User Interface Technology

Block 4 2022

Daniel Ashbrook & Valkyrie Savage

Introductions

Instructor & course responsible

DANIEL ASHBROOK, PHD

email: dan@di.ku.dk

web: http://danielashbrook.com

BSc, MSc, PhD Computer Science

Georgia Tech 1995–2010

Industrial Research

Nokia	2009–2013
Samsung	2013–2014

Academia

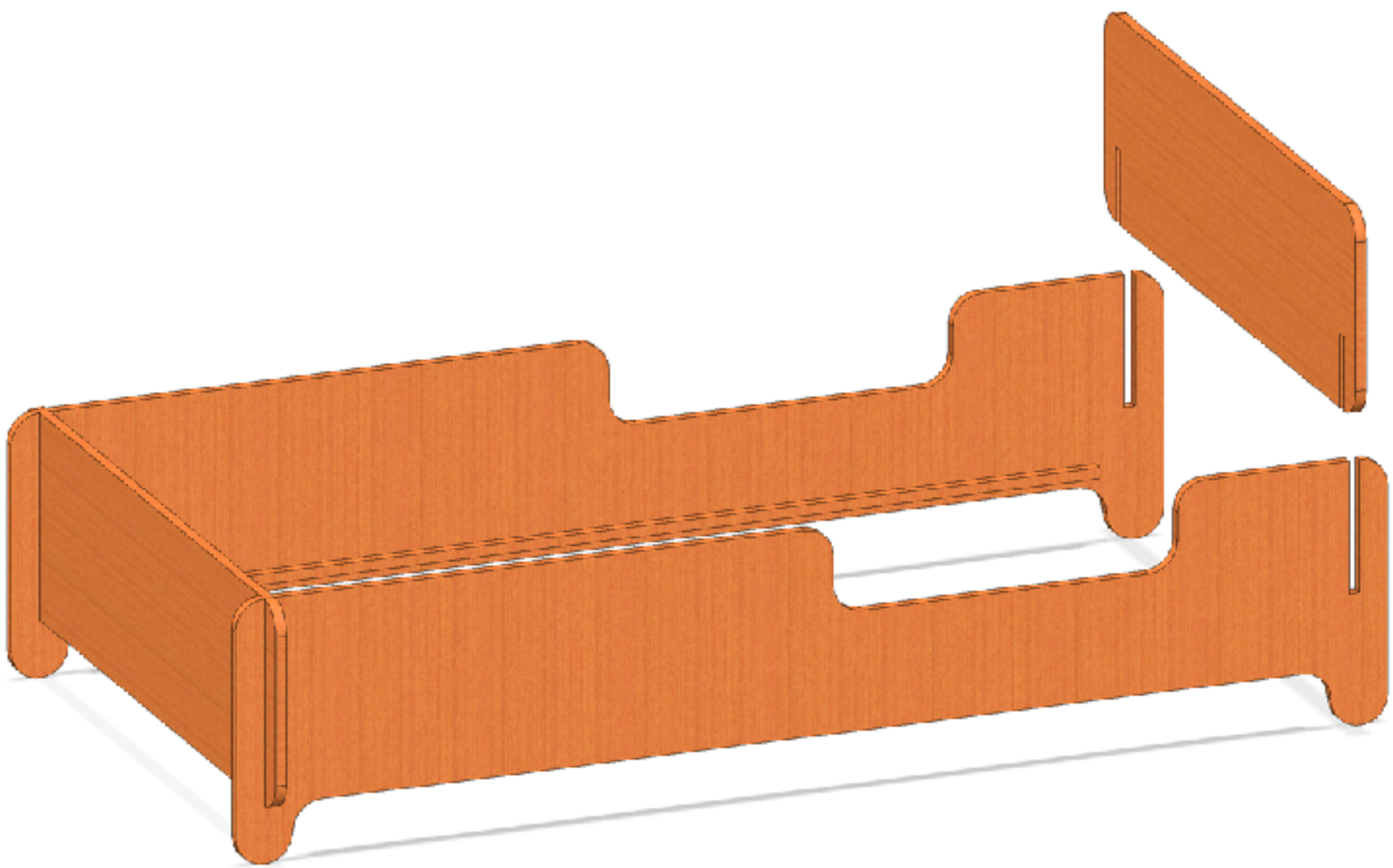
Rochester Institute of Technology	2014–2018
University of Copenhagen	2018–now

Why should you trust me?

I make lots of stuff!

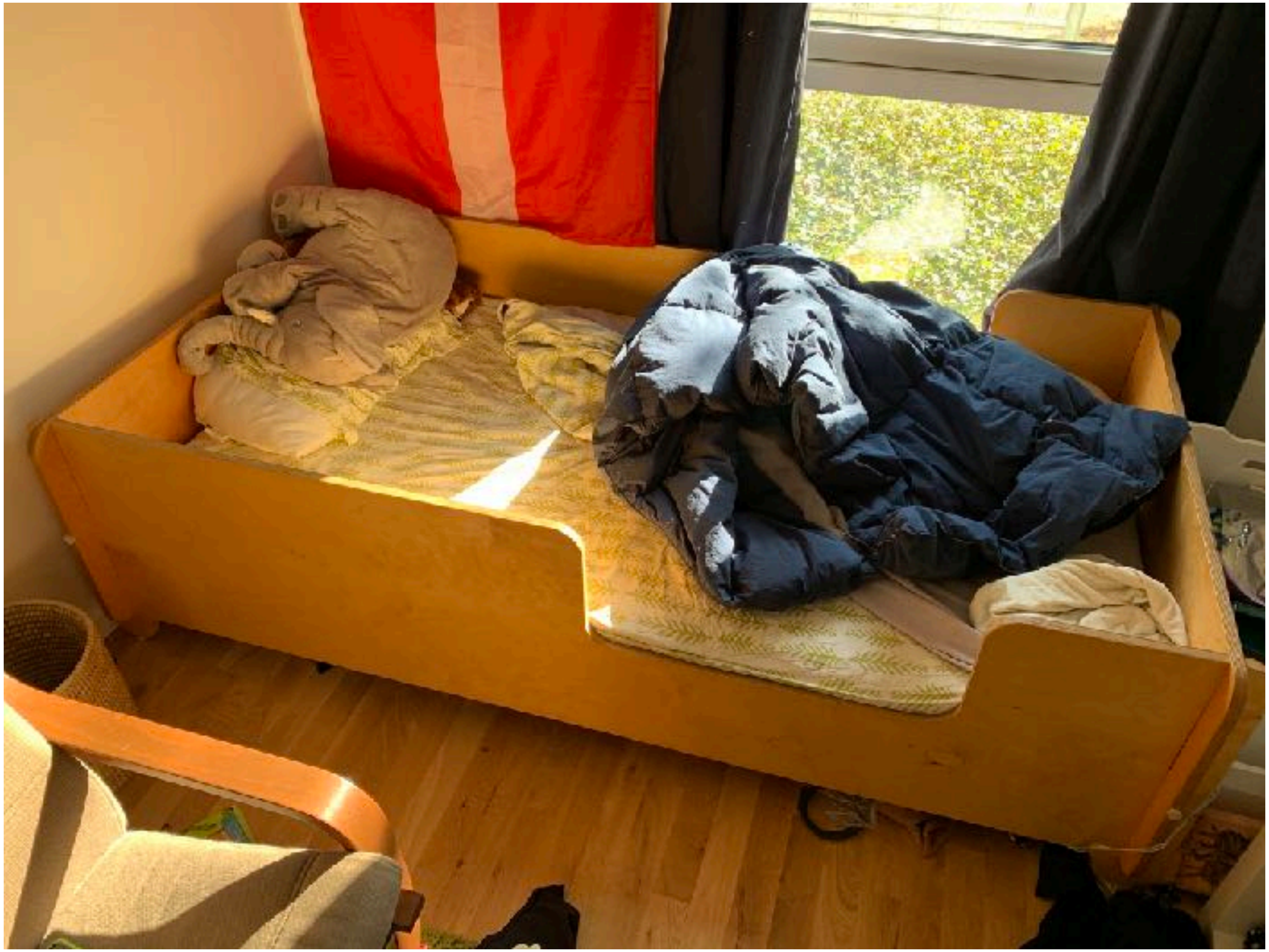
Why should you trust me?

I make lots of stuff!



Why should you trust me?

I make lots of stuff!



Why should you trust me?

I make lots of stuff!

Introductions

Co-instructor

Valkyrie Savage, PhD
email: vasa@di.ku.dk
web: https://valkyriesavage.com

PhD: 2016 UC Berkeley (Computer Science)
BSc: 2010 Indiana University (Computer Science, Math, *Psychology, Spanish*)

Industry
Tactual Labs 2018–2020
My own startup 2016–2018

Academia
Assistant Professor, University of Copenhagen, 2021–now

Why should you trust me?

I make lots of stuff!

Why should you trust me?

I make lots of stuff!

Why should you trust me?

I make lots of stuff!

Why should you trust me?

I make lots of stuff!

Why should you trust yourself?

I’ve taught literal children to do this stuff

Class details

What is this class about?

This is what I started with

in the early 80s.

Now we have these.

And lots of them.


But they are kind of...
boring.

Computation & Fabrication are exploding

into the world around us!

The goal of this course is to teach you how to

participate in this revolution!

Skills you can learn

…and research you might learn about

Technical skills

- 3D design
- 3D printing
- Microcontroller programming
- Working with sensors
- Machine learning
- Soldering
- Basic electronics
- Networking

Research topics

- Fabrication
- Sensing-based UIs
- Shape-change for output
- Tangible UIs
- Haptic output
- Material science
- Soft robotics
- …and more!

After this class…

Maybe you will want to do more?

- You have to do a thesis eventually…
  - Your project can be a great jumping-off point
  - You can also consider a project outside of course scope, related to your project, the class topic, or our research
- Talk to us at any point about these—we love to talk about our research!

Details

Meeting & Communication

- Primary communication method: Slack
  - If possible, send messages via Slack rather than email or Absalon: we will respond more quickly
  - Please add a photo to your profile
- Information, assignments, due dates on Absalon

Details
Workload

Offline work
Break
Lecture
Participation

Category	Hours	Hours/week (approx)
Lectures	24	3
Preparation	32	4
Practical exercises	24	3
Project work	125	16
Exam	1	—
Total	206	26

Details

Rough block schedule

First four weeks:

- intense skill-building
- intense project startup

Second four weeks:

- focus on research topics
- apply skills to projects

Entire course:

- read UIT theory papers
- discuss on Absalon

Details

Rough daily schedule

- Goals
  - Not much lecture: semi-flipped class model
    - Material for you to consume before/during class
    - Work during class
    - We help during class time & via Slack
- **Feedback is super important!**
  → Use Slack

11:00	Lecture
11:15	
11:30	Homework discussion
11:45	
12:00	Lunch break *(+ group work if desired)*
12:15	
12:30	
12:45	
13:00	In-class activity
13:15	
13:30	
13:45	
14:00	Project updates
14:15	Project work & assistance in lab
14:30	
14:45	

Offline work
Break
Lecture
Participation

Group project introduction

Logistics

Group project & exam

Make a cool thing in 8 weeks

Goal: develop a novel fabrication-based interactive technology.

How? Use our fun hardware, (optionally) order parts.

Evaluation: ideally evaluate your technology via studies.

Project video: short (30s) video showcasing project.

Project report: well-written, in the style of a research paper; contributes 60% to grade for course.

Group project

Previous projects

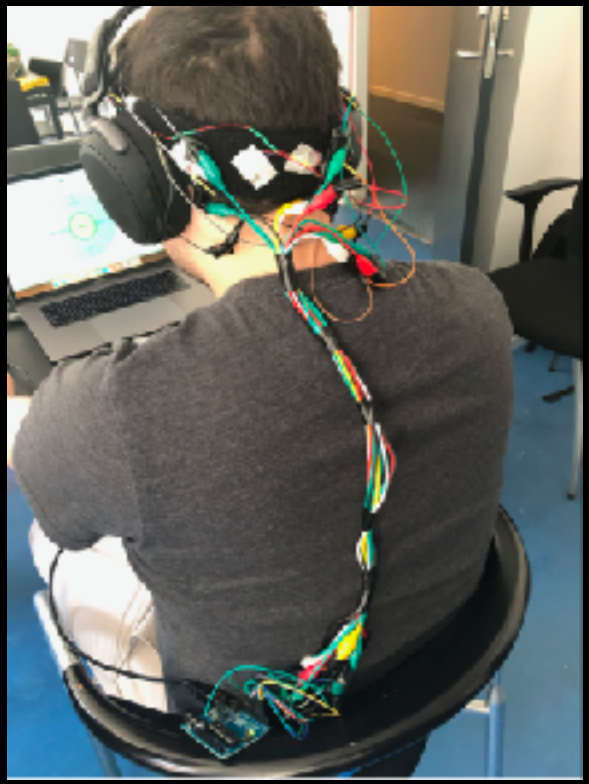
Exploring Sequential Vibration as Haptic Feedback

Inducing A Sense Of Tension Using EMS

MYSTIK: Experimenting with Shape-Shifting E-textile Technologies

Group project

Previous projects



Head-mounted device for enhanced spatial sound localisation

Using the Hand for Multivariable Control

Where Are You Going: Changing Walking Direction with Auditory Feedback and Force Sensing Resistors

Group project

What to do

- Do project-related assignments to find a project—talk to us when you have questions!
- Find some team members. Try to have a variety of skills.
- Figure out what project ideas are appealing to the team and explore them in more depth: what would actually doing the project involve?
- Send me messages on Teams to ask for feedback.
- Write your project outline.

Group project

Forming groups

- Groups should be 3 students/ea, with 1–2 exceptions (see assignment on Absalon)
- Organize yourselves on Slack in **#project_pitches**
- Use your idea exploration assignment to find teammates with similar interests and complimentary skills
- After lunch: everyone describes at least one project idea, we give feedback, you learn about your classmates

Group project

Timeline—tight!

Yesterday	Theme mapping
Tomorrow	Watch videos from classmates
Sunday	Project idea exploration due
Next Tuesday	Form teams
Next Wednesday	Finalize project idea
Next Thursday	Project start!
Tuesday, May 31	Mid-term project report check-in due
Friday, June 17?	Final report and video due

Exam info

Exam forms

The exam consists of two parts:

1. A **group project** developed during the course and documented with a report wherein the individual contributions are stated (**60%**) (written assignment)

2. An **individual** 3-day take-home exam (**40%**) (written exam)

Both parts of the exam must be passed in order to pass the exam, i.e. each part must achieve at least the grade **02**.

Exam & project dates

SCIENCE is making life complicated

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
25 Apr Official class start	26	27 Theme map	28	29	30	1 May Idea exploration
2	3 Choose teams	4 Choose project	5 Project kickoff	6	7	8
9	10	11	12	13 Holiday	14	15
16	17	18	19	20	21	22
23	24 Class 10–12	25	26 Holiday	27	28	29
30	31 Check-in	1 Jun	2	3	4	5
6 Holiday	7	8	9	10	11	12
13	14	15	16	17	18	19
20 Exam week	21	22	23	24	25	26

Exam & project dates

Option 1—everything in last week

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
25 Apr Official class start	26	27 Theme map	28	29	30	1 May Idea exploration
2	3 Choose teams	4 Choose project	5 Project kickoff	6	7	8
9	10	11	12	13 Holiday	14	15
16	17	18	19	20	21	22
23	24 Class 10–12	25	26 Holiday	27	28	29
30	31 Check-in	1 Jun	2	3	4	5
6 Holiday	7	8	9	10	11	12
13	14 Proj reports	15 ← Take-home exam →	16 Presentations	17	18	19
20 Exam week	21	22	23	24	25	26

Exam & project dates

Option 2—project report in exam week

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
25 Apr Official class start	26	27 Theme map	28	29	30	1 May Idea exploration
2	3 Choose teams	4 Choose project	5 Project kickoff	6	7	8
9	10	11	12	13 Holiday	14	15
16	17	18	19	20	21	22
23	24 Class 10–12	25	26 Holiday	27	28	29
30	31 Check-in	1 Jun	2	3	4	5
6 Holiday	7	8	9	10	11	12
13	14	15	16 ← Take-home exam → Presentations	17	18	19
20 Exam week	21	22	23	24 Proj reports	25	26

How to get a good grade

In general

*In order to earn the grade **12**,*
students must demonstrate
*the **knowledge**, **skills** and **competences***
described in the Learning Outcome.

Exam info

How to get a good grade

Knowledge of

- classic and current research in user interface technology;
- models of user interaction and methods of evaluating research in user interface technology; and
- advantages and disadvantages of particular user interface technologies.

Skills in

- locating and understanding the latest research in user interface technology; and
- designing and building software and hardware prototypes of novel user interface technology.

Competences in

- developing novel user interfaces based on an understanding of classic and contemporary research; and
- analyzing uses of technology in user interfaces to describe the pros and cons of the choices made.

Exam info

How to get a good grade

Knowledge of

- classic and current **research** in user interface technology;
- **models** of user interaction and methods of **evaluating** research in user interface technology; and
- **advantages and disadvantages** of particular user interface technologies.

Skills in

- locating and understanding the latest **research** in user interface technology; and
- designing and building software and hardware **prototypes** of novel user interface technology.

Competences in

- **developing** novel user interfaces based on an understanding of classic and contemporary **research**; and
- **analyzing** uses of technology in user interfaces to describe the pros and cons of the choices made.

