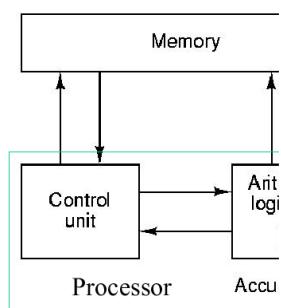


CS 349 User Interfaces

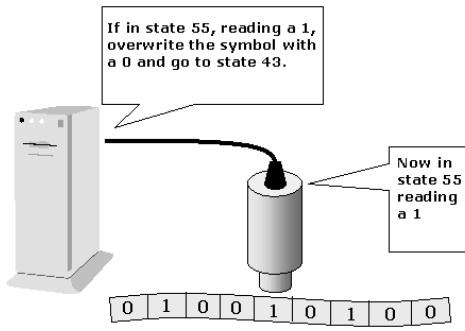
<https://www.student.cs.uwaterloo.ca/~cs349>

Daniel Vogel

computer

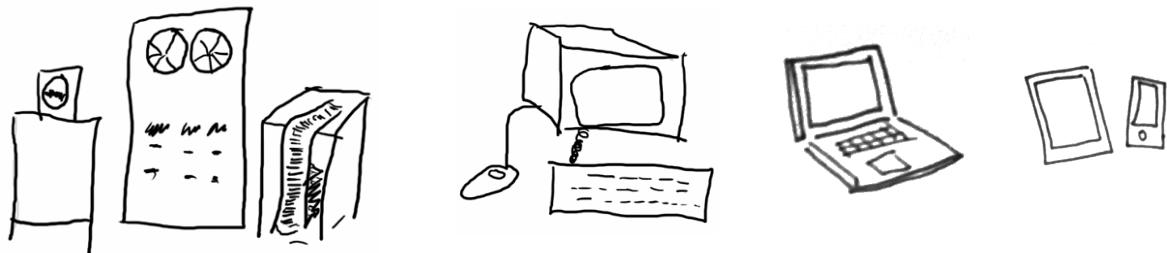


$$M = \langle Q, \Gamma, b, \Sigma, \delta, q_0, F \rangle$$



User Interface

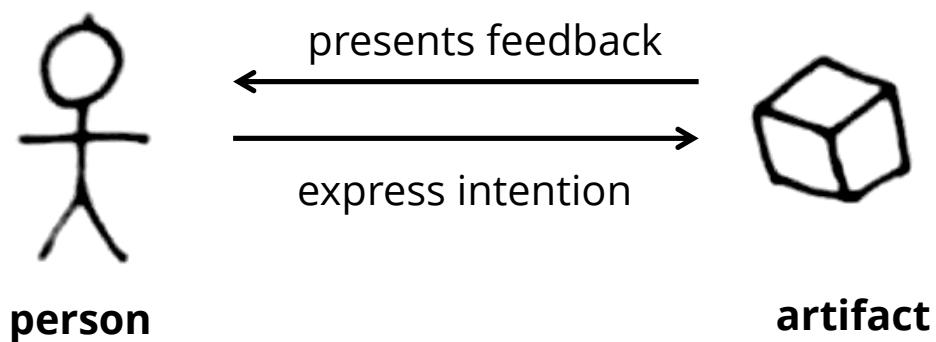
- The place where humans and computers meet
- The human's view of the computer



