calls to support center
  - $30-100 per call
- Reduce errors
- Can help identify what's really needed and would be helpful

Why Important? Cont'd.

- Recognized by industry, government, etc.
  - Plenty of jobs
  - Money for research
- Significant time and code devoted to HCI now!

End result: big impact on the world

- **Large numbers use computers**
  - **Most Americans & Europeans own computers**
  - **Majority have internet access**
- **No one in our society is not affected in some way by computers**
- **Critical time:**
  could go good

  could also go bad

Bad UIs Can Cause Disasters

- **Therac-25** (6 accidents 1985-87)

  http://courses.cs.vt.edu/~cs3604/lib/Therac_25/Therac_1.html

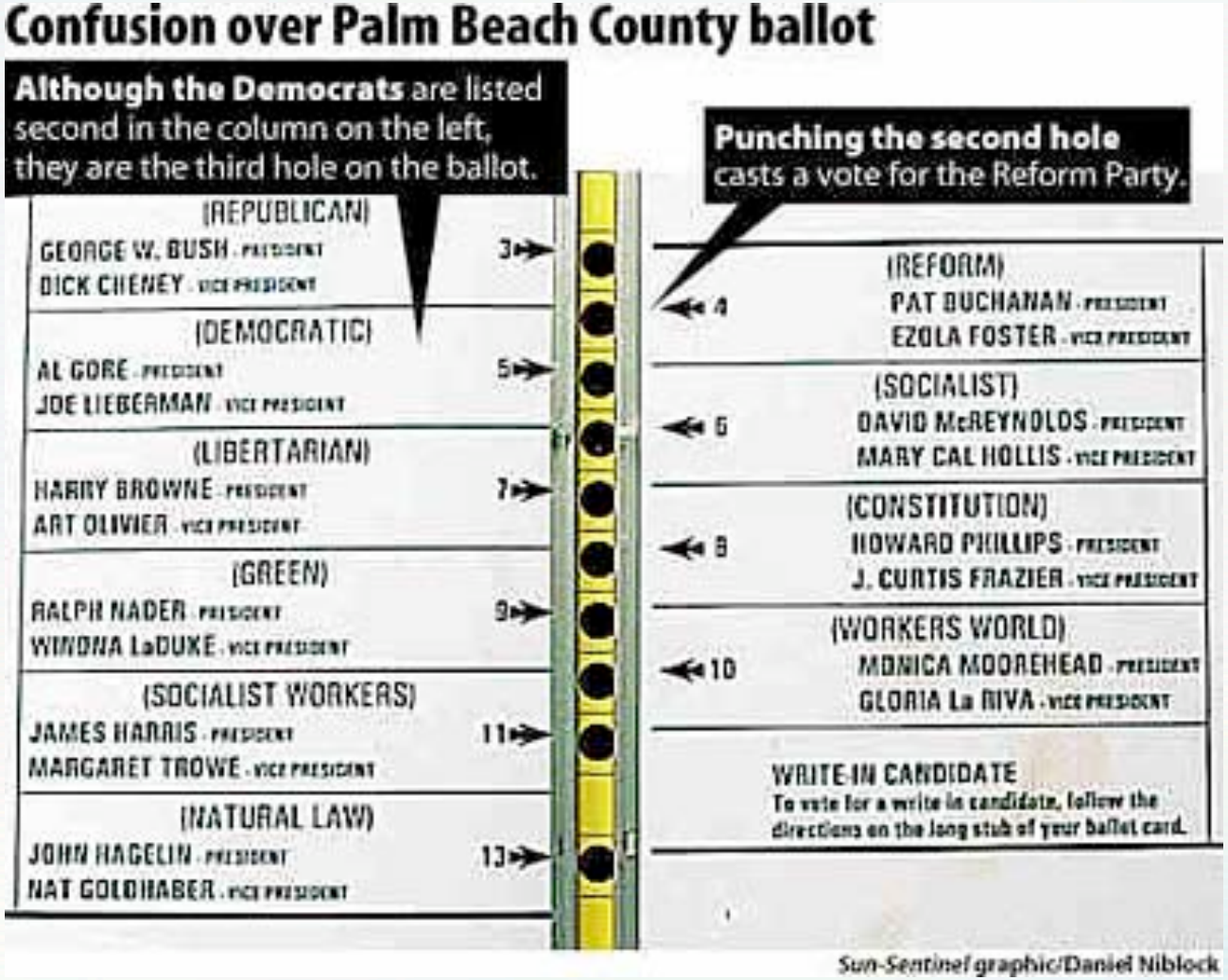
  - Repeated in 2000 (5 more deaths)