How to get a good grade

Project report

- Project idea & motivation—why did you do this? ← **advantages/ disadvantages; research**
- Related work—who else tried to solve a similar problem? What other research were you informed by? ← **research**
- Approach—what did you do? How did you do it?← **developing prototypes**
- Results—did it work? How do you know it worked? ← **evaluating; analyzing**
- Discussion—what did you learn? What would you have done differently, or what would you do if you had more time? If you didn't achieve your goals, why? ← **all of the above**

How to get a good grade

Take-home exam

- Written assignment
- What you'll do (subject to slight modifications):
  - you'll get a UIT-related problem
  - discuss the problem ← **models, advantages/disadvantages**
  - find related research ← **research**
  - prototype design of a solution to one challenge ← **develop**
  - discuss solution, possible evaluation ← **research, models, analyzing, evaluating**

How to get a good grade

In general

*In order to earn the grade **12**,*
students must demonstrate
*the **knowledge**, **skills** and **competences***
described in the Learning Outcome.

→ Convince us that you learned these things

→ Be creative and interesting

→ Good presentation helps us understand what you did and what you learned

Group project introduction

How to do it

Group project
Inspiration: UIST

User Interface Software & Technology

- One of the most influential HCI conferences
- Technology-focused
- First was held in 1988; content has evolved with trends
- The session titles alone are inspiring!

Group project

Inspiration: UIST '21

Fabrication

Hand-y Stuff in AR/VR

Summarisation & Semantics

Alternative Programming

Applications in Mixed Reality

Tasks & Tutorials

Fabrics Meet Lasers

Reality: Augmented or Virtual?

Automotive and Proximate Interaction

Brushing, Talking, and Virtual Conferencing

Building Experiences in AR/VR

Efficient Tasks for All

Fluids & Fabrication

Enhancing Complex Interactions

Lasers, Sharks Not Included

Touring AR/VR… Sometimes in Cars

Augmentation & Accessibility

Haptics

Illustration and Information Management

Interacting Across and Through

More Haptics, Now With Modeling

Wrangling Data for Authoring & Analysis

Head-Based Peripheral and Vibrotactile Interactions

Typing and Pointing

Fabrication & Games

Gesture Input and Authoring

Even More Haptics

Healthcare and Information Management

Steering, Pointing, and Recovery

Input & Output

Proximate Interaction and Haptics

Working with Data

Motion Tracking

Pointing and BCI

Understanding and Modifying UI

Exploring and Writing Code

On-Body Interaction

Text Entry and Dealing with Errors

Interaction in the World

Understanding UIs and Managing Epidemics

Interactive Editing and Workflows

Tracking Devices, Eyes, and Fingers

Gestures and Sketching

Illustration, Games, and Accessibility

Tracking and Touch

Device Augmentation and Communication

Live Interactive Programming

More Touch and Other Input Methods

Fabrication and Fabric

Sensing in Various Modalities

Web Programming and Assembly

Group project

Inspiration: UIST ‘21

Fabrication

Hand-y Stuff in AR/VR

Summarisation & Semantics

Alternative Programming

Applications in Mixed Reality

Tasks & Tutorials

Fabrics Meet Lasers

Reality: Augmented or Virtual?

Automotive and Proximate Interaction

Brushing, Talking, and Virtual Conferencing

Building Experiences in AR/VR

Efficient Tasks for All

Fluids & Fabrication

Enhancing Complex Interactions

Lasers, Sharks Not Included

Touring AR/VR… Sometimes in Cars

Augmentation & Accessibility

Haptics

Illustration and Information Management

Interacting Across and Through

More Haptics, Now With Modeling

Wrangling Data for Authoring & Analysis

Head-Based Peripheral and Vibrotactile Interactions

Typing and Pointing

Fabrication & Games

Gesture Input and Authoring

Even More Haptics

Healthcare and Information Management

Steering, Pointing, and Recovery

Input & Output

Proximate Interaction and Haptics

Working with Data

Motion Tracking

Pointing and BCI

Understanding and Modifying UI

Exploring and Writing Code

On-Body Interaction

Text Entry and Dealing with Errors

Interaction in the World

Understanding UIs and Managing Epidemics

Interactive Editing and Workflows

Tracking Devices, Eyes, and Fingers

Gestures and Sketching

Illustration, Games, and Accessibility

Tracking and Touch

Device Augmentation and Communication

Live Interactive Programming

More Touch and Other Input Methods

Fabrication and Fabric

Sensing in Various Modalities

Web Programming and Assembly

Group project

Inspiration: UIST ’21 Position-dependent appearance

Fabrication
Hand-y Stuff
Summarisatic
Alternative Pr
Applications
Tasks & Tutor
Fabrics Mee
Reality: Augm
Automotive a
Brushing, Tal
Building Exp
Efficient Task
Fluids & Fab
Enhancing Co
Lasers, Shark
Touring AR/V
Augmentation
Haptics
Illustration an
Interacting A
More Haptic
Wrangling Da

Lenticular Objects: 3D Printed Objects with Lenticular Lens Surfaces That Can Change their Appearance Depending on the Viewpoint

Jiani Zeng*
MIT
Cambridge, USA
jnzeng@mit.edu

Honghao Deng*
MIT
Cambridge, USA
honghaod@mit.edu

Yunyi Zhu*
MIT CSAIL
Cambridge, USA
yunyizhu@mit.edu

Michael Wessely
MIT CSAIL
Cambridge, USA
wessely@mit.edu

Axel Kilian
MIT Architecture
Cambridge, USA
akilian@mit.edu

Stefanie Mueller
MIT CSAIL
Cambridge, USA
stefanie.mueller@mit.edu

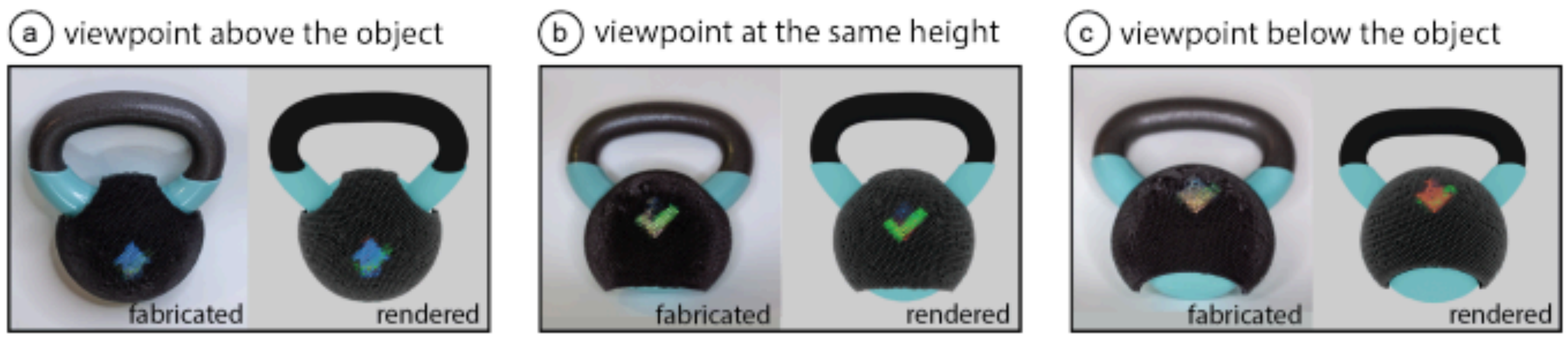


Figure 1: 3D printed objects with lenticular lens surfaces enable viewers to see different appearances from different viewpoints (a: looking down; b: looking horizontally; c: looking up). Our user interface supports designers in setting up different viewpoints and assigning the corresponding textures. On export, it automatically generates the files for fabrication. Designers can then 3D print the object geometry, lenses, and underlying color patterns in a single pass with a multi-material 3D printer.

Group project

Inspiration: UIST '21 Fabricate big objects

Fabrication

Hand-y Stuff in A

Summarisation &

Alternative Progr

Applications in

Tasks & Tutorials

Fabrics Meet La

Reality: Augment

Automotive and

Brushing, Talkin

Building Exper

Efficient Tasks

Fluids & Fabric

Enhancing Con

Lasers, Sharks

Touring AR/VR.

Augmentation &

Haptics

Illustration and

Interacting Acr

More Haptics,

Wrangling Data

Trusscillator: a System for Fabricating Human-Scale Human-Powered Oscillating Devices

Robert Kovacs*
Hasso Plattner Institute, University of Potsdam, Germany
robert.kovacs@hpi.de

Lukas Rambold*
Hasso Plattner Institute, University of Potsdam, Germany
lukas.rambold@student.hpi.de

Lukas Fritzsche
Hasso Plattner Institute, University of Potsdam, Germany
lukas.fritzsche@student.hpi.de

Dominik Meier
Hasso Plattner Institute, University of Potsdam, Germany
dominik.meier@student.hpi.de

Jotaro Shigeyama
Hasso Plattner Institute, University of Potsdam, Germany
jotaro.shigeyama@hpi.de

Shohei Katakura
Hasso Plattner Institute, University of Potsdam, Germany
shohei.katakura@hpi.de

Ran Zhang
Hasso Plattner Institute, University of Potsdam, Germany
ran.zhang@hpi.de

Patrick Baudisch
Hasso Plattner Institute, University of Potsdam, Germany
patrick.baudisch@hpi.de

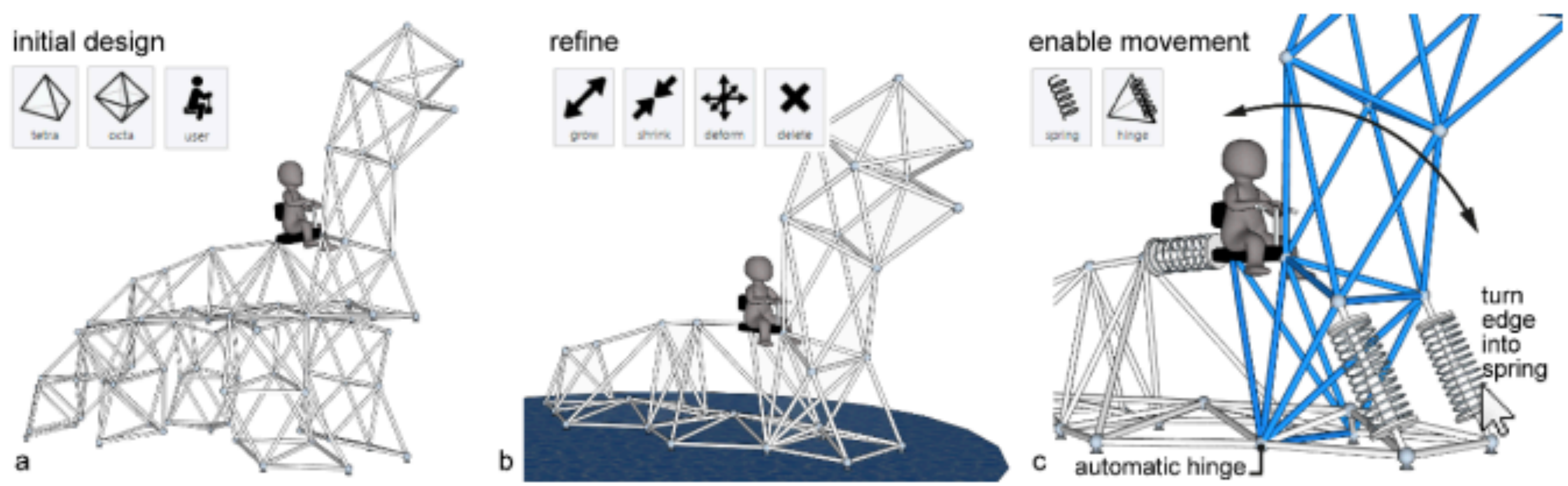


Figure 3: (a) The initial design of the brachiosaurus playground object is created using rigid truss primitives. (b) Designers adjust the shape and lower the height for safety reasons. (c) Using the spring tool, designers enable parts of the model to move. The newly created moving part of the model gets briefly highlighted in blue.

Group project

Inspiration: UIST ’21 Sense with lasers

Fabrication
Hand-y Stuff
Summarisati
Alternative P
Applications
Tasks & Tuto
Fabrics Mee
Reality: Augr
Automotive a
Brushing, Tal
Building Exp
Efficient Task
Fluids & Fab
Enhancing C
Lasers, Shar
Touring AR/V
Augmentatio
Haptics
Illustration ar
Interacting A
More Haptic
Wrangling Da

SensiCut: Material-Aware Laser Cutting Using Speckle Sensing and Deep Learning

Mustafa Doga Dogan
MIT CSAIL
doga@mit.edu

Steven Vidal Acevedo Colon
MIT CSAIL
sacevedo@mit.edu

Varnika Sinha
MIT CSAIL
varnika@mit.edu

Kaan Akşit
University College London
kaksit@cs.ucl.ac.uk

Stefanie Mueller
MIT CSAIL
stefanie.mueller@mit.edu

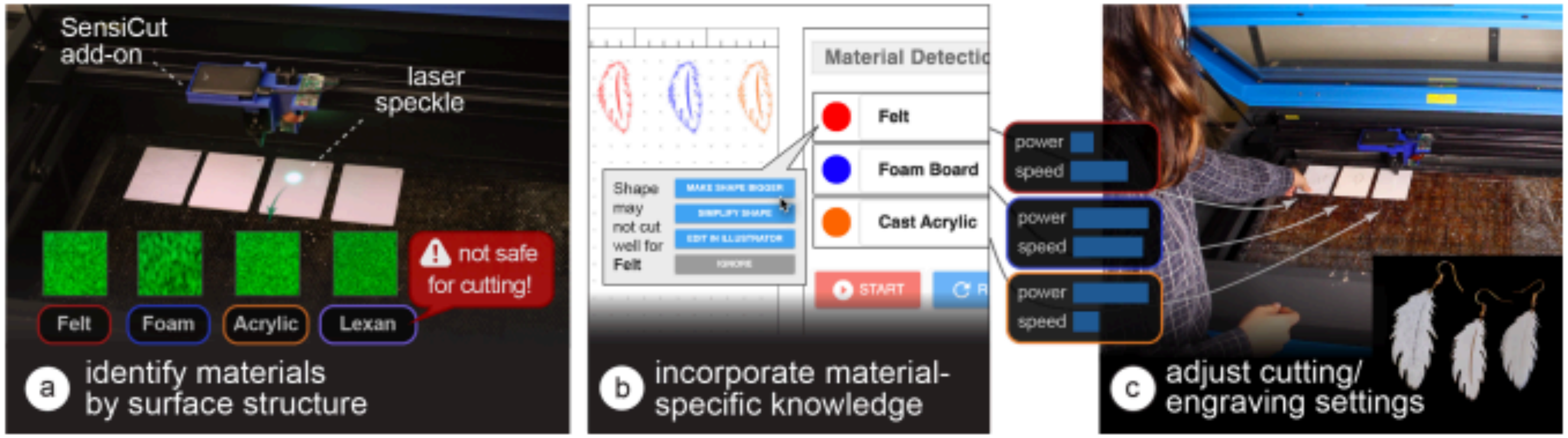


Figure 1: SensiCut augments standard laser cutters with a speckle sensing add-on that can (a) identify materials often found in workshops, including visually similar ones. (b) SensiCut's user interface integrates material identification into the laser cutting workflow and also offers suggestions on how to adjust a design's geometry based on the identified material (e.g., adjusting the size of an earring cut from felt since the kerf for felt is larger than for other materials). (c) Each identified sheet is cut with the correct power and speed settings.

Group project

Inspiration: UIST '21 Support isn't just support

FabHydro: Printing Interactive Hydraulic Devices with an Affordable SLA 3D Printer

Zeyu Yan
Department of Computer Science, University of Maryland, College Park
College Park, MD, USA
zeyuy@umd.edu

Huaishu Peng
Department of Computer Science, University of Maryland, College Park
College Park, MD, USA
huaishu@umd.edu

Figure 1: *FabHydro* overview: a) an off-the-shelf SLA printer with a modified tank and printing plate; b) a complete hydraulic device with a bellows generator and a bending actuator connected with a short piece of tubing; c) a bending actuator is activated by an automatic generator; d) a printed lamp lights up with the change of its posture; e) a phone stand acts as an ambient display when the phone rings.

Fabri
Hand
Sum
Alter
Appl
Tasks
Fabri
Reali
Auto
Brus
Build
Effici
Fluid
Enha
Laser
Touri
Augm
Hapt
Illustr
Interacting Across and Through
More Haptics, Now With Modeling
Wrangling Data for Authoring & Analysis

Group project

Inspiration: UIST ’21 Squeeze more from our sensors

Fabrication
Hand-y Stuff in AR
Summarisation & S
Alternative Prograr
Applications in Mi
Tasks & Tutorials
Fabrics Meet Lase
Reality: Augmentec
Automotive and Pr
Brushing, Talking, a
Building Experien
Efficient Tasks for A
Fluids & Fabricatio
Enhancing Complex
Lasers, Sharks Not I
Touring AR/VR… So
Augmentation & Acc
Haptics
Illustration and Infor
Interacting Across a
More Haptics, Now
Wrangling Data for A

ReflecTrack: Enabling 3D Acoustic Position Tracking Using Commodity Dual-Microphone Smartphones

Yuzhou Zhuang
Department of Computer Science and Technology, Global Innovation Exchange Institute, Tsinghua University
Beijing, China
zhuangyz19@mails.tsinghua.edu.cn

Yuntao Wang*
Department of Computer Science and Technology, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua University
Beijing, China
yuntaowang@tsinghua.edu.cn

Yukang Yan
Department of Computer Science and Technology, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua University
Beijing, China
yyk@mail.tsinghua.edu.cn

Xuhai Xu
Information School, University of Washington
Seattle, Washington, USA
xuhaixu@uw.edu

Yuanchun Shi
Department of Computer Science and Technology, Key Laboratory of Pervasive Computing, Ministry of Education, Tsinghua University
Beijing, China
shiyc@tsinghua.edu.cn

Figure 1: ReflecTrack enables 3D acoustic positioning for commodity dual-microphone smartphones by introducing a nearby reflective surface and developing echo-aware Frequency-Modulated-Continuous-Wave (FMCW). Applications include: (a)

See all UIST ’21 videos

tinyurl.com/uist2021videos

Group project

How do I develop an interesting project?