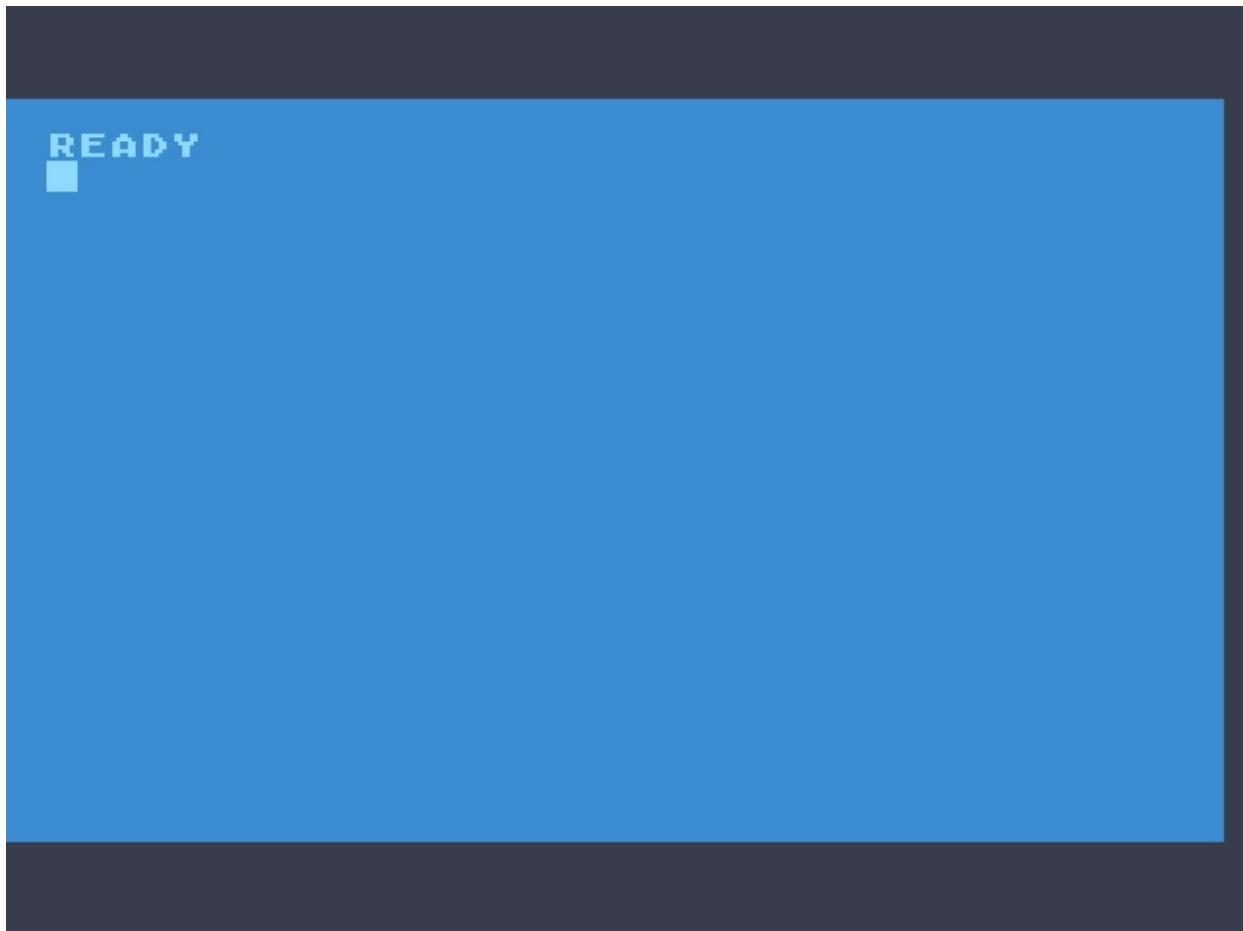
Introduction 5

Definition: User Interface

- A user interface is the place where a person expresses intention to an artifact, and the artifact presents feedback to the person.



Introduction 6



Bal 196.33
3-28 37.55
Bal 158.78
158.78 Final Bal

ATARI 400

241 A.J Market

System includes Console,
Basic Language Cartridge,
Basic Language Instructional Cassette,
322-page programming textbook,
operating manual,
TV switch box and AC adapter

\$ 549⁰⁰
(sold on facing page)