  http://archives.seattletimes.nwsource.com/cgi-bin/texis/web/vortex/display?slug=radiation14&date=20010614

- **Aegis (July 4, 1988)**
  - Iranian Airbus shootdown by *Vincennes*

  http://washingtonpost.com/wp-srv/inatl/longterm/flight801/stories/july88crash.htm

Florida Ballots (2000)



Usability is Also Important for More Mundane Reasons

- Usability is critical to SW sales
- Novices more effective quicker
- Makes experts more *efficient*
- Reduce calls to support center
  - Can cost $30 - $100 per call
- Reduce errors, etc…

Usability is Also Important for More Mundane Reasons

- Very expensive to *not* do usability engineering
  - Interfaces will be re-designed: before or after release
  - Studies show that usability engineering saves money
    - Well-defined methods and techniques
    - *Not just opinions*, luck, domain-experience

But, it's hard to design good UIs

"It is easy to make things hard. It is hard to make things easy."
(A. Chapanis 1982)

- User interface design is a creative process
- Designers have difficulty thinking like users
  - Often need to understand task domain
  - Can't "unlearn" things

Hard to Unlearn

But, it's hard to design good UIs

- **Specifications are generally wrong**
  - **"Only slightly more than 30% of code developed in application software development ever gets used as intended by end-users. The reason for this statistic may be a result of developers not understanding what users need."**

    -- Hugh Beyer and Karen Holtzblatt, "Contextual Design: A Customer-Centric Approach to Systems Design" ACM Interactions, Sep/Oct 1997

  → need for prototyping and iteration

But, it's hard to design good UIs

- Tasks and domains are complex
  - MacDraw 1 vs. Illustrator
  - Word 1 vs. Office XP
- Existing theories and guidelines are not sufficient
  - people are complex

But, it's hard to design good UIs

- All UI design involves tradeoffs
- Pressures:
  - Standards (style guidelines, related products)
  - Graphic design issues
  - Technical writing
  - Internationalization
  - Performance
  - Multiple platforms
  - High vs. low-level details
  - Legal issues
  - External factors (social issues)
  - Time to develop and test ("time to market")

What to do?

What to do?

Usability slogans for Nielsen text?

- Your best guess is not good enough
- The user is always right
- The user is not always right
- Users are not designers
- Designers are not users
- Less is more
- Details matter

What to do?

Usability slogans for Nielsen text?

- Your best guess is not good enough
- The user is always right
- The user is not always right
- Users are not designers
- Designers are not users
- Less is more
- Details matter

The user is not like me!

What to do?

- No silver bullet
- But we do have methodologies which work
  - Have seen much of this in "methods"
  - Practice here and look at prototyping and implementation aspects

Unfortunately, user interfaces are also hard to implement

- They are hard to design, requiring iterative implementation
- They are *reactive* and are programmed from the "inside-out"
  - Event based programming
  - More difficult to modularize

Unfortunately, user interfaces are also hard to implement

- Need for robustness
  - No crashing, on any input, ever
  - Helpful messages & recover gracefully
  - Aborts
  - Undo, …
- Harder to test than other SW
  - Few tools for regression testing

Unfortunately, user interfaces are also hard to implement

- Little language support
  - Primitives in computer languages make bad UI
  - Enormous, complex libraries
- Complexity of other tools
  - Full bookshelf for Microsoft MFC, etc.
- Difficulty of modularization