1. **What is the problem or opportunity?**
   Describe a real-world problem, or a future opportunity that might be enabled by emerging technology.

2. **Why is it worth working on?**
   The motivation for solving the problem or taking advantage of the opportunity.

3. **What is the solution?**
   A short and simple general approach.

4. **What are the steps in the solution?**
   A list breaking down how you will approach the solution.

5. **What is the evaluation?**
   How will you know if your solution worked? Did it solve the problem? Does it work better than other methods?

Group project

How do I develop an interesting project?

1. **What is the problem or opportunity?**
   Making 3D-printed objects interactive by integrating electronics requires many iterations to get it right, which is time-consuming. Embedding electronics makes it difficult to recycle objects.

2. **Why is it worth working on?**
   Printing interactive objects without integrated electronics could be much faster and less-wasteful, leading to easier, environmentally friendly customized objects.

3. **What is the solution?**
   I propose making fully printable sensing, computation, and actuation as an integrated part of a 3D printed object. I will do this by creating shaped channels and forcing air through them.

4. **What are the steps in the solution?**
   I need to develop structures that enable these capabilities and ensure they are functional and printable. This will require 3D design skills, 3D printing, and a plan to sense if they are functional.

5. **What is the evaluation?**
   I will create demo objects based on these structures, and I will quantify how others could use these structures in their own work in the future through a series of airflow tests.

AirLogic:

Embedding Pneumatic Computation and I/O in 3D Models to Fabricate Stand-Alone, Interactive Objects

Intro to UIT theory

How and why to reason about UIs?

How to reason about UIs?

- Today: various theories and ideas about UI(T)
  - Direct manipulation
  - Post-WIMP UIs
  - Reality-Based Interaction
  - Instrumental Interaction
  - Challenges

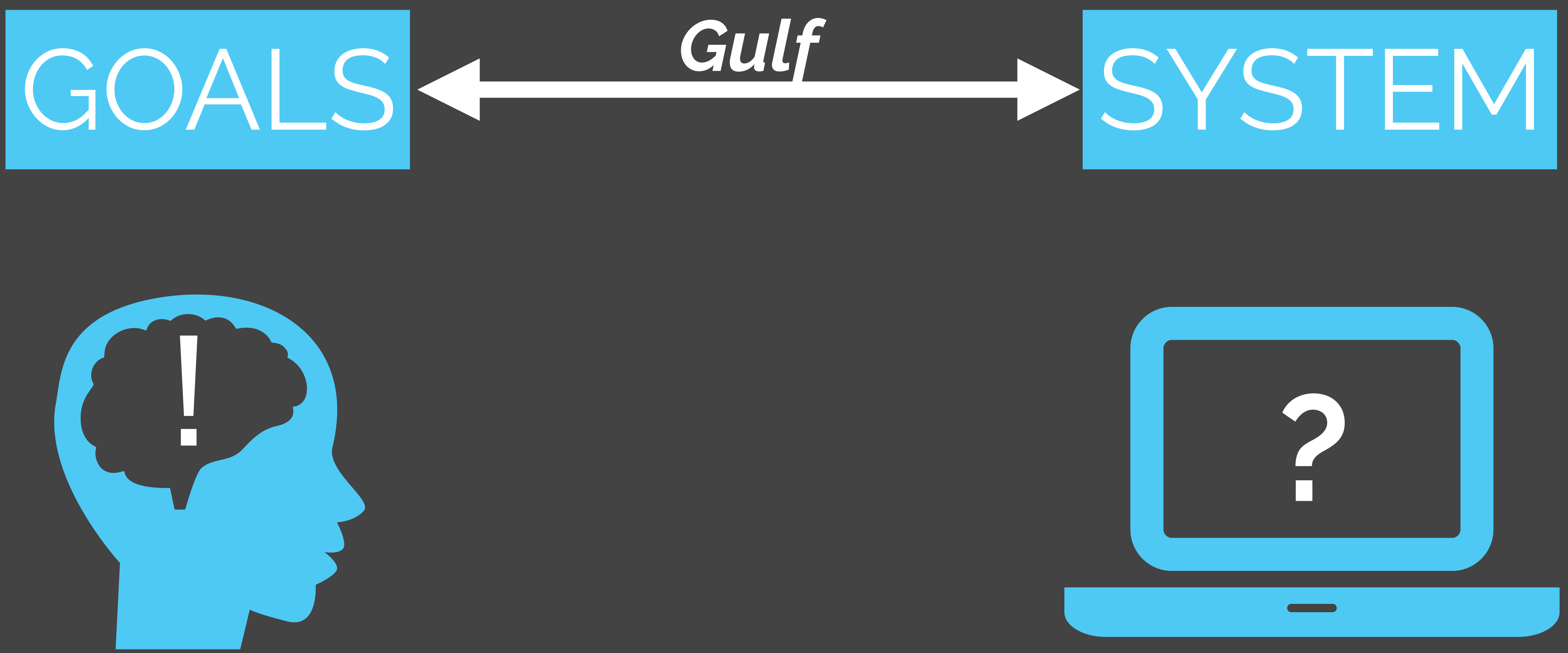
Why reason about UIs?

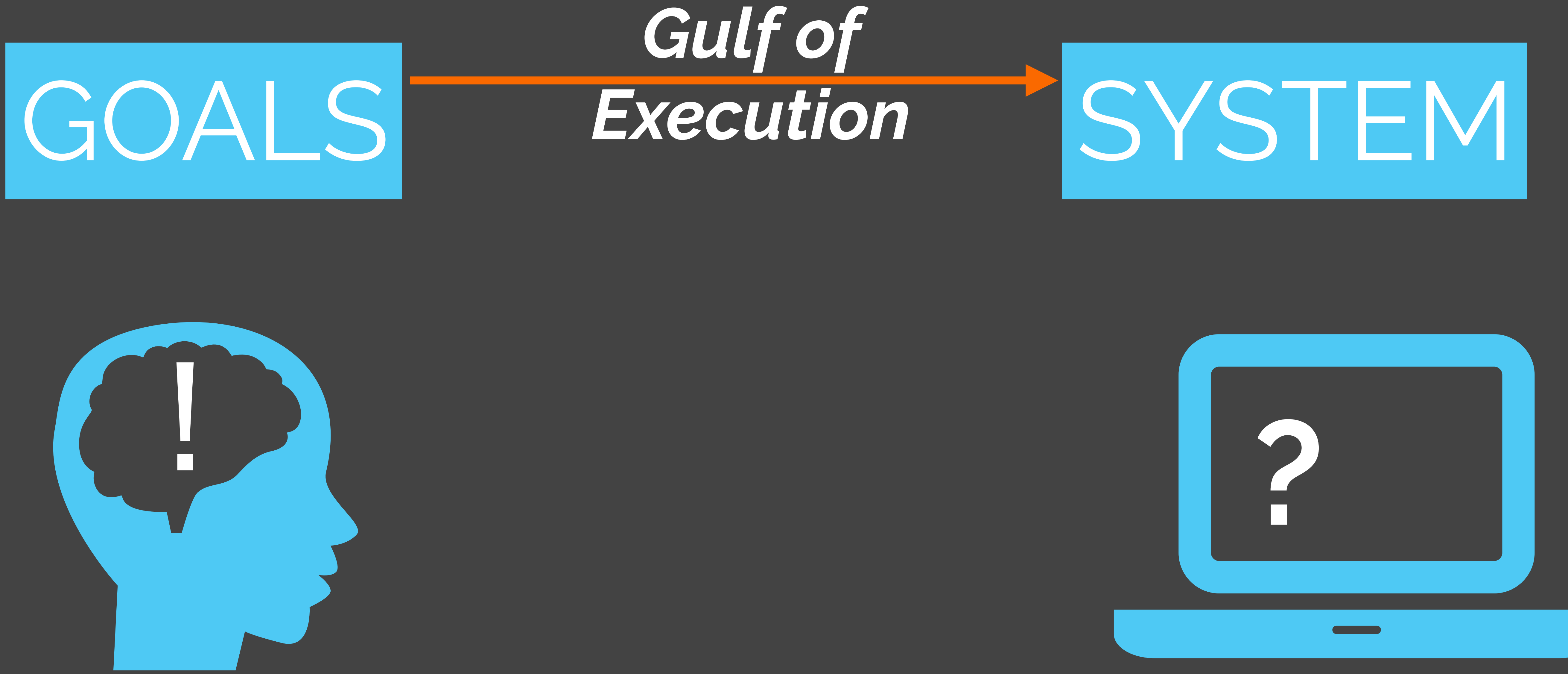
- Why is one interface better than another?
    - (Not just “how”, but “*why*”)
- What might work for a new UI?
    - Learn by analogy to other kinds of UIs
    - Pro tip: think about how to apply screen-based UI paradigms to physical & fabricated UIs
- Inspiration!

Direct manipulation

or, “Why are GUIs easy to use?”

Direct manipulation


GOALS
Gulf
SYSTEM
!
?

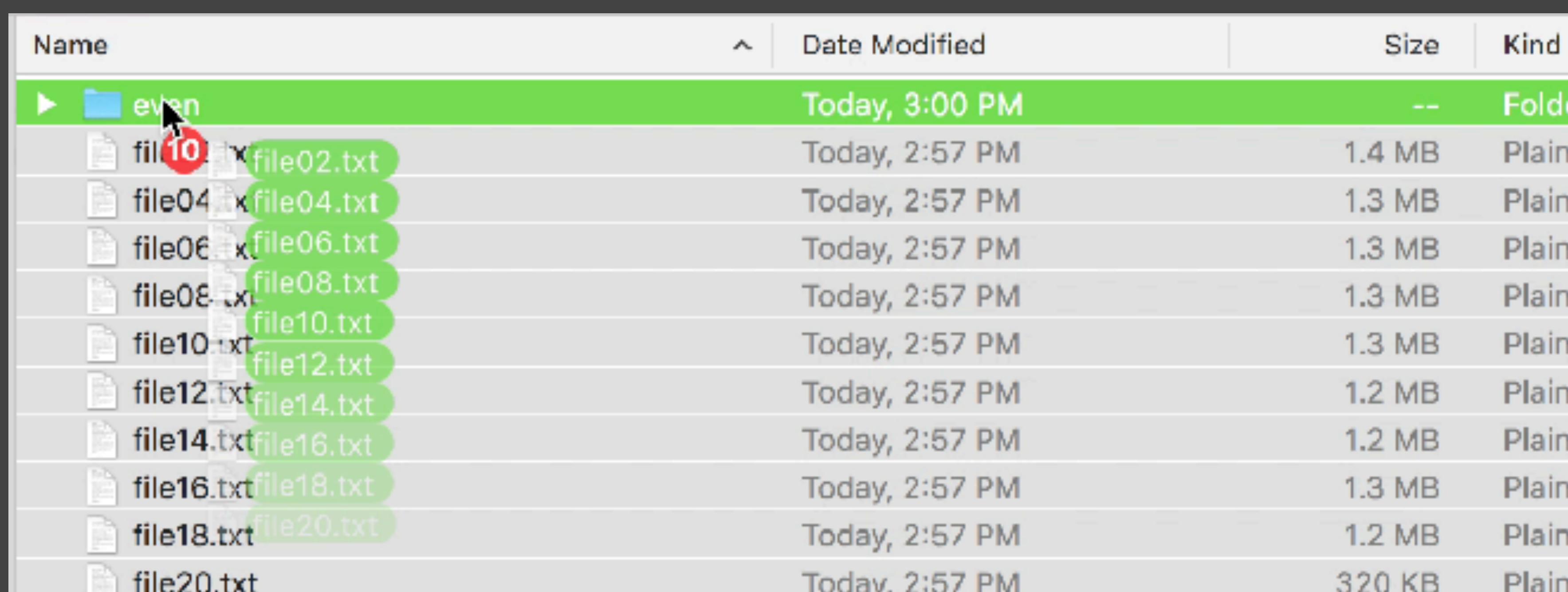

GOALS
Gulf of Execution
SYSTEM
!
?

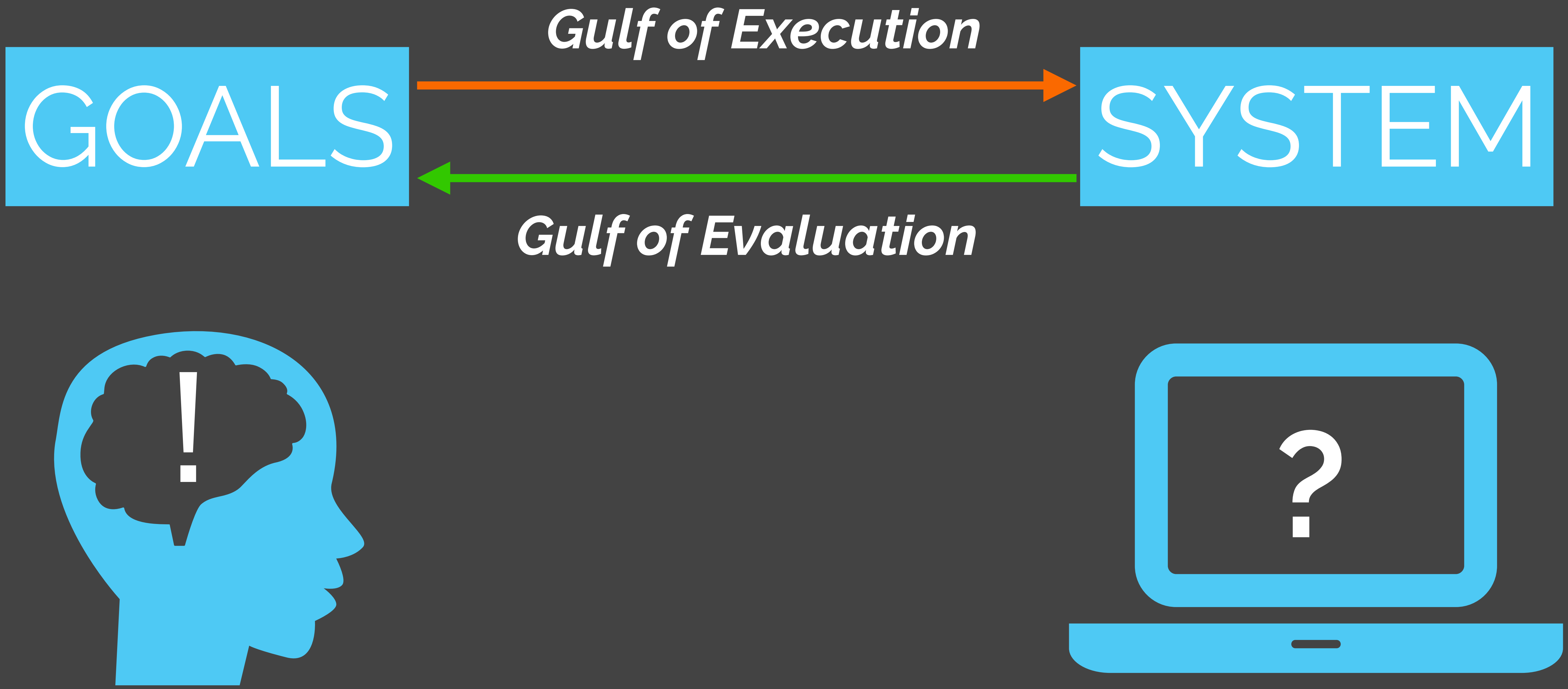
1. Remember commands: `ls`, `mkdir`, `mv`
2. Repeat information to system (filenames)

```
dlaics:/tmp/demo>ls
file01.txt      file05.txt      file09.txt      file13.txt      file17.txt
file02.txt      file06.txt      file10.txt      file14.txt      file18.txt
file03.txt      file07.txt      file11.txt      file15.txt      file19.txt
file04.txt      file08.txt      file12.txt      file16.txt      file20.txt
dlaics:/tmp/demo>mkdir even
dlaics:/tmp/demo>mkdir odd
dlaics:/tmp/demo>mv file01.txt file03.txt file05.txt file07.txt file09.txt file1
1.txt file13.txt file15.txt file17.txt file19.txt odd
```

GOALS → *Gulf of Execution* → SYSTEM

1. No commands to remember
2. Operate directly on “objects” rather than representations




Gulf of Execution
GOALS
SYSTEM
Gulf of Evaluation
!
?