Console attaches to
any color or
black-and-white TV

HARDWARE, SOFTWARE AND ACCESSORIES FOR
THE ATARI 400™ PERSONAL COMPUTER SYSTEM

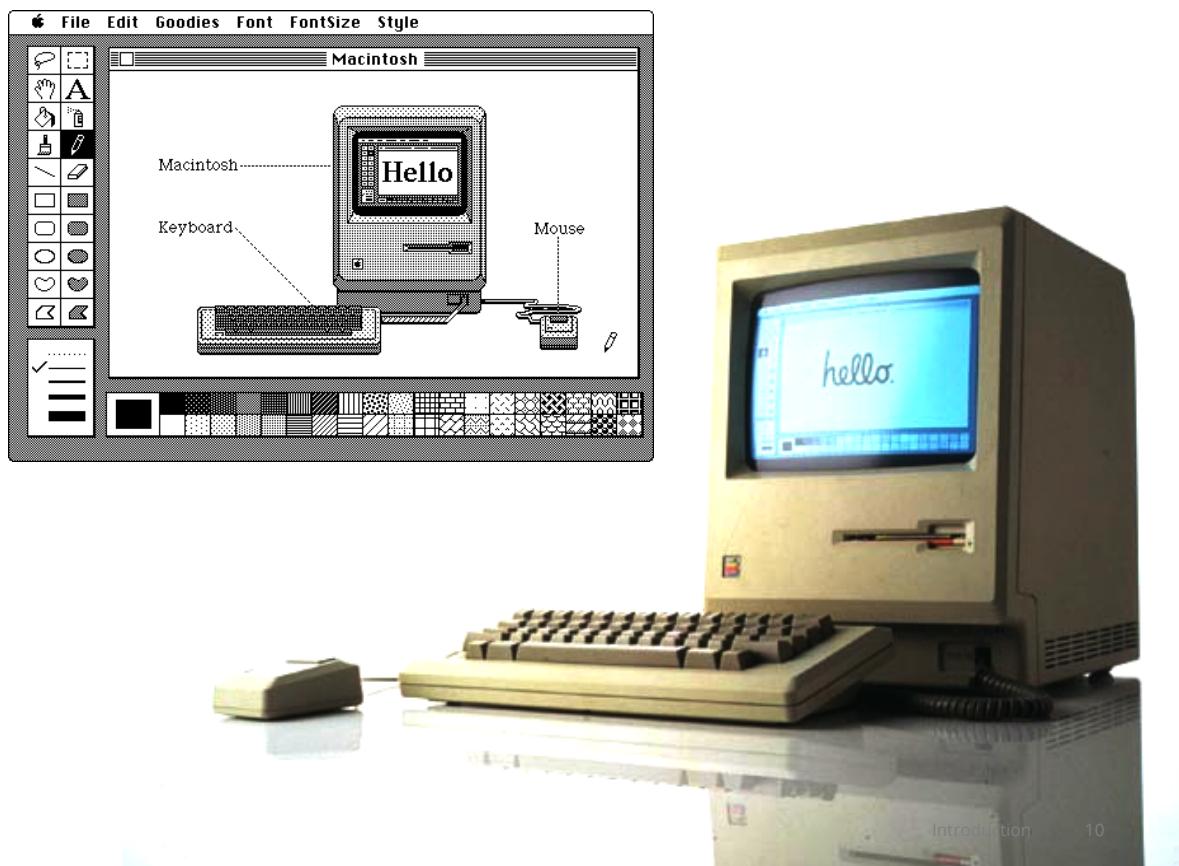
(1) PROGRAM RECORDER
Use with Educational Cassettes (sold
at right) or use to record and store
your own programs on standard audio

(3) EDUCATIONAL CASSETTES
Traditional educational techniques are used to
help teach a subject. While a written text and
graphics appear on the screen, a lecturer covers

Atari User Interface

1. Place a cassette tape in the data recorder.
2. Press REWIND or FORWARD to position tape where the program is located.
3. Boot the computer to the Atari BASIC READY prompt.
4. Type the command: LOAD "C:" (assuming program was saved with SAVE "C:")
5. The computer will "beep" as a signal for you to press PLAY on the recorder.
6. Press the RETURN key on the computer keyboard, and the program will load into the computer.
7. Press STOP on the recorder when loading has finished.
8. Enter the command: RUN