Unfortunately, user interfaces are also hard to implement

- Often require multi-processing
  - Deal with user typing; aborts
  - Window refresh
  - Windows system as different process
  - Multiple input devices
- Real-time requirements for handling input events
  - Output: 60 times/second
  - Keep up with mouse tracking
  - Video, sound, multimedia

Because of the difficulty, we can end up with a tension

Good UI vs. ease of implementation

Because of the difficulty, we can end up with a tension

Good UI vs. ease of implementation

"Ye who writes the code has the last word"

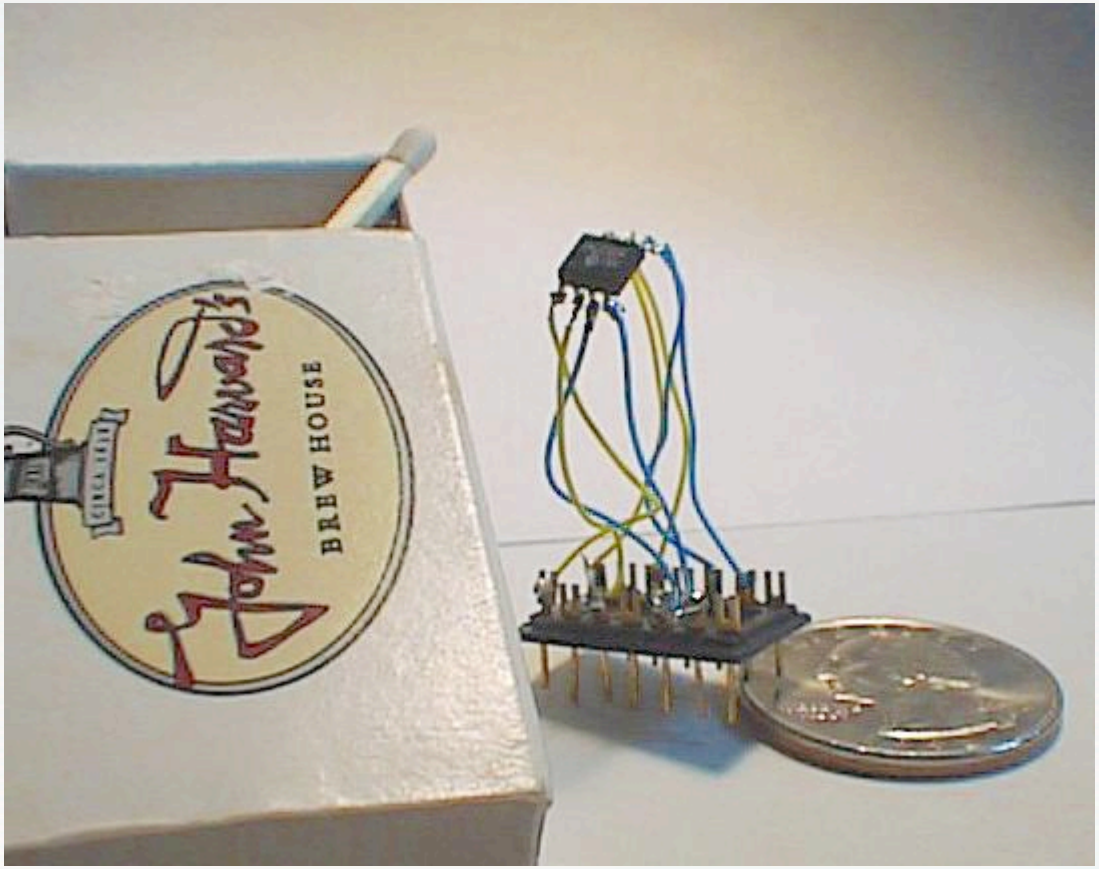
- Being able to express your ideas computationally and speak well with programmers puts you in a strong position to advocate for the user
- That is why we have this class