GOALS ← SYSTEM

Gulf of Evaluation

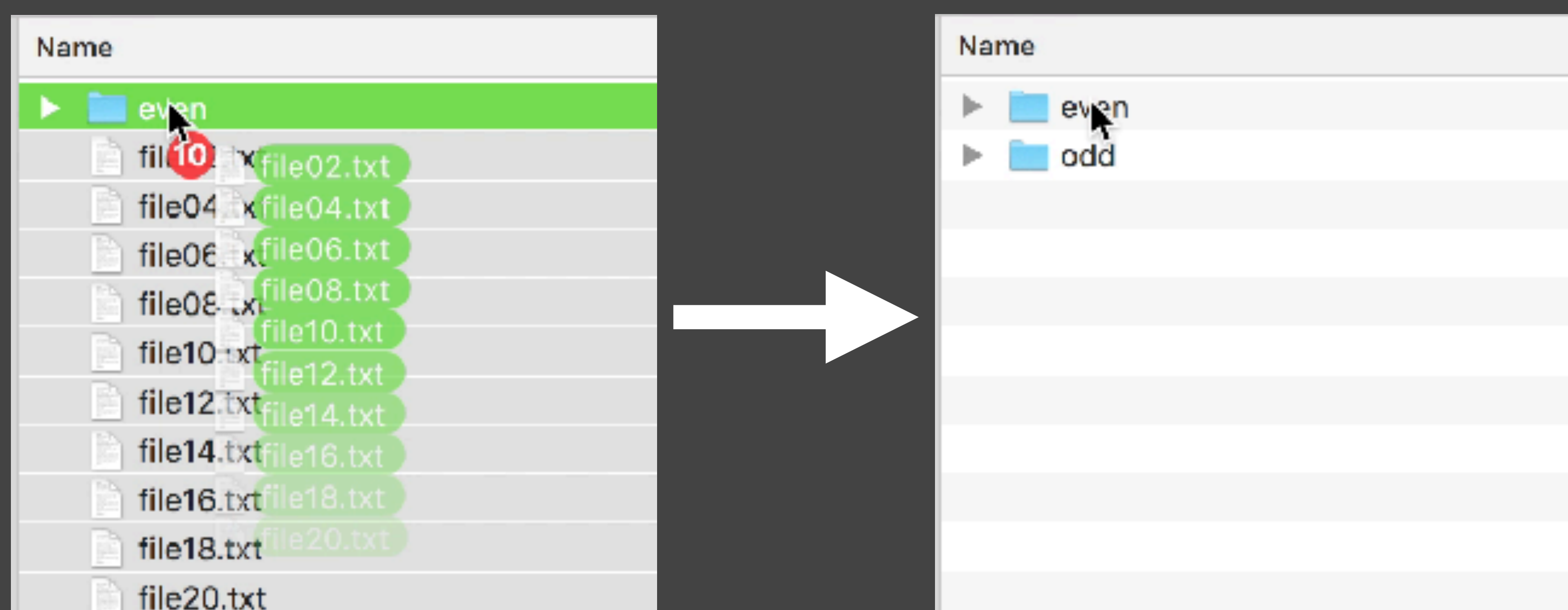
- Verify success

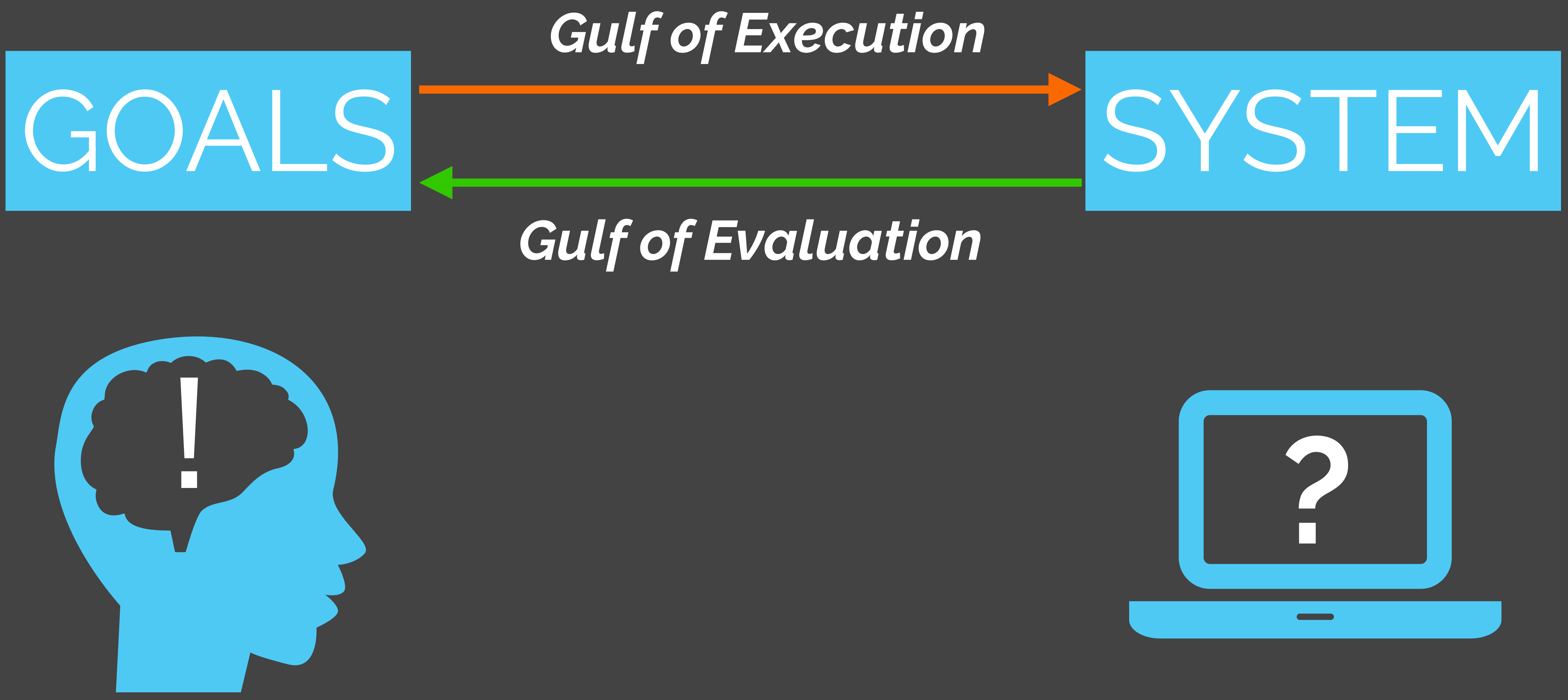
```
dlaics:/tmp/demo>mv file01.txt file03.txt file05.txt file07.txt file09.txt file1
1.txt file13.txt file15.txt file17.txt file19.txt odd
dlaics:/tmp/demo>ls
even/           file06.txt      file12.txt      file18.txt
file02.txt      file08.txt      file14.txt      file20.txt
file04.txt      file10.txt      file16.txt      odd/
dlaics:/tmp/demo>
```

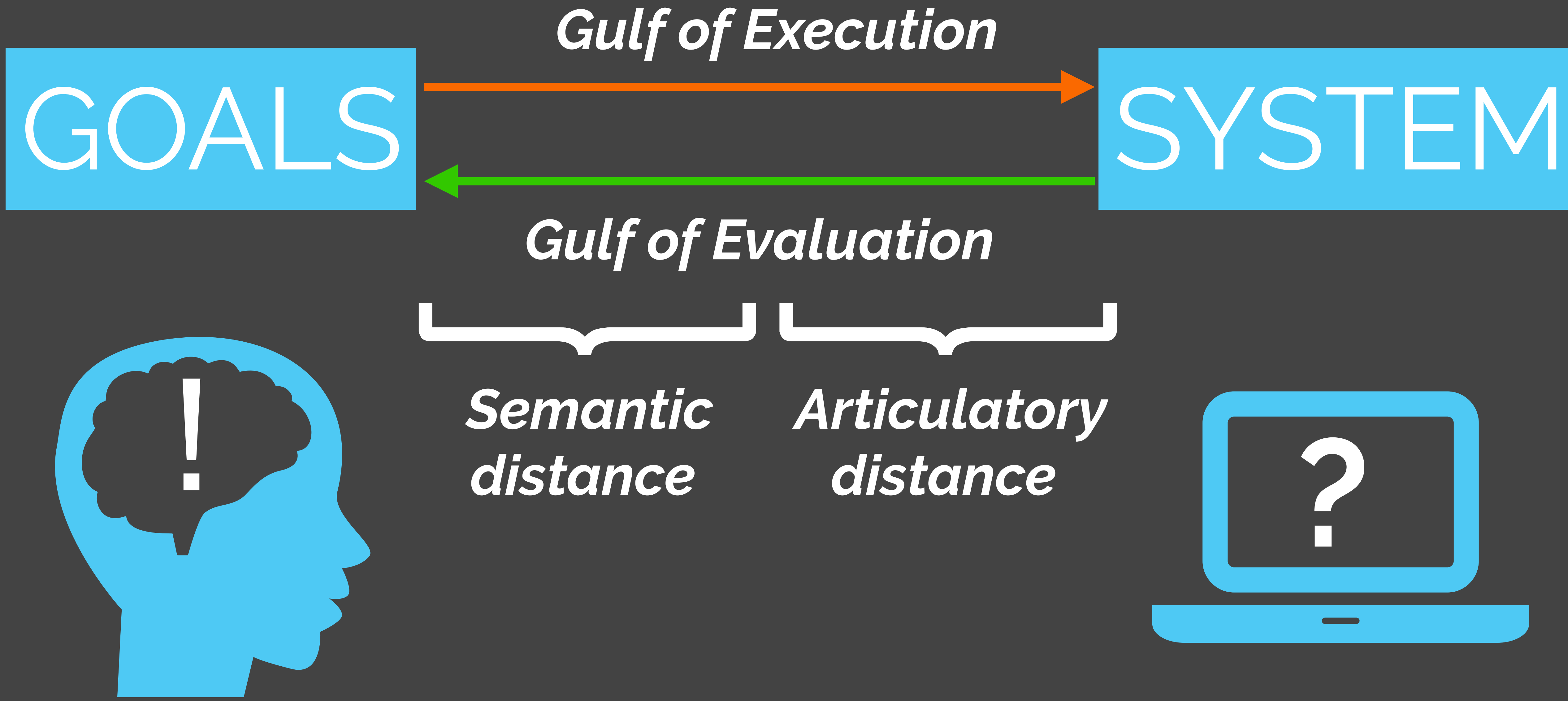
GOALS ← SYSTEM

Gulf of Evaluation

- Results are immediately visible




Gulf of Execution
GOALS
SYSTEM
Gulf of Evaluation
!
?


Gulf of Execution
GOALS
SYSTEM
Gulf of Evaluation
Semantic distance
Articulatory distance
!
?

Semantic distance:

Is it possible to:

- *say what I what I want to say;*
- *say it concisely?*

```
    global  _start
    section .text
_start:
    mov     rax, 1
    mov     rdi, 1
    mov     rsi, message
    mov     rdx, 13
    syscall
    mov     eax, 60
    xor     rdi, rdi
    syscall
message:
    db      "Hello, World", 10
```

```
print('Hello world')
```

Semantic distance:

Is it possible to:

- *say what I what I want to say;*
- *say it concisely?*

Gulf of Execution

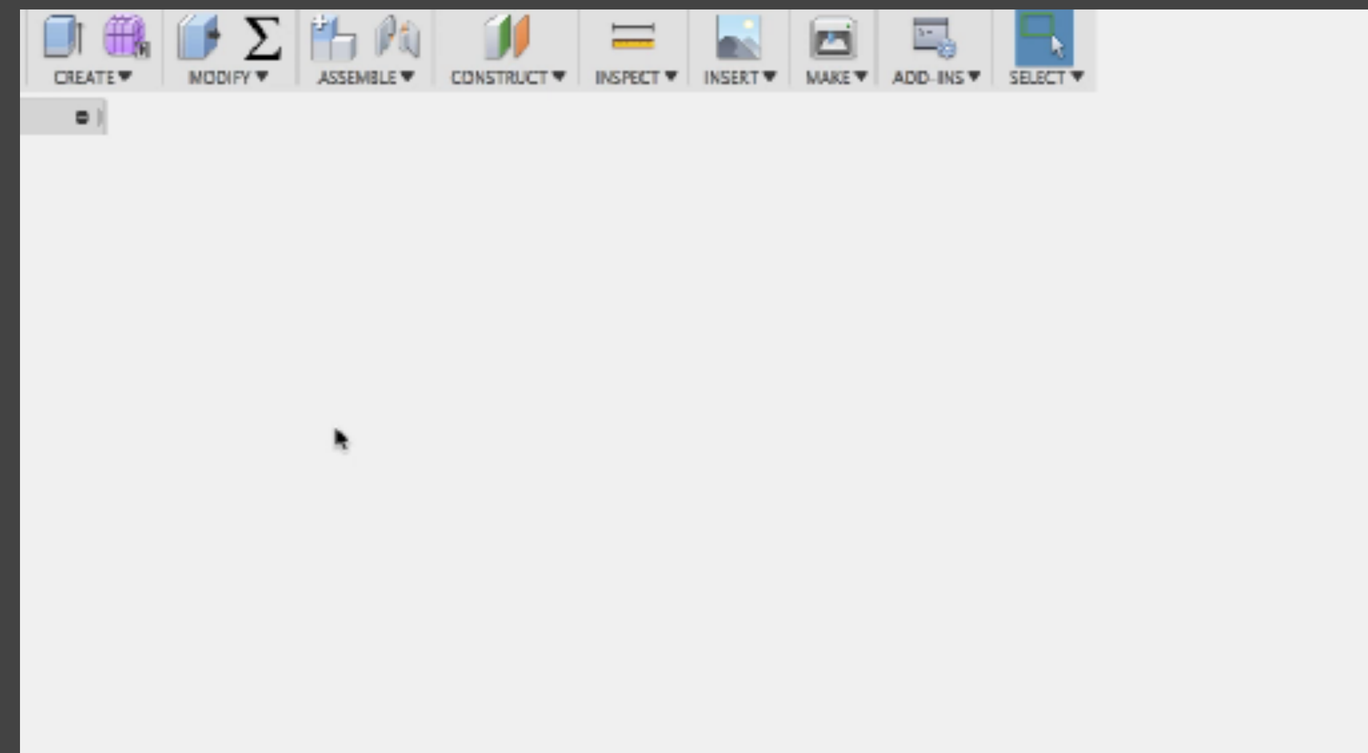
```
module elliptic_ring(r1 = 10, r2 = 5, r = 2,
slices = 100, h = 0, w = 360)
{
  dz = h/slices;
  dwx = atan(h/(r1+r2)/PI);
  for (i = [0:slices-1])
    hull()
    {
      translate([r1*cos((i+1)*w/slices),
r2*sin((i+1)*w/slices), (i+1)*dz])
        rotate([90+dwx, 0, (i+1)*w/slices])
          cylinder(r = r, h = .01);

      translate([r1*cos((i)*w/slices),
r2*sin((i)*w/slices), i*dz])
        rotate([90+dwx, 0, i*w/slices])
          cylinder(r = r, h = .01);
    }
}


module elliptic_spring(Windings = 5, R1 = 20,
R2 = 0, r=1, h = 20, slices = 50)
{
   dh = h/Windings;
   r2 = (R2==0)?R1:R2;
  for(j = [0:Windings-1])
    translate([0, 0, j*dh])
      elliptic_ring(R1, r2, r, slices, dh);
}

module spring(Windings = 5, R = 20, r=2, h =
20, slices = 50)
  elliptic_spring(Windings, R, 0, r, h,
slices);

spring()
```



Semantic distance:

Is it possible to:

- *say what I what I want to say;*
- *say it concisely?*

Gulf of Execution

```latex
\begin{table}[hbt]
  \centering
  \caption
  {
    Devices carried by participants and their location.  ``Body'' refers to on-body placement of a
    device, such as clipping to clothing or using a wrist strap.
    ``Bag'' includes purses and backpacks.
  }
  \label{table:stats_participants}
  \begin{tabular}{|l|c|c|c|c|c|} \hline
                & Pocket & Bag & Hip & Body & \textit{Total} \\ \hline \hline
    Mobile phone  & 7      & 4   & 1   & ---  & 12     \\ \hline
    Phone headset & 4      & --- & --- & ---  & 4      \\ \hline
    Audio player  & 4      & 5   & 1   & 1    & 11     \\ \hline
    Camera        & 6      & 5   & --- & 1    & 12     \\ \hline
    \textit{Total}& 21     & 14  & 2   & 2    & 39     \\ \hline
  \end{tabular}
\end{table}
```

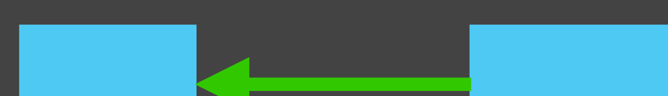
Table 1: Devices carried by participants and their location. "Body" refers to on-body placement of a device, such as clipping to clothing or using a wrist strap. "Bag" includes purses and backpacks.