Introduction 9



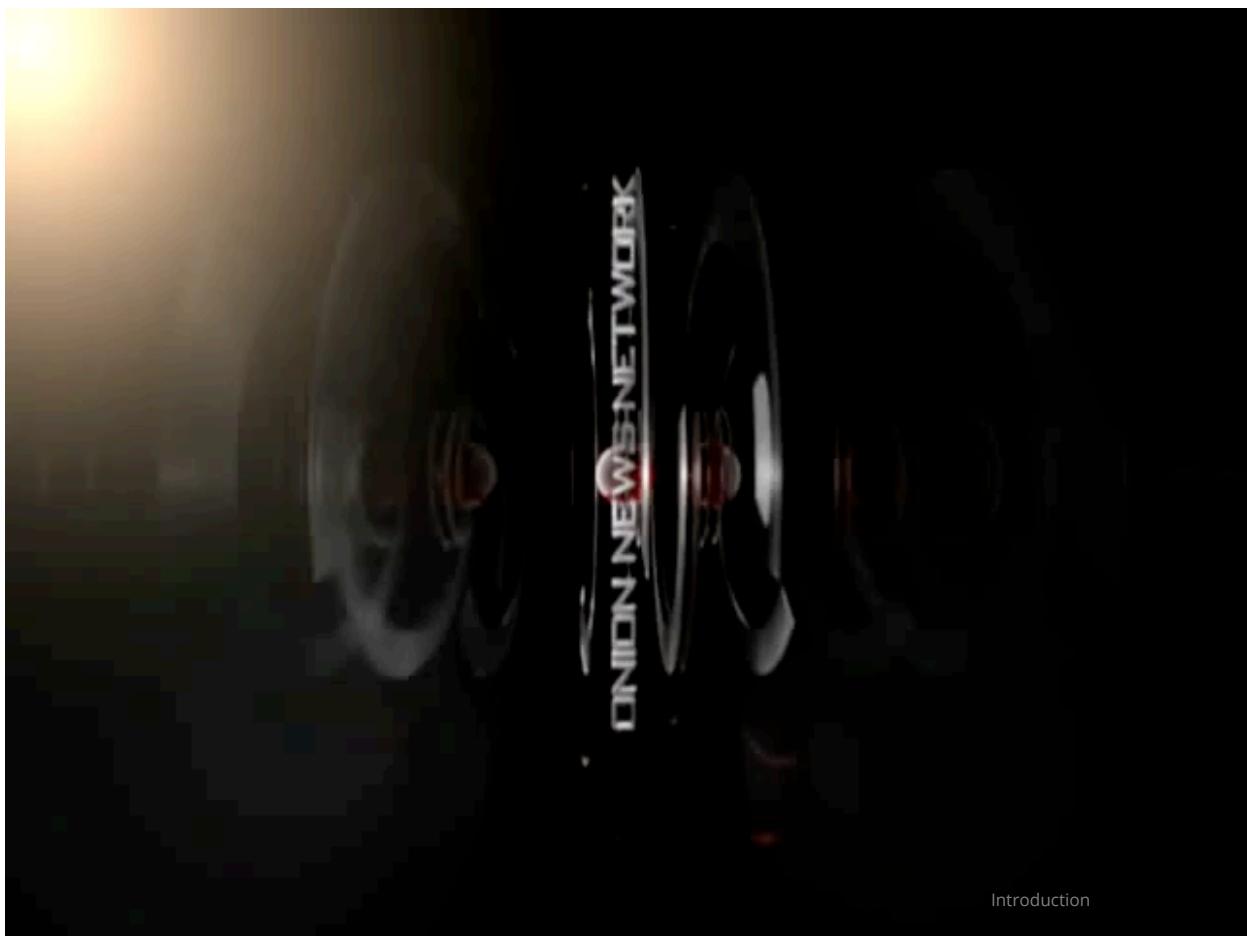
Introduction 10

User Interfaces

- Does a microwave have an interface?
- A refrigerator?
- A door bell?
- A hammer?