A critical time

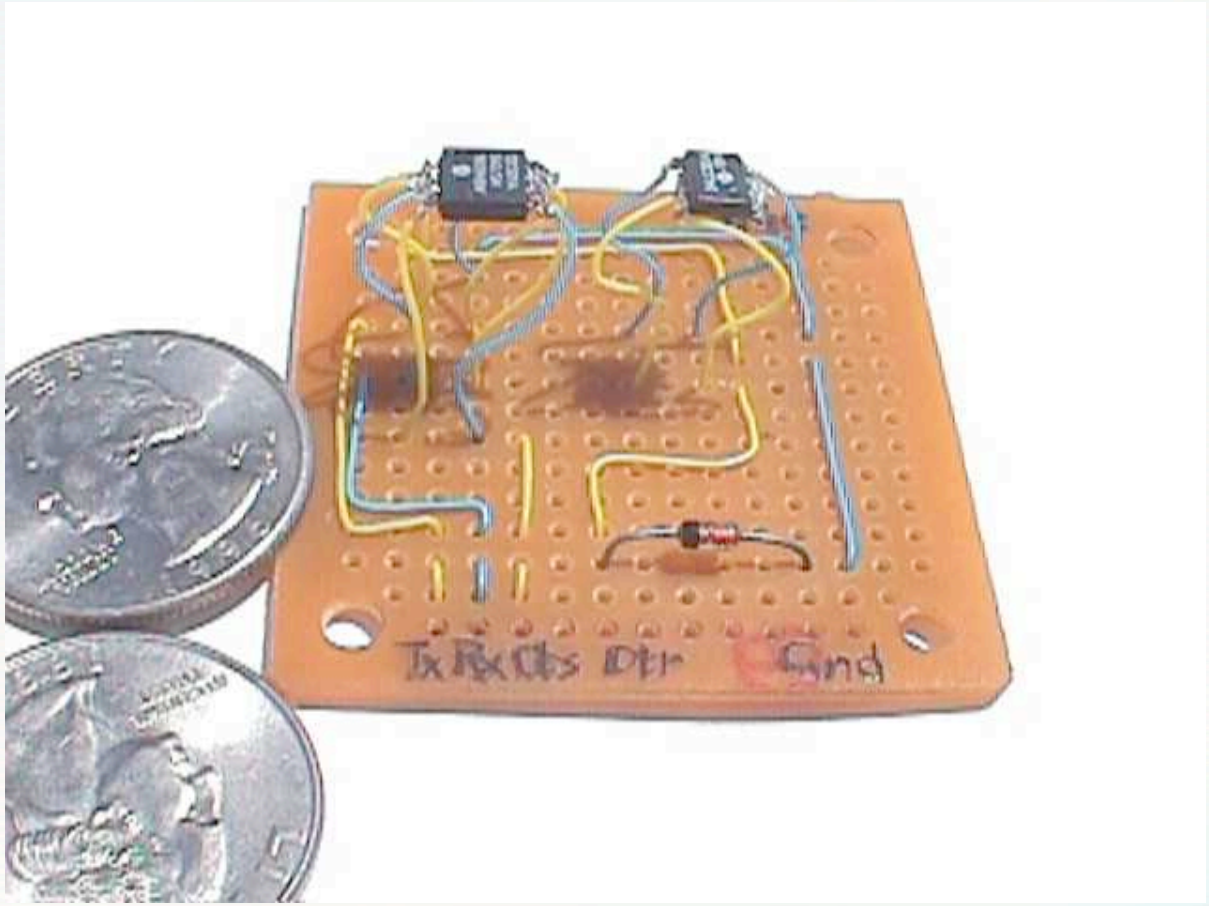
- **Computers are exploding into society**
  - **Pervasive computing power**
  - **Small, cheap, powerful**
  - **→Everywhere: watches, phones, homes, …**

A few years back: World's smallest web-server (runs Linux)

That only lasted 3 months…



That only lasted 3 months…



http://www-ccs.cs.umass.edu/~shri/iPic.html

Processor costs < $1

- If you can add $3-$5 to the cost of something, you can add a processor
  - if there is something of value to be gained (doesn't have to be much)

Processor costs < $1

- ~80x faster than the computer that "landed men on the moon"