	Pocket	Bag	Hip	Body	*Total*
Mobile phone	7	4	1	—	12
Phone headset	4	—	—	—	4
Audio player	4	5	1	1	11
Camera	6	5	—	1	12
Total	21	14	2	2	39

Semantic distance:

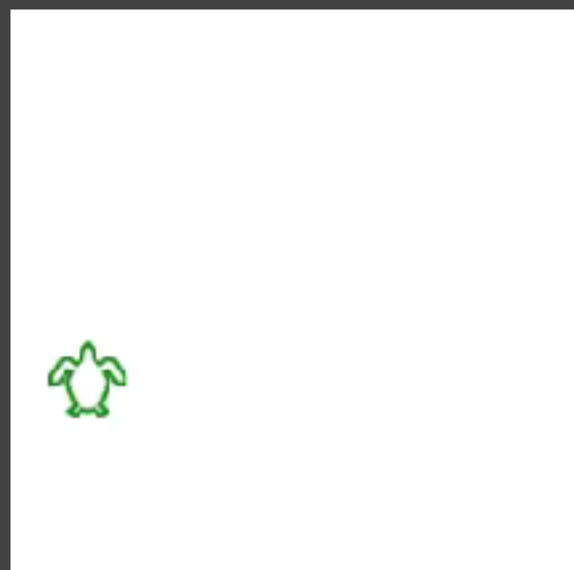
Is it possible to:

- *say what I what I want to say;*
- *say it concisely?*

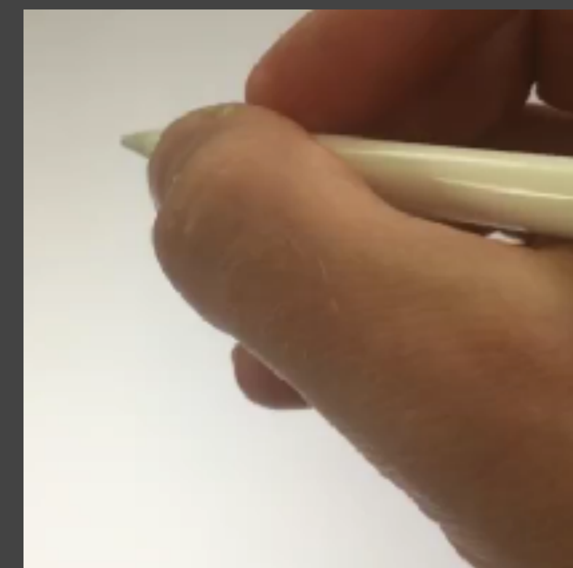


Gulf of Evaluation

Task: draw a square



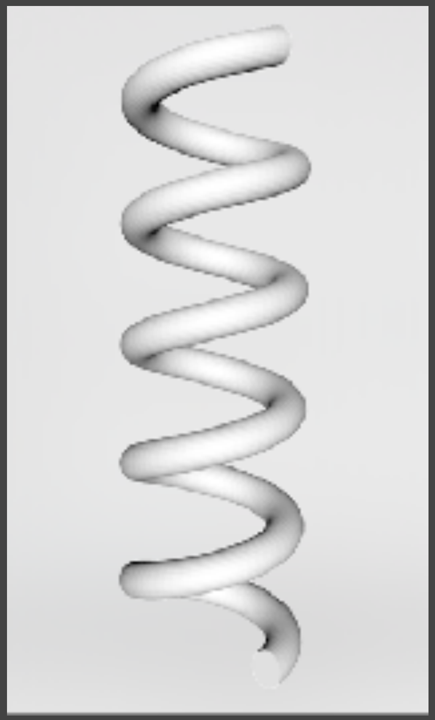
```
repeat 4 [ forward 100 right 90 ]
```



Articulatory distance: Similarity of form of [input] expression to my task

Gulf of Execution

```
solid
  facet normal 9.931888E-01 -1.043883E-01 -5.175938E-02
    outer loop
      vertex 1.250000E+01 -3.061617E-16 0.000000E+00
      vertex 1.244537E+01 -5.197792E-01 0.000000E+00
      vertex 1.247648E+01 1.562500E-01 -7.665092E-01
    endloop
  endfacet
  facet normal 9.880498E-01 -1.332693E-01 -7.743974E-02
    outer loop
      vertex 1.247648E+01 1.562500E-01 -7.665092E-01
      vertex 1.244537E+01 -5.197792E-01 0.000000E+00
      vertex 1.238287E+01 -2.941009E-01 -1.185764E+00
    endloop
  endfacet
  facet normal 9.885426E-01 -1.012378E-01 -1.119570E-01
    outer loop
      vertex 1.247648E+01 1.562500E-01 -7.665092E-01
      vertex 1.238287E+01 -2.941009E-01 -1.185764E+00
      vertex 1.240599E+01 3.125000E-01 -1.530133E+00
    endloop
  endfacet
  facet normal 9.761579E-01 -1.341036E-01 -1.706810E-01
    outer loop
      vertex 1.240599E+01 3.125000E-01 -1.530133E+00
      vertex 1.238287E+01 -2.941009E-01 -1.185764E+00
      vertex 1.221090E+01 -5.093982E-02 -2.360338E+00
    endloop
  endfacet
  facet normal 9.767817E-01 -1.219231E-01 -1.761599E-01
    outer loop
      vertex 1.240599E+01 3.125000E-01 -1.530133E+00
      vertex 1.221090E+01 -5.093982E-02 -2.360338E+00
      vertex 1.228882E+01 4.687500E-01 -2.287999E+00
    endloop
  endfacet
  facet normal 9.657287E-01 -1.122084E-01 -2.340457E-01
    outer loop
```



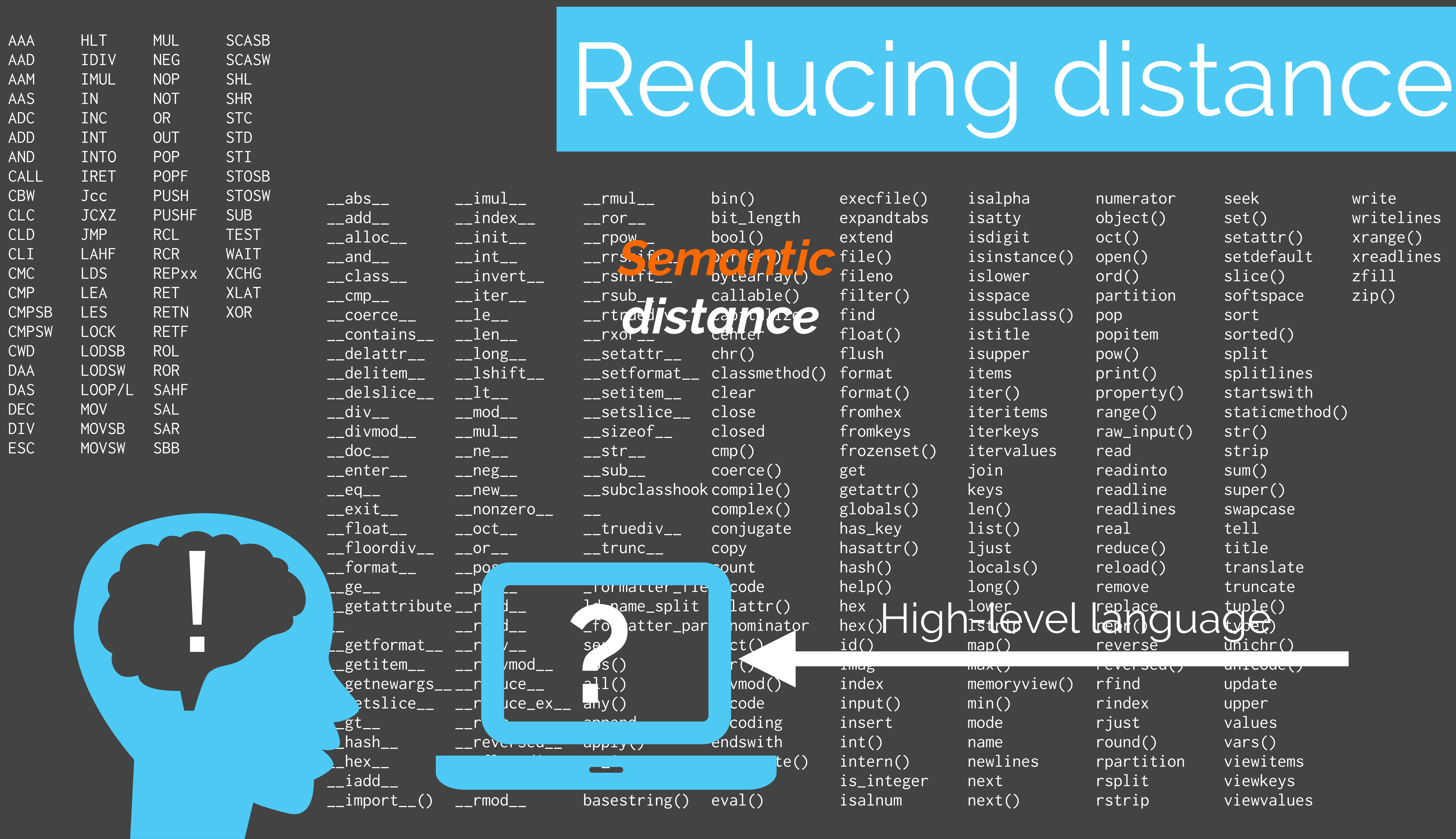
Articulatory distance: Similarity of form of [output] expression to my task

Gulf of Evaluation

Reducing distance

Reducing distance

Semantic
distance


Reducing distance
AAA AAD AAM AAS ADC ADD AND CALL CBW CLC CLD CLI CMC CMP CMPSB CMPSW CWD DAA DAS DEC DIV ESC
HLT IDIV IMUL IN INC INT INTO IRET Jcc JCXZ JMP LAHF LDS LEA LES LOCK LODSB LODSW LOOP/L MOV MOVSB MOVSW
MUL NEG NOP NOT OR OUT POP POPF PUSH PUSHF RCL RCR REPxx RET RETN RETF ROL ROR SAHF SAL SAR SBB
SCASB SCASW SHL SHR STC STD STI STOSB STOSW SUB TEST WAIT XCHG XLAT XOR
__abs__ __add__ __alloc__ __and__ __class__ __cmp__ __coerce__ __contains__ __delattr__ __delitem__ __delslice__ __div__ __divmod__ __doc__ __enter__ __eq__ __exit__ __float__ __floordiv__ __format__ __getformat__ __getitem__ __getnewargs__ __import__()
__imul__ __index__ __init__ __int__ __invert__ __iter__ __le__ __len__ __long__ __lshift__ __lt__ __mod__ __mul__ __ne__ __neg__ __new__ __nonzero__ __oct__ __or__ __rmod__
__rmul__ __ror__ __rpow__ __rshift__ __rsub__ __setattr__ __setformat__ __setitem__ __setslice__ __sizeof__ __str__ __sub__ __subclasshook__ __truediv__ __trunc__ basestring()
bin() bit_length bool() bytearray() callable() center chr() classmethod() clear close closed cmp() coerce() compile() complex() conjugate copy eval()
execfile() expandtabs extend file() fileno filter() find float() flush format format() fromhex fromkeys frozenset() get getattr() globals() has_key hasattr() hash() help() hex hex() id() index input() insert int() intern() is_integer isalnum
isalpha isatty isdigit isinstance() islower isspace issubclass() istitle isupper items iter() iteritems iterkeys itervalues join keys len() list() ljust locals() long() map() memoryview() min() mode name newlines next next()
numerator object() oct() open() ord() partition pop popitem pow() print() property() range() raw_input() read readinto readline readlines real reduce() reload() remove reverse rfind rindex rjust round() rpartition rsplit rstrip
seek set() setattr() setdefault slice() softspace sort sorted() split splitlines startswith staticmethod() str() strip sum() super() swapcase tell title translate truncate tuple() unichr() update upper values vars() viewitems viewkeys viewvalues
write writelines xrange() xreadlines zfill zip()
Semantic distance
!
?
High-level language

Reducing distance

Semantic distance

Reducing distance

Command mode
- Movement
  - hjkl: ←↓↑→
  - w/W: word forward
  - b/B: word back
  - e/E: to end of word
  - ^B: page up
  - ^F: page down
  - /: search
  - ?: reverse search
  - n/N: next/prev result
  - {}: paragraph down/up
  - (): sentence down/up
  - t<character>: move to before character
  - f<character>: move to character
- Changes
  - d<movement>: delete
  - y<movement>: copy
  - p: paste
  - D: delete to end of line
  - x: delete character
  - s: delete and insert character
  - a: insert mode after current character
  - i: insert mode at current character

***Semantic* distance**

Reducing distance

***Articulatory* distance**

Direct manipulation

Benefits

- Easy to learn for novices—often through demonstration
- Experts can work rapidly
- Intermittent users can remember basic concepts
- Reduced need for error messages
- Easy to see if activity is having the desired effect

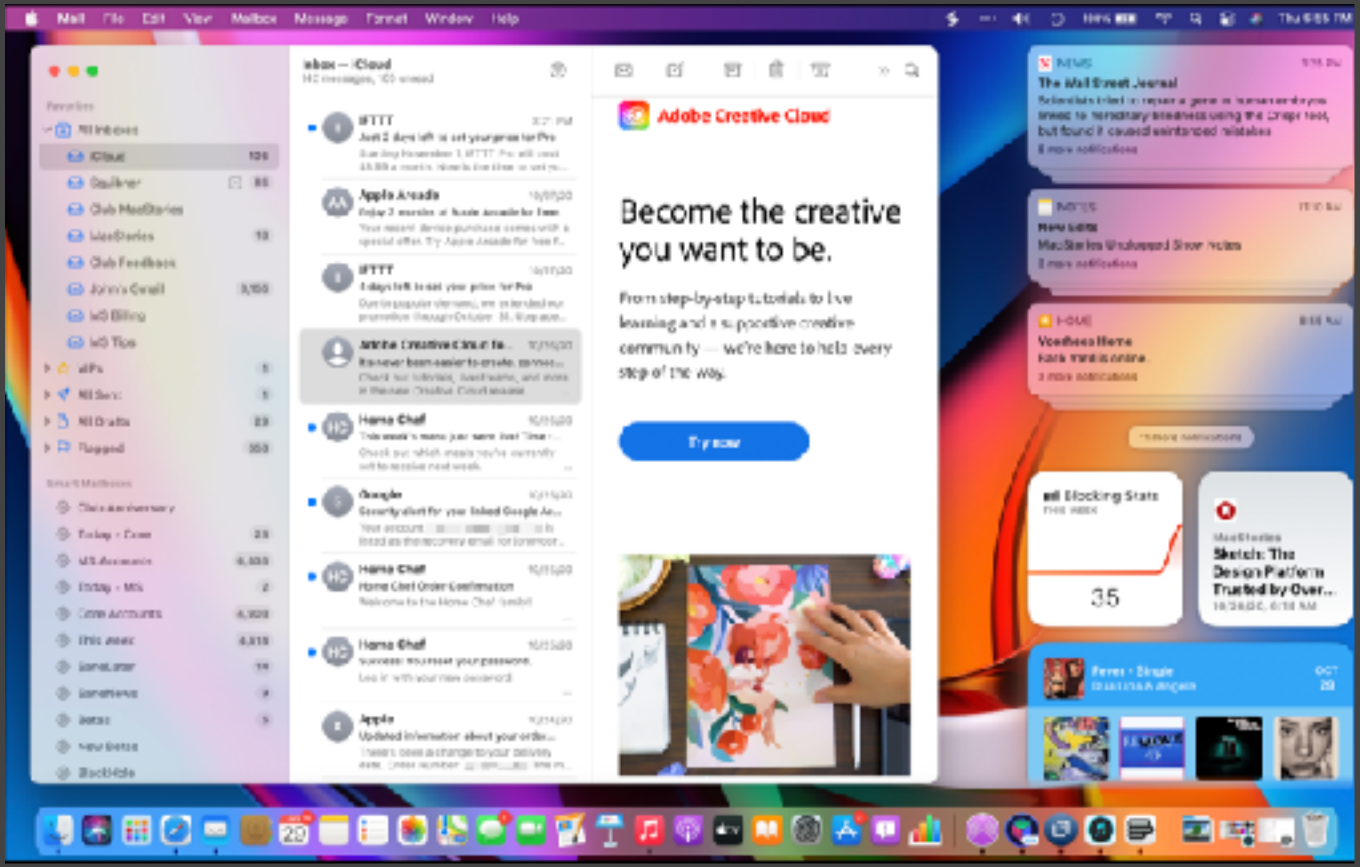
Drawbacks

- Repetitive operations can be slow
- Can only act on visible objects
- Lack of error messages can be frustrating
- Precision can sometimes be difficult

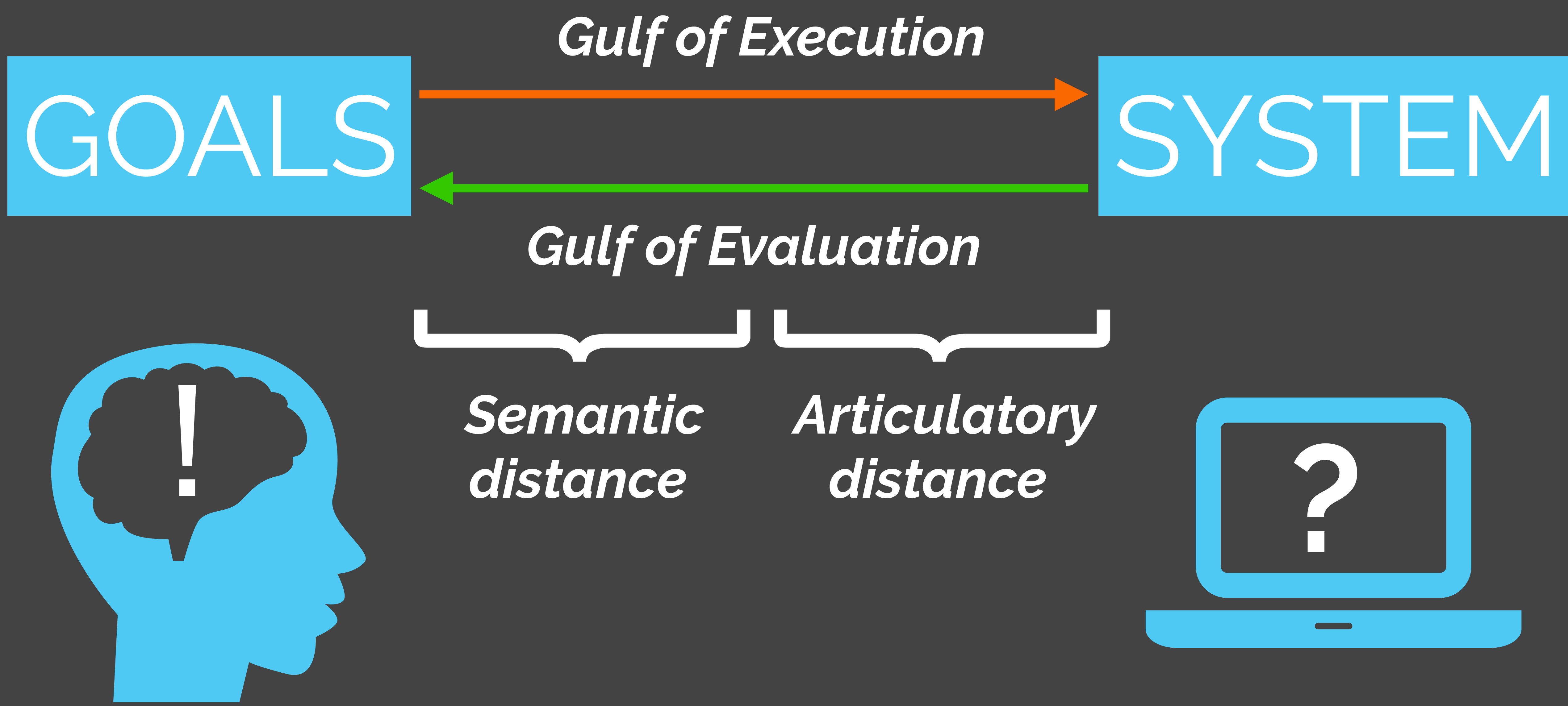
Direct manipulation

Properties

- Continuous representation of objects of interest
- "Physical" actions on objects rather than complex syntax
- Fast, incremental, and reversible operations
    - …with immediate effect on objects
- Smooth approach to learning



Direct manipulation



What comes after GUIs?