Introduction 11



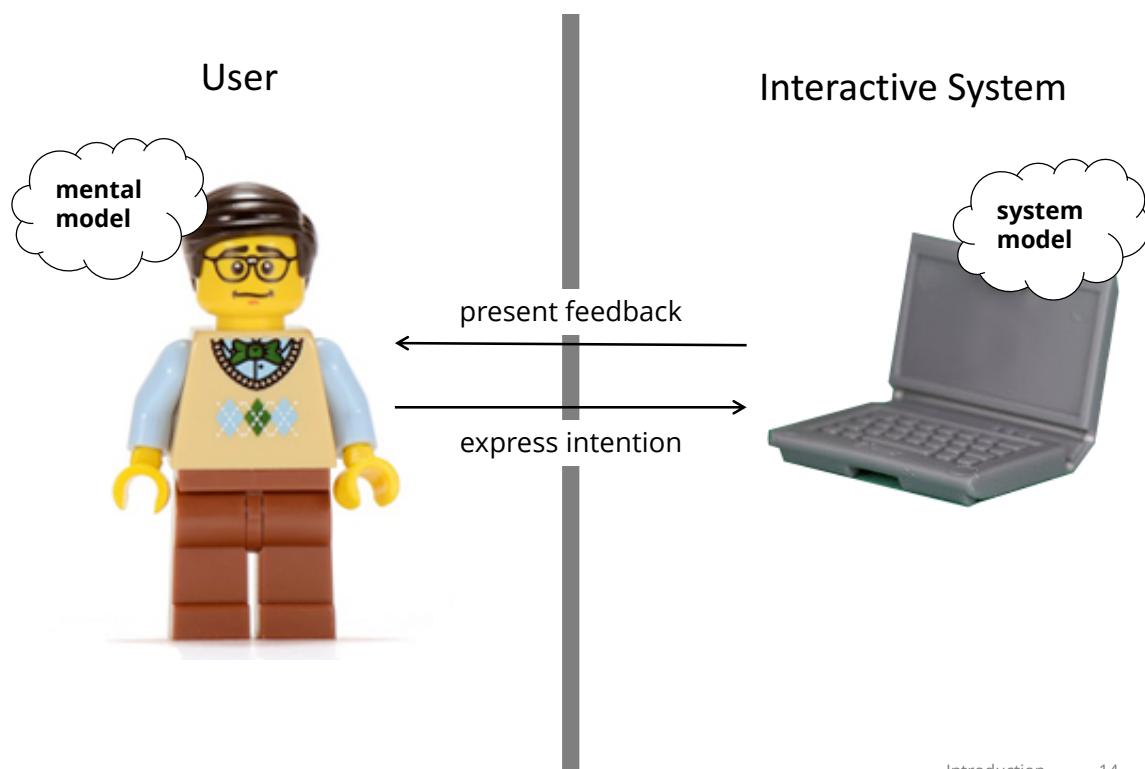
Introduction

Interactive System Architecture



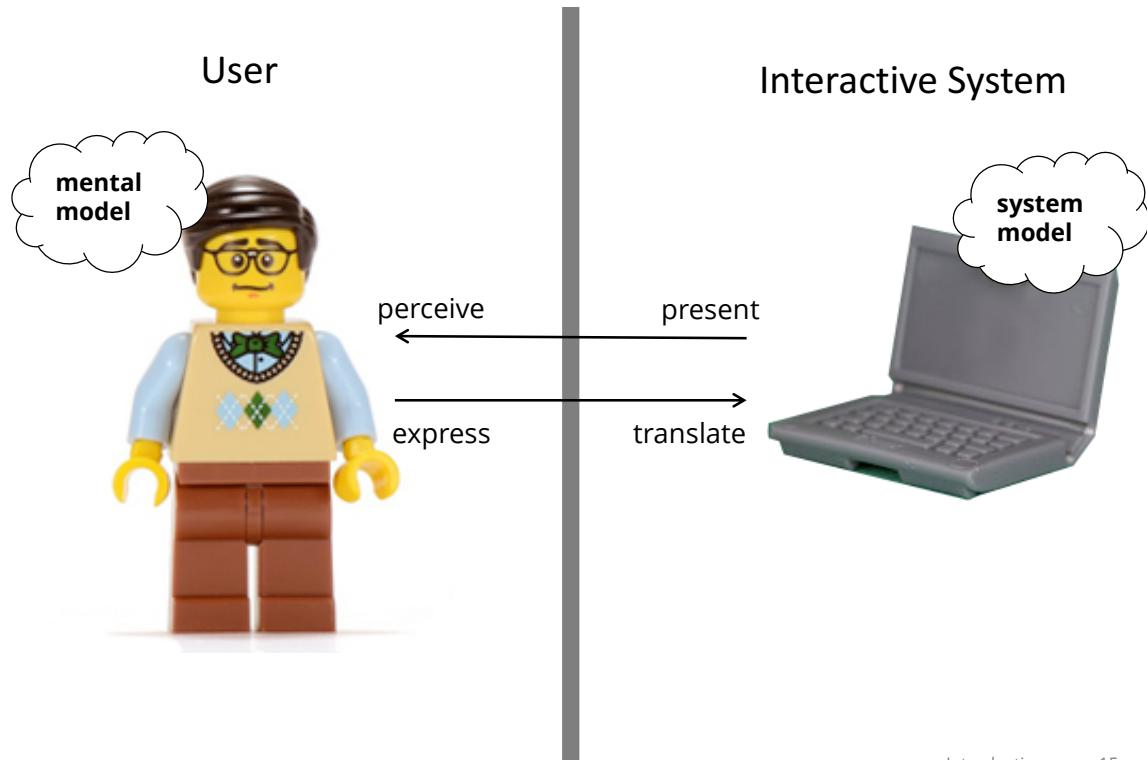
Introduction 13

Interactive System Architecture