  <50Khz (0.05Mhz) vs. 4Mhz

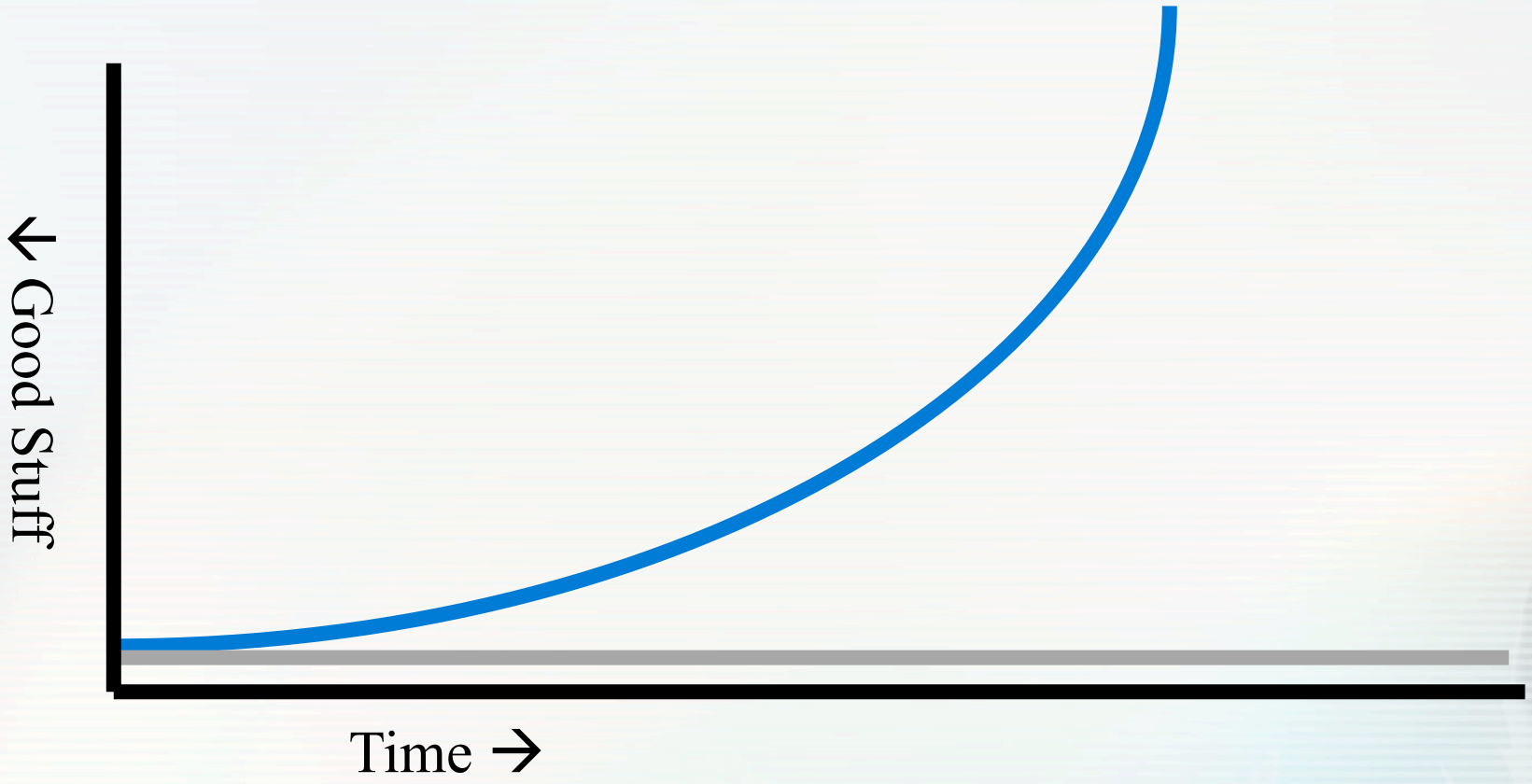
- And ~2x memory (incl. 2nd chip)

  ~2K RAM, ~64K ROM vs.