Post-WIMP UIs

Towards the "ideal" UI

"The user wants to focus on the task,
not [on] the technology for
specifying the task."
Van Dam, A. (1997). Post-WIMP user interfaces.

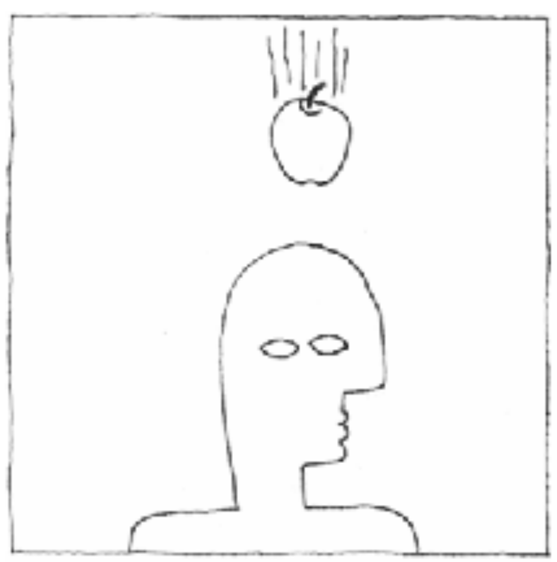
"Natural" interactions

- Speech
- Gesture
- Physical manipulation
- Intuitive physical output

Reality-based interaction

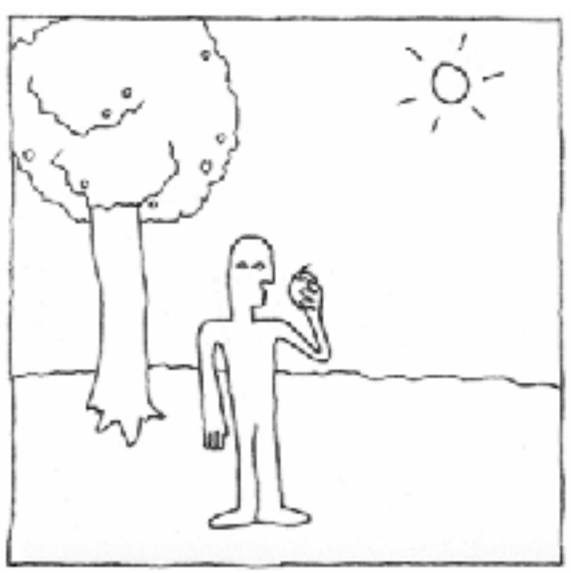
UIs more like the real world

Basic idea: interfaces emerging ca. 2008 *(VR/AR, tangible, ubicomp…)* were starting to be more “natural” than WIMP—what made them feel like this?



Naïve physics

Body awareness & skills

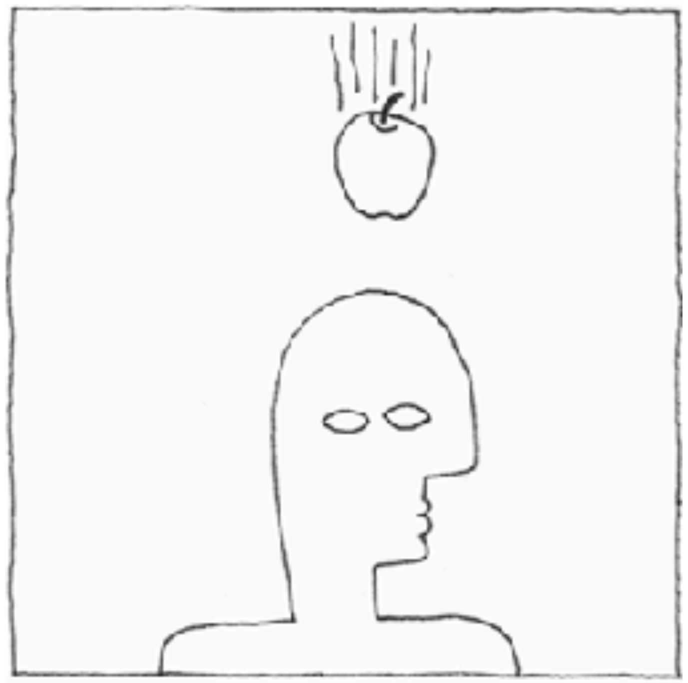


Environment awareness & skills

Social awareness & skills

Reality-based interaction

Naïve physics



Humans intuitively understand how the physical world works: weight, inertia, friction, etc

Rock & Rails

Rocks are heavy, so a "rock" gesture "holds down" the corner of the picture, letting you stretch it.

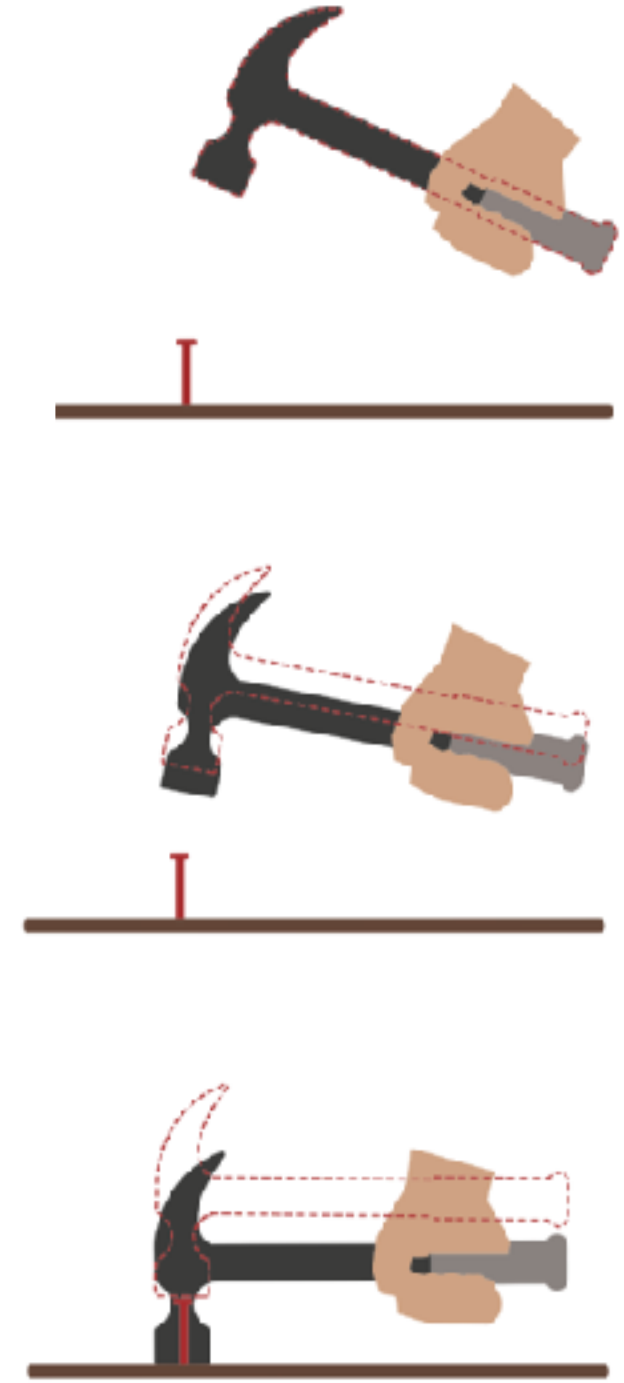
Reality-based interaction

Body awareness and skills

People are good at their own bodies—we can coordinate actions, locate our limbs, move in the world…

Knock on wood

I "know" where my hand is.
The virtual hammer lags behind the real one, hitting the virtual nail when the real hammer hits the real table.



How to analyze interfaces?

Instrumental Interaction:

An Interaction Model for Designing Post-WIMP User Interfaces

Instrumental interaction

A way to think about how interfaces work

Main idea: extend principles of direct manipulation

Goal: new **interaction model** to guide interface design in post-WIMP era

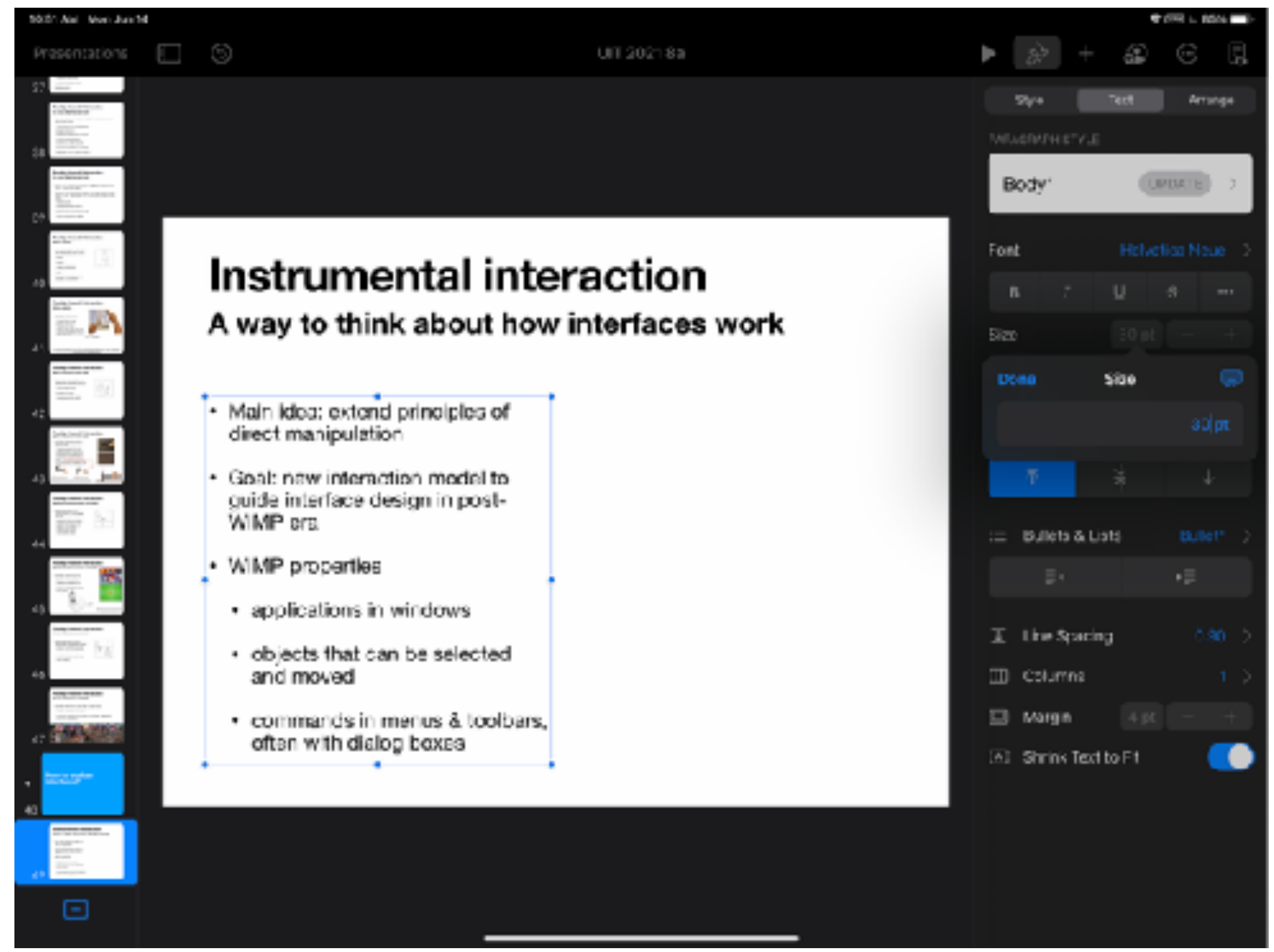
What's an interaction model?

- "principles, rules, and properties that guide the design of an interface"
- Example:
  direct manipulation is an interaction model

Direct manipulation vs WIMP

Main DM principles

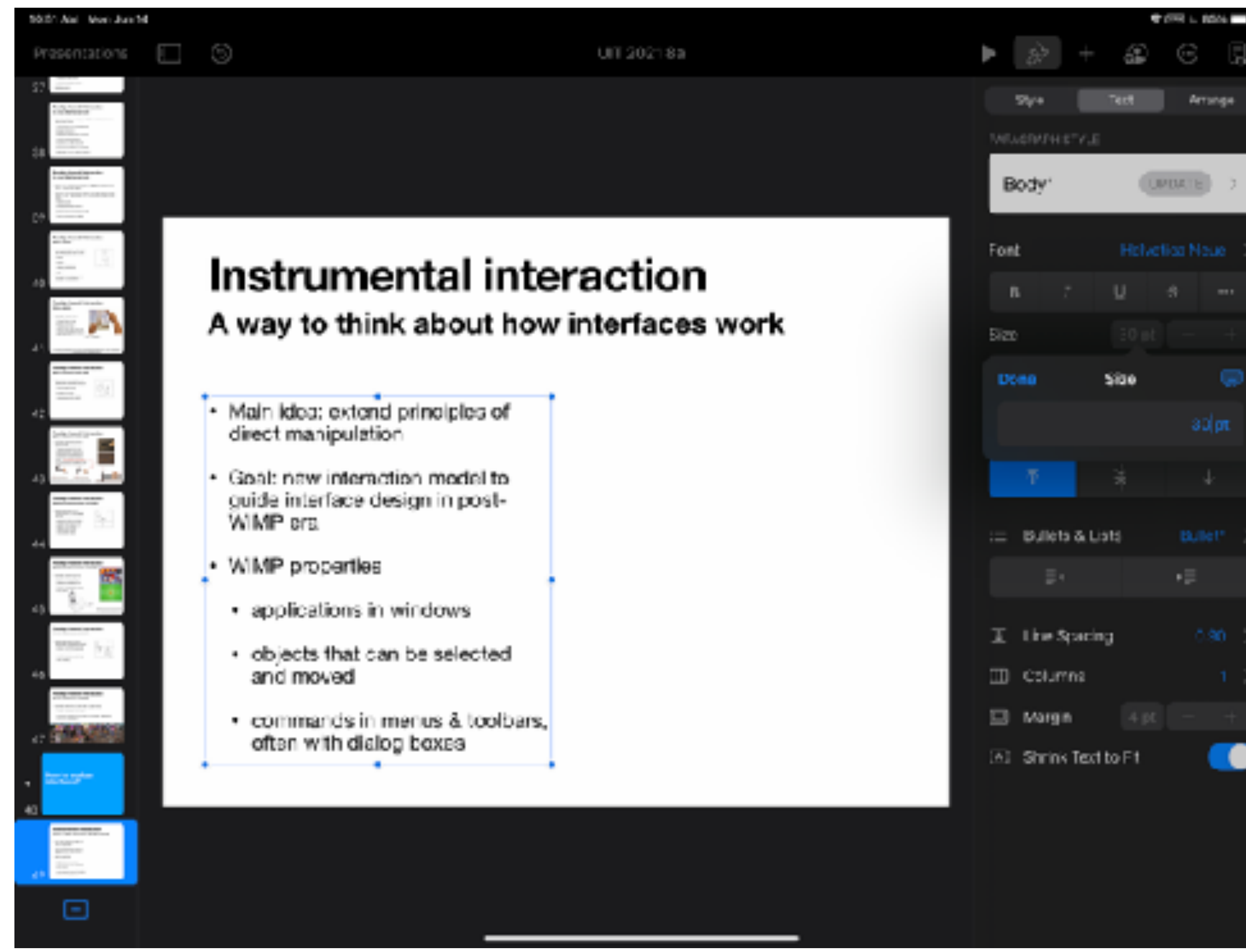
- **Object(s) of Interest (OOI):** the thing(s) we care about in the interface: document, image, web page, etc
- **Continuous representation** of OOI
- **Physical actions** on OOI
- **Fast, incremental, reversible actions**
- **Smooth learning curve**



Direct manipulation vs WIMP

What does direct manipulation tell us about WIMP?

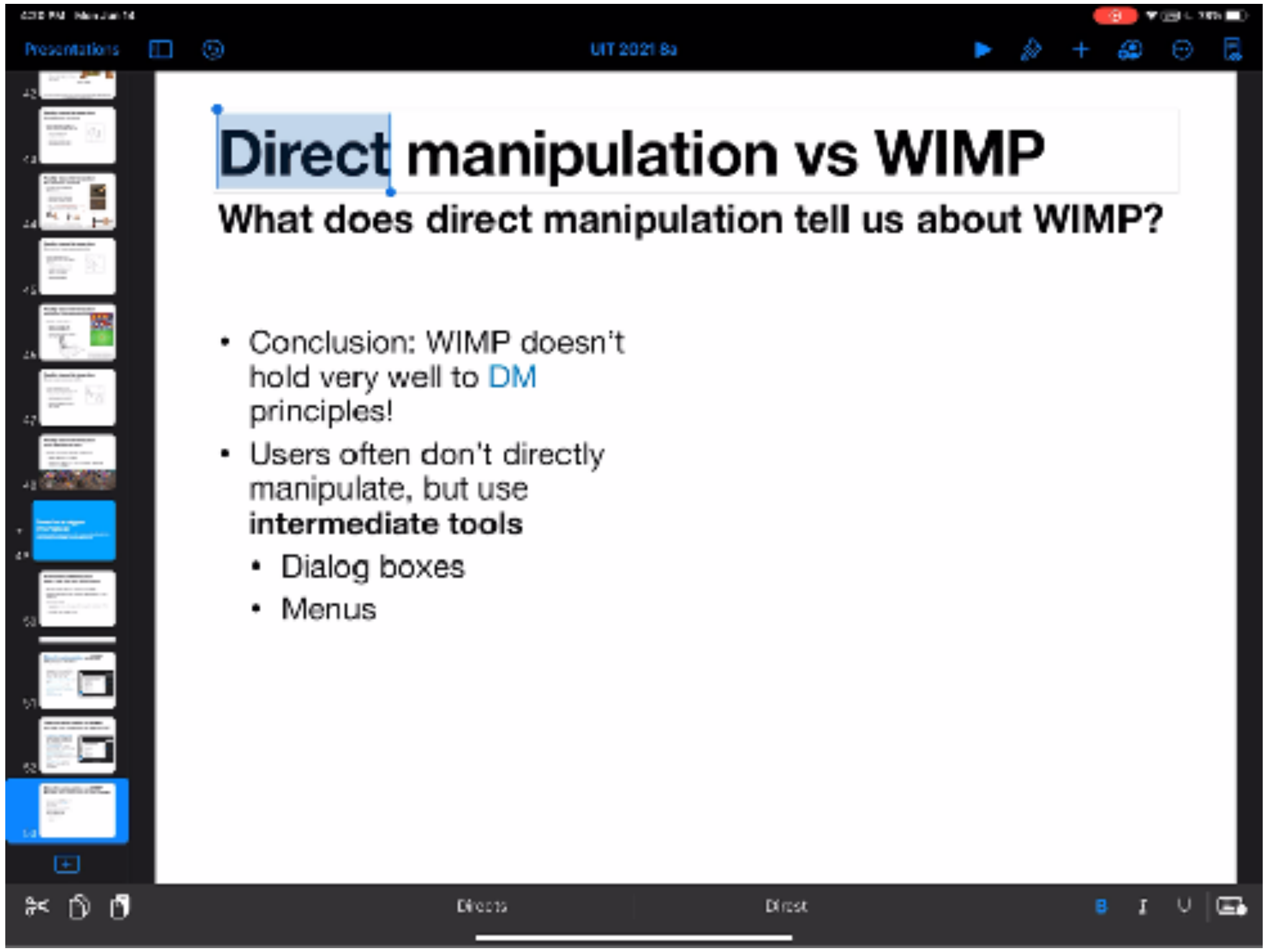
- **Continuous representation**:
  some OOIs are hidden (e.g., doc styles) so dialog boxes have to be used
- **Physical actions**:
  not that many in WIMP; lots of dialog boxes and menus
- **Fast, incremental, reversible actions**:
  often must confirm before seeing effects (e.g. font size)
- **Smooth learning curve**:
  hidden shortcuts, low inter-app consistency



Direct manipulation vs WIMP

What does direct manipulation tell us about WIMP?

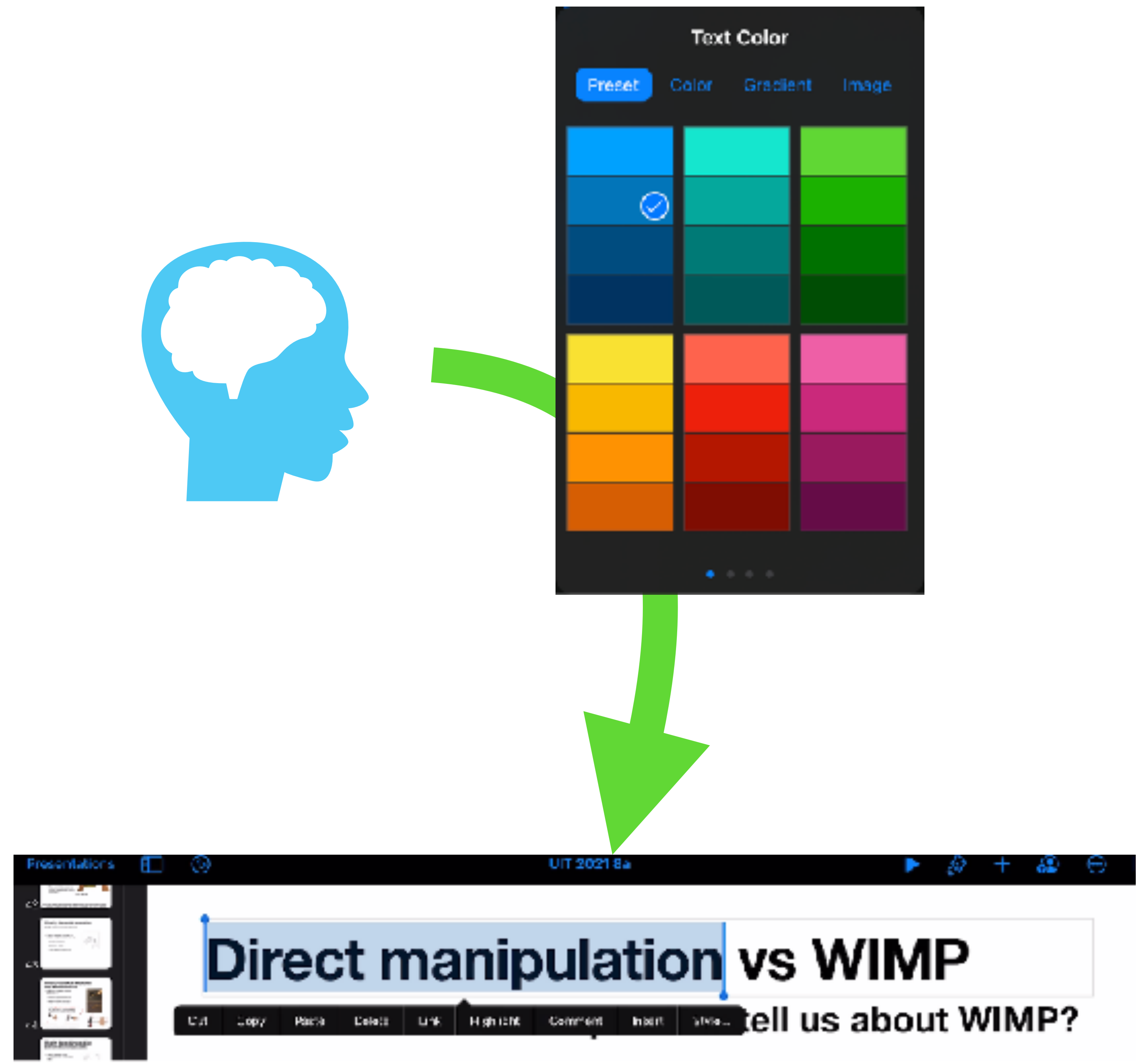
- Conclusion: WIMP doesn't hold very well to **DM** principles!
- Users often don't directly manipulate, but use **intermediate tools** like
  - Dialog boxes
  - Menus
  - Scroll bars



Direct manipulation vs WIMP

What does direct manipulation tell us about WIMP?

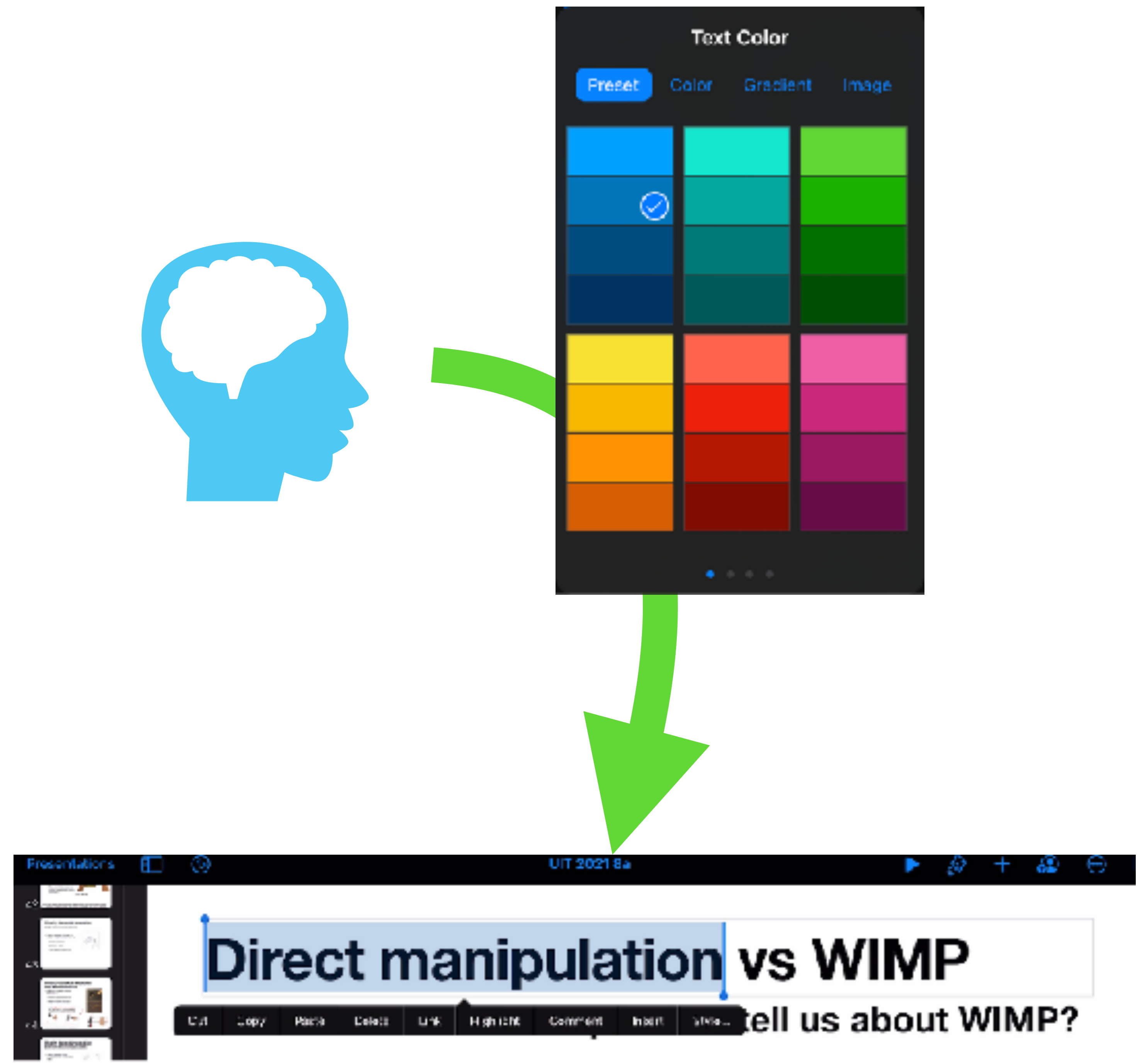
- Conclusion: WIMP doesn’t hold very well to DM principles!
- Users often don’t directly manipulate, but use **intermediate tools**
  - Dialog boxes
  - Menus
  - Scroll bars
- Tools act as **mediators** between user and OOIs!



Tools as mediators

We’re used to it

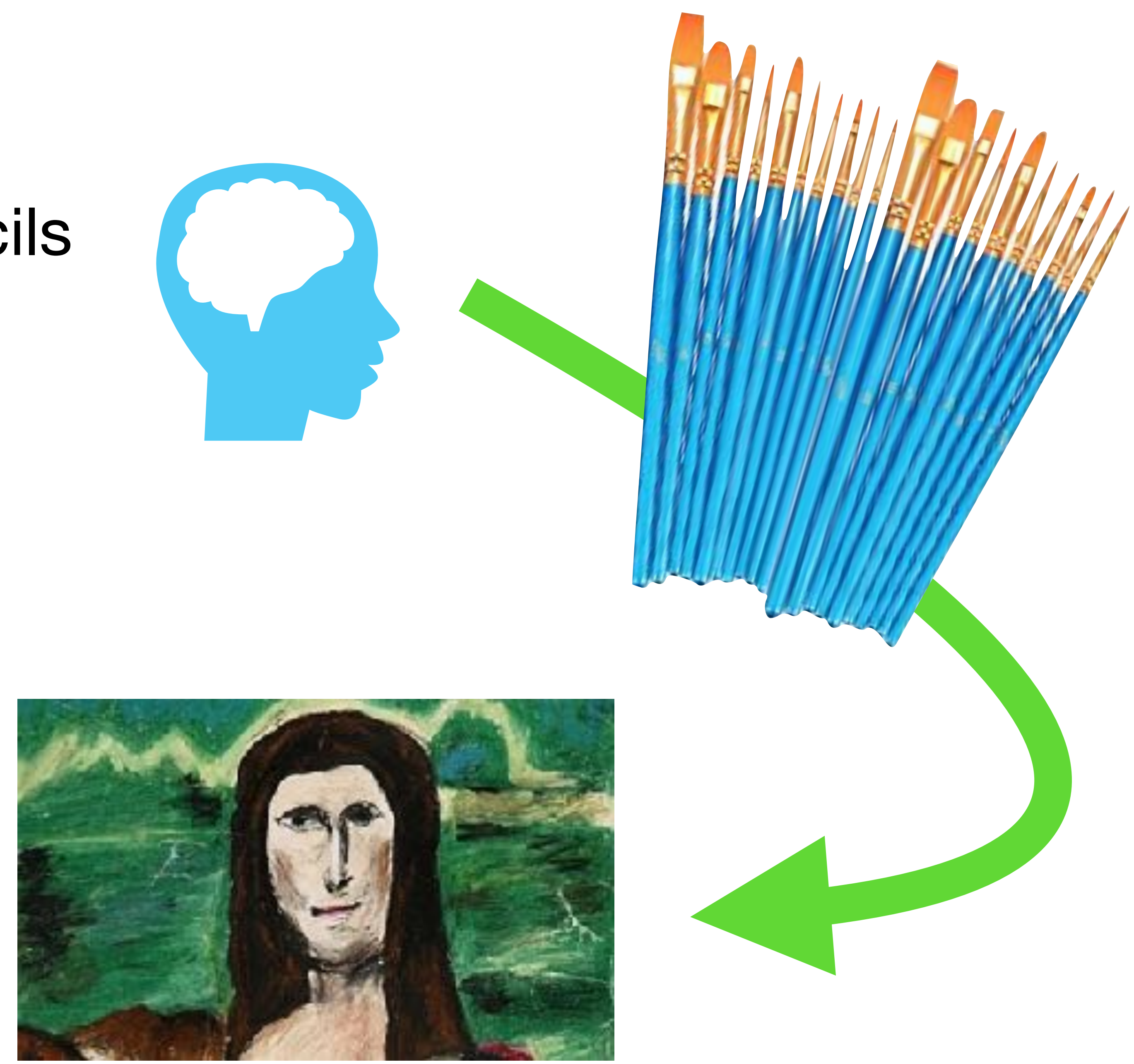
- We use tools to manipulate OOIs all the time!



Tools as mediators

We’re used to it

- We use tools to manipulate OOIs all the time!
  - Paintbrushes, pens, pencils
  - Forks, knives, spoons
  - Screwdrivers, hammers, saws



Domain objects

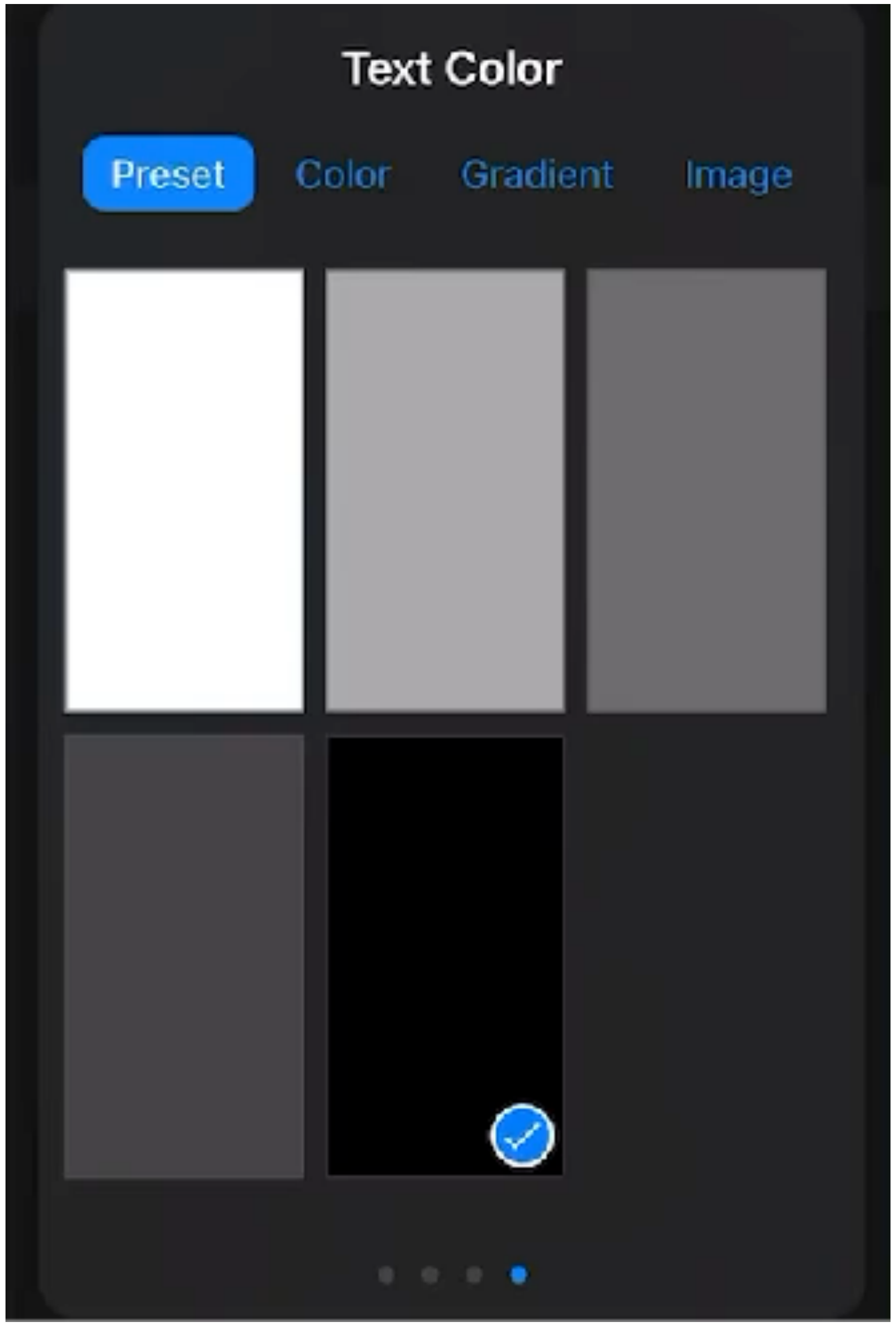
OOIs we manipulate

- Example: **Direct manipulation**
  - Font: Helvetica Neue
  - Weight: bold
  - Size: 30pt
  - Color: black
- Example: **Header style**
  - Same attributes but more general

Instruments

Tools we use to manipulate Domain Objects

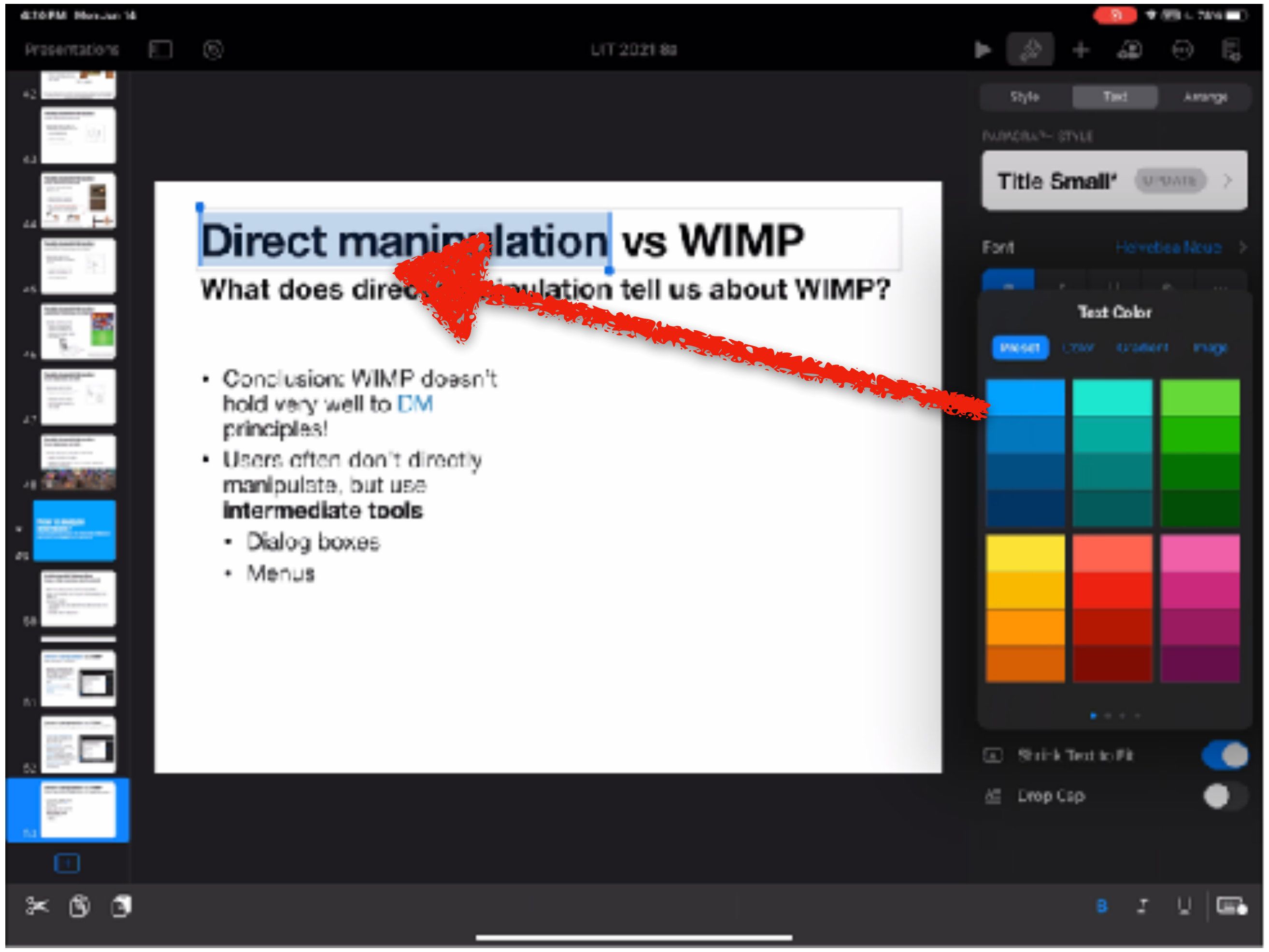
- Two-way mediator between user and **Domain Object**
- User *acts* on **Instrument**
- In turn, **Instrument**
  - *reacts* to provide *feedback*
  - *commands* **Domain Object**



Instruments

Properties to help us think about them

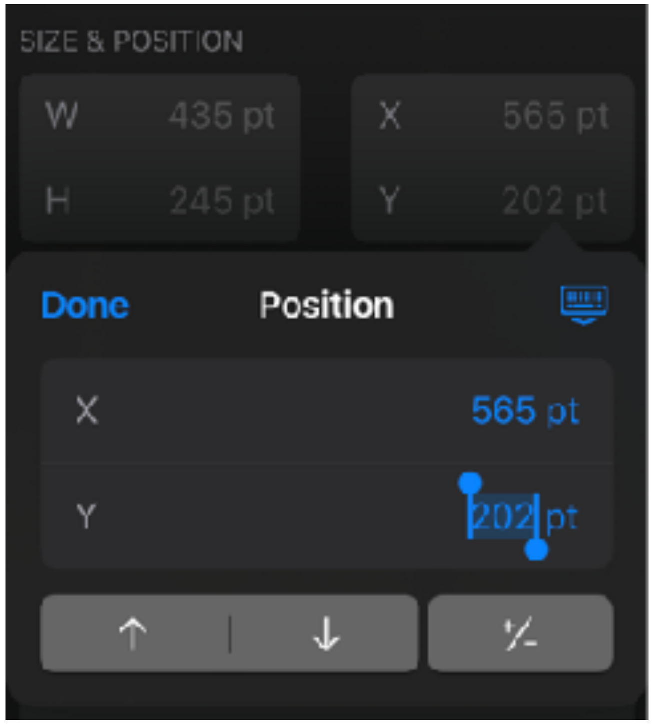
- Degree of **indirection**: distance from instrument to object
  - *Spatial:* "physical" distance
  - *Temporal:* time distance (how long to take effect?)
  - If both distances are zero, it's direct manipulation!



Instruments

Properties to help us think about them

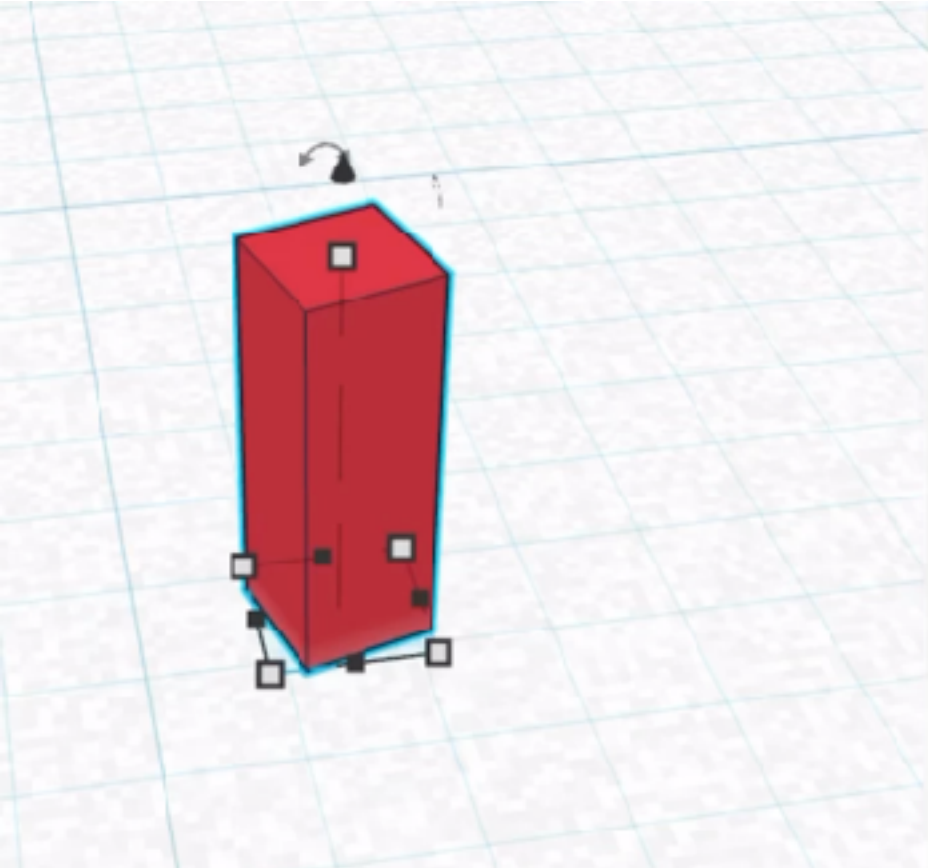
- Degree of indirection: distance from instrument to object
- Degree of **compatibility**: input action vs feedback similarity
  - Touchscreen drag object: high compatibility
  - Type in numbers to move object: low compatibility



Instruments

Properties to help us think about them

- Degree of indirection: distance from instrument to object
- Degree of compatibility: input action vs feedback similarity
- Degree of **integration**: dimensionality of action vs effect
    - Drag slide object: 2D input, 2D output
    - Drag to place & rotate object in space: 2D input, 6D output



Wrapping up

Concept summary

- **Direct manipulation**: feeling like you're working directly with the things you care about rather than telling the computer what to do
- **Post-WIMP interfaces**: using non-traditional I/O like gesture, speech, or haptics
- **Reality-based interaction**: using innate knowledge of how the world works to use interfaces
- **Instrumental interaction**: what leads to a feeling of directness?

Questions?

You can also always post to Slack.

What next?

After lunch
Working with the lab

- We’ll meet here, then move to the lab for the second half of class
- You should expect to spend time in the lab outside of class! — Make sure you send in your ID card info (this was on Absalon)
- Nathalia will be in the lab sometimes during the week; office hours to be posted (watch on Absalon/Slack)
- You’ll have bins in the lab to keep your things in

Lunch time!

Come back by 13:00—we will start on time

The lab

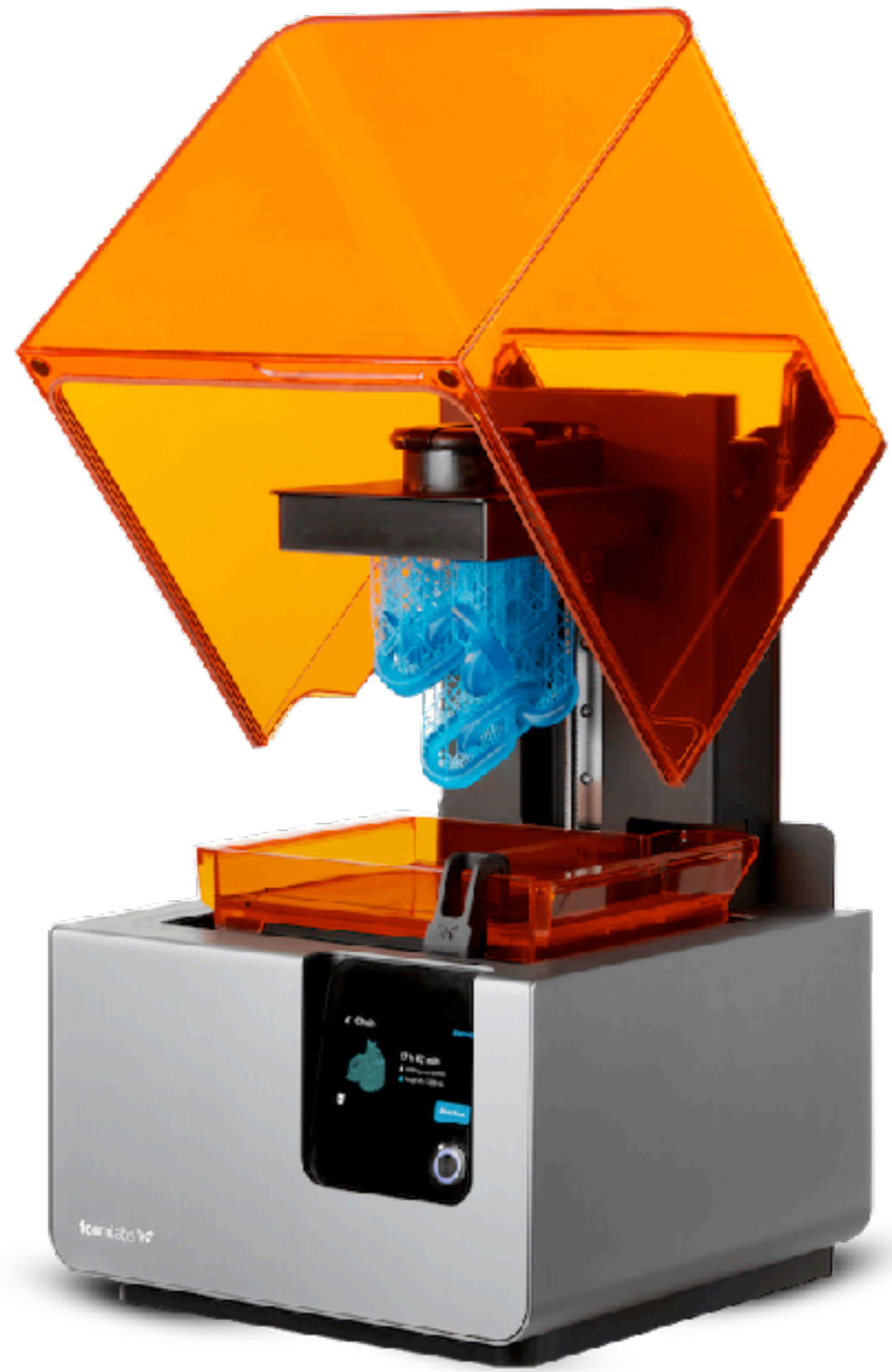
Wonderland of fun

Equipment & supplies

For class and your project

3D printers