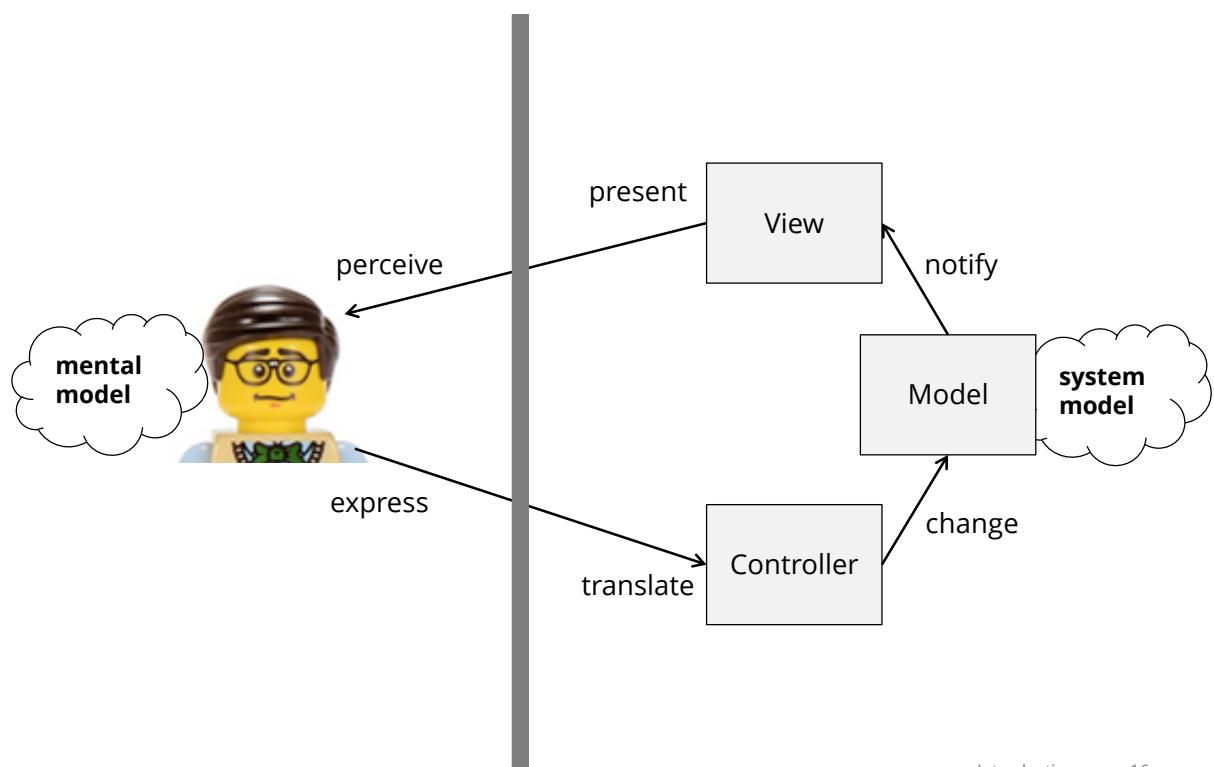
Introduction 14

Interactive System Architecture



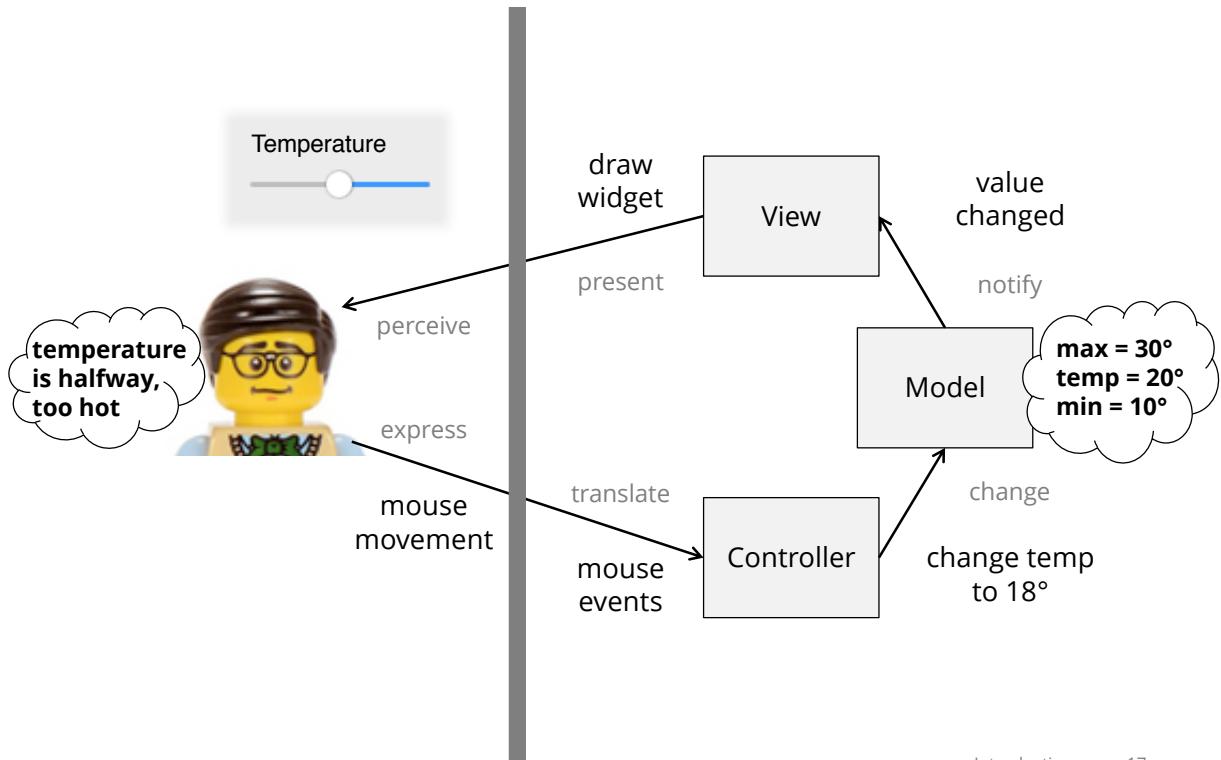
Introduction 15

Model-View-Controller (MVC)



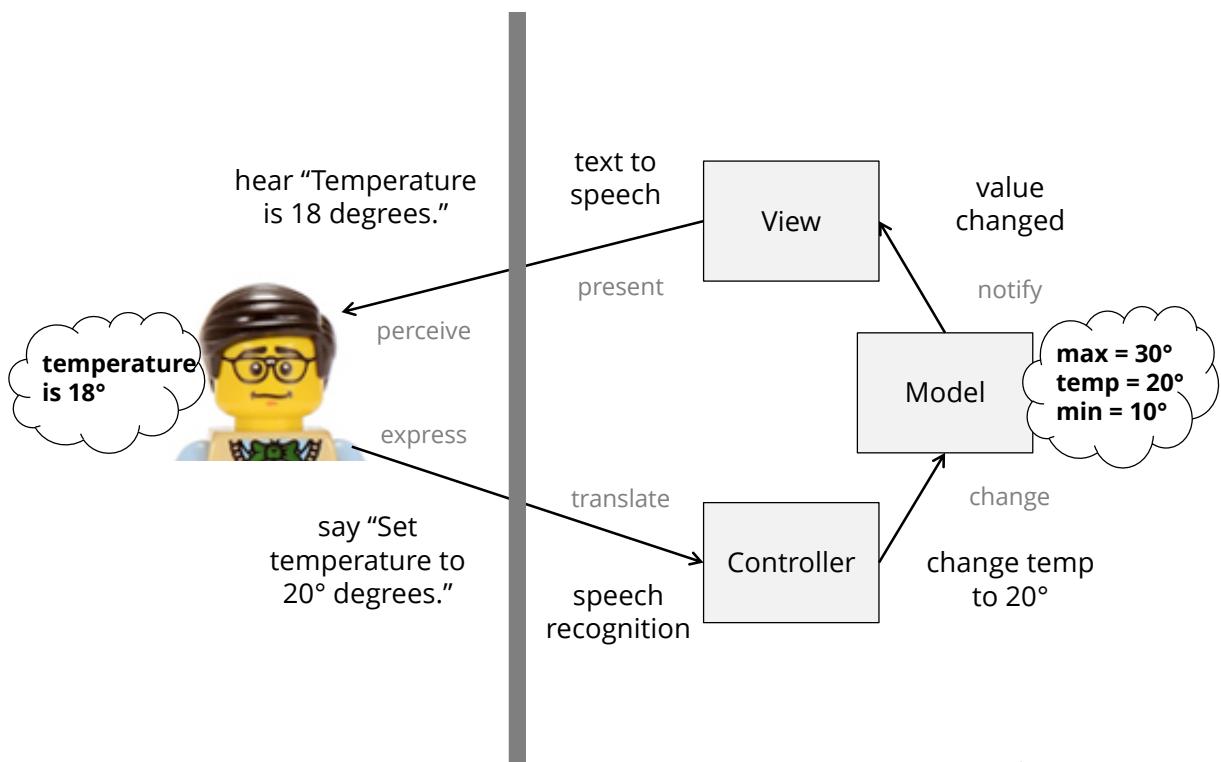
Introduction 16

Graphical Temperature Control



Introduction 17

Speech Temperature Control



Introduction 18

Interface vs. Interaction

- What is the difference between an *interface* and *interaction*?
- **Interface** refers to the external presentation to the user
 - Controls (what you can manipulate to communicate intent)
 - Feedback (what the program uses to communicate its response)
- **Interaction** refers to actions by user and system over time
 - interaction is a dialog with a cycle alternating between the user manipulating controls and the system responding with feedback

Introduction 19

Interface and Interaction Design

- What is the **interface**?
- What is the **interaction**?
- Why is this good **interaction design**?



Introduction 20

Interaction Design

- Challenging because of variability in users and tasks
 - Varying levels of expertise
 - Range of tasks performed with the same tool
- No one “right way” to design an interface, interfaces can always be improved

Introduction 21



Introduction 22

Empowering People

- Well designed interfaces empower people to do things they couldn't otherwise do
 - Desktop publishing, grassroots journalism (blogs), movie production, music production, image editing, assistive technologies, ...
- A well designed tool can change the world
 - The web browser, Linux, iPhone, spreadsheet, email, instant messaging, git, live streaming, ...



Syllabus

What to expect

How to be successful

Next steps

Learning Objectives

- Underlying UI architecture and algorithms
- How to implement UIs from scratch and using UI frameworks
- Basic theories and methods for UI design
- An introduction to Human-Computer Interaction

Note: user-centered design is covered more extensively in CS 449.

Grading

- Assignments: 40%
 - Must pass assignment portion to pass the course
- **Midterm (Mon Oct 23, 7 – 9 PM): 20%**
- Final: 40%
 - Must pass weighted exam average to pass the course

Laptop and Tablet Use

- Laptops and tablets may only be used to:
 - view lecture slides
 - take your own lecture notes
 - look up content related to lecture
 - try lecture sample code
- During lecture you may not:
 - work on assignments
 - email, text, im, ...
 - facebook, twitter, tumbler, youtube, ...
- No smartphones
- No headphones

**Be
Present**

Introduction 27

Topics

- History
- Windowing Systems
- Drawing
- Events
- Modern GUI Systems
- Intro to Java
- Widgets
- Event Dispatch
- Event Handling
- Layout
- 2D Graphics
- Affine Transformations
- Model-View-Controller
- Input Devices
- Input Performance
- Direct Manipulation
- Undo
- Clipboard, Drag-and-Drop
- Responsiveness
- Touch Interfaces
- Touchless Interfaces
- Wearables
- Mobile UI
- Web UI
- Design Principles
- Visual Design
- Accessibility

Introduction 28

Assignments

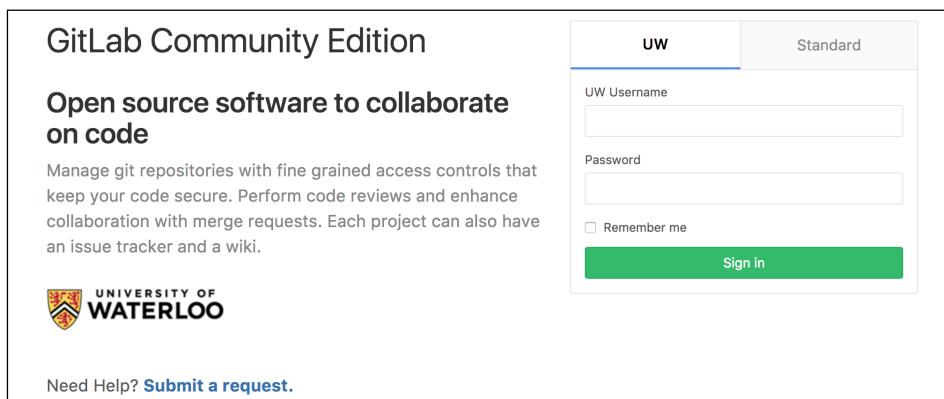
- 1% A0: Git and VM Setup
- 10% A1: Events and Drawing (C++)
- 10% A2: Model-View-Controller; Layout (Java)
- 10% A3: Direct Manipulation (Java)
- 10% A4: TBD (Java or Android)

Assignment Policies

- Due dates are typically Fridays at 5:00 PM
 - no extensions, no late assignment accepted
- Always test on the official course **Virtual Machine (VM)**
 - this is what TAs use to mark
- If we can't compile and run your code, you could get 0
 - submit all code, assets, and a **makefile**
- All submissions must made be via your **Git repository**
- Assignments are your own individual work
 - automated plagiarism detection

Git

- All submissions must be made via your Git repository
 - <https://git.uwaterloo.ca>
- TAs will push assignment feedback and marks to your Git repo
- A0 is about getting Git setup
- **Git Tutorial Mon Sept 18 at 7 PM in MC 4020**



GitLab Community Edition

Open source software to collaborate on code

Manage git repositories with fine grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

UNIVERSITY OF WATERLOO

Need Help? [Submit a request.](#)

Standard

UW Username

Password

Remember me

Sign in

Introduction 31

CS 349 Virtual Machine

- Must test your code on official course Virtual Machine (VM)
 - free Virtual Box VM host
- Download and setup instructions coming soon ...



Resources

- Web Site: www.student.cs.uwaterloo.ca/~cs349/
 - schedule with topics, lecture notes, sample code, assignment, midterm dates
- Piazza: <https://piazza.com/uwaterloo.ca/fall2017/cs349/home>
 - announcements
 - questions and answers about course material and assignments
- Recommend Text Books
- Curated links for Git, makefiles, X Windows, Java, ...

Getting Help

- Office Hours
 - Lecture and assignment help
 - Details posted on course website
- Piazza
 - Sign up and *use your real name*
 - Staff will monitor (best-effort response, no time-guarantees)
 - When posting
 - Search before you post
 - Use a meaningful title for every post
 - Answer questions, but don't be too explicit
 - No code taken directly from assignment
 - Build one collaborative answer
 - Be nice

Next Steps

- Explore the web site
- Sign up for Piazza
- A0 and VM setup steps will be out this weekend