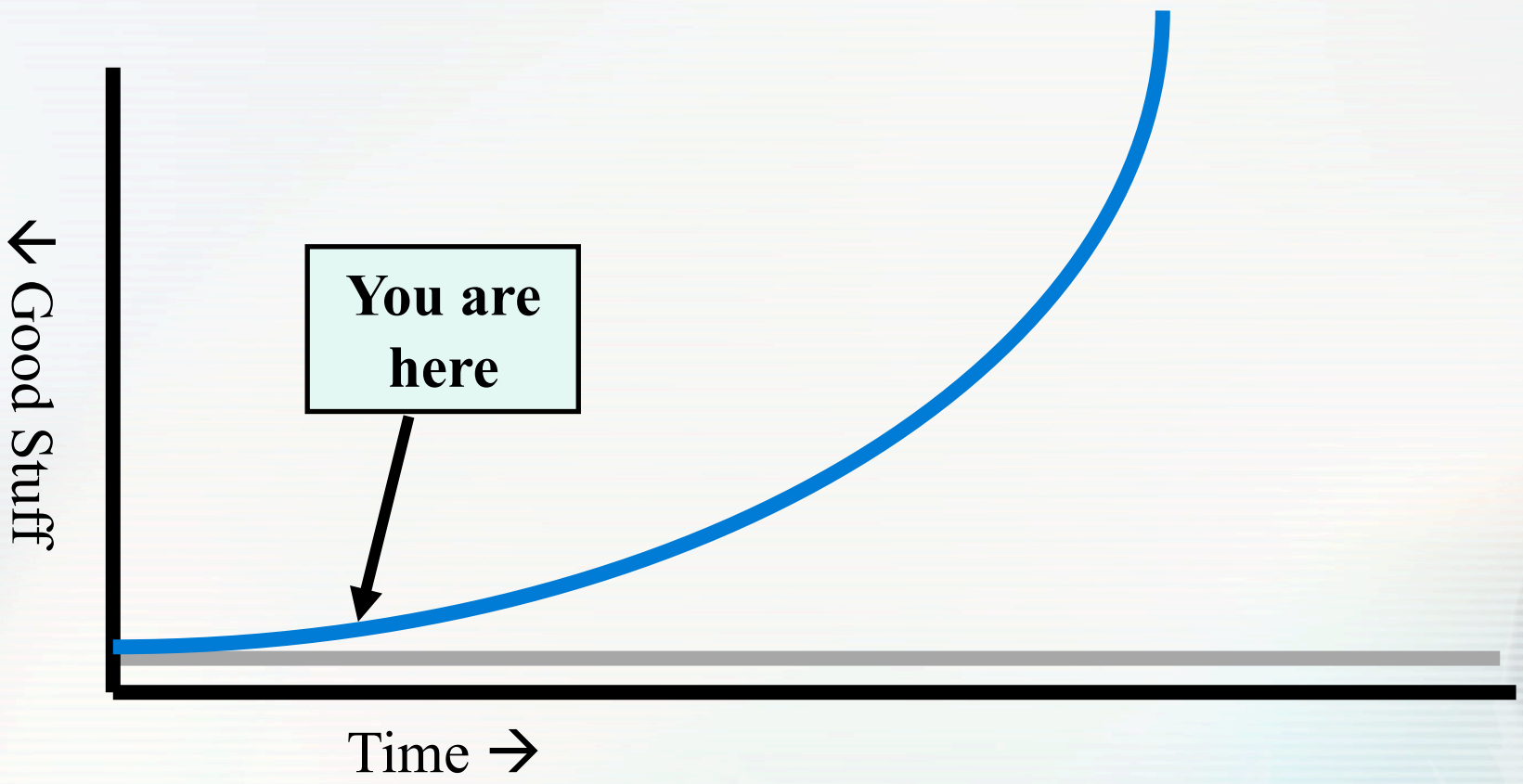
  41b RAM, 259K EEPROM

We haven't seen anything yet

- **Pop quiz: what is this**



Moore's law



Moore's law implies

- At given price point, CPU speed doubles every 18 months
  - Low end (<$1) chip will have today's high-end performance in ~10 years
- Corollary: at a given performance point price drops fast

Hard to really understand exponential growth

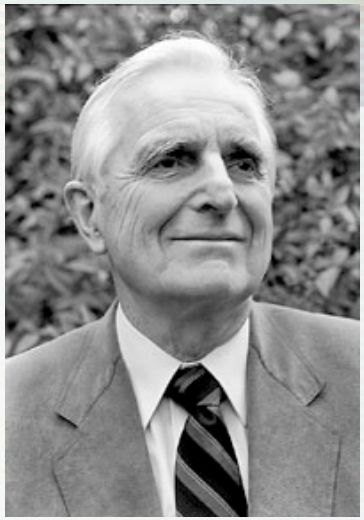
- There has been huge performance gains since (say) 1965

Hard to really understand exponential growth

- There has been huge performance gains since (say) 1965
- EVERY SINGLE BIT of those speed gains will happen again in 18 months!!
- Big changes can have huge impacts

Thought experiment

From Douglas Engelbart
(invented mouse & hypertext)



- **Suppose you and everything you could see suddenly got exactly 10 times bigger.**

- **Could you tell the difference?**

Thought experiment

From Douglas Engelbart
(invented mouse & hypertext)

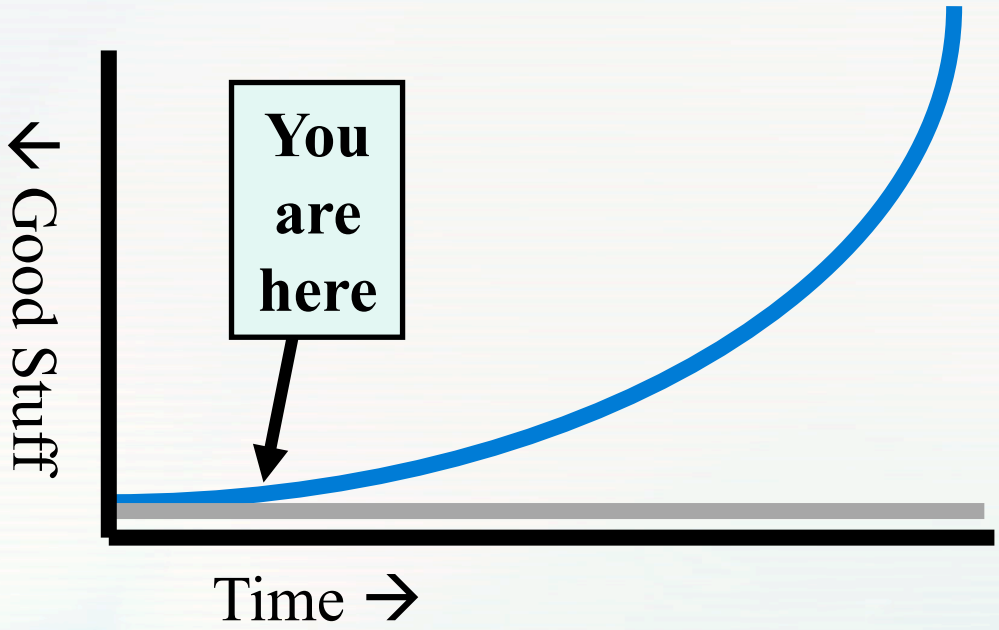
- **Mass scales by volume, but strength of your chair scales by cross-sectional area**
  - **Chair collapses under 10x weight ratio**
  - **Not so important because most of your bones will break for the same reason**
  - **Not really an issue because oxygen consumption scales by volume, but cells' ability to absorb it scales by area**

Thought experiment

From Douglas Engelbart
(invented mouse & hypertext)

- **Moral of the story:**

  Large (order of magnitude) changes can have a huge impact



End result: big impact on world

- **Large numbers use computers**
- **No one in our society is not affected in some way by computers**
- **All hell is about to break loose!**

Cool! But only if its usable!

Administrative Details

- Sign up to the forum

Questions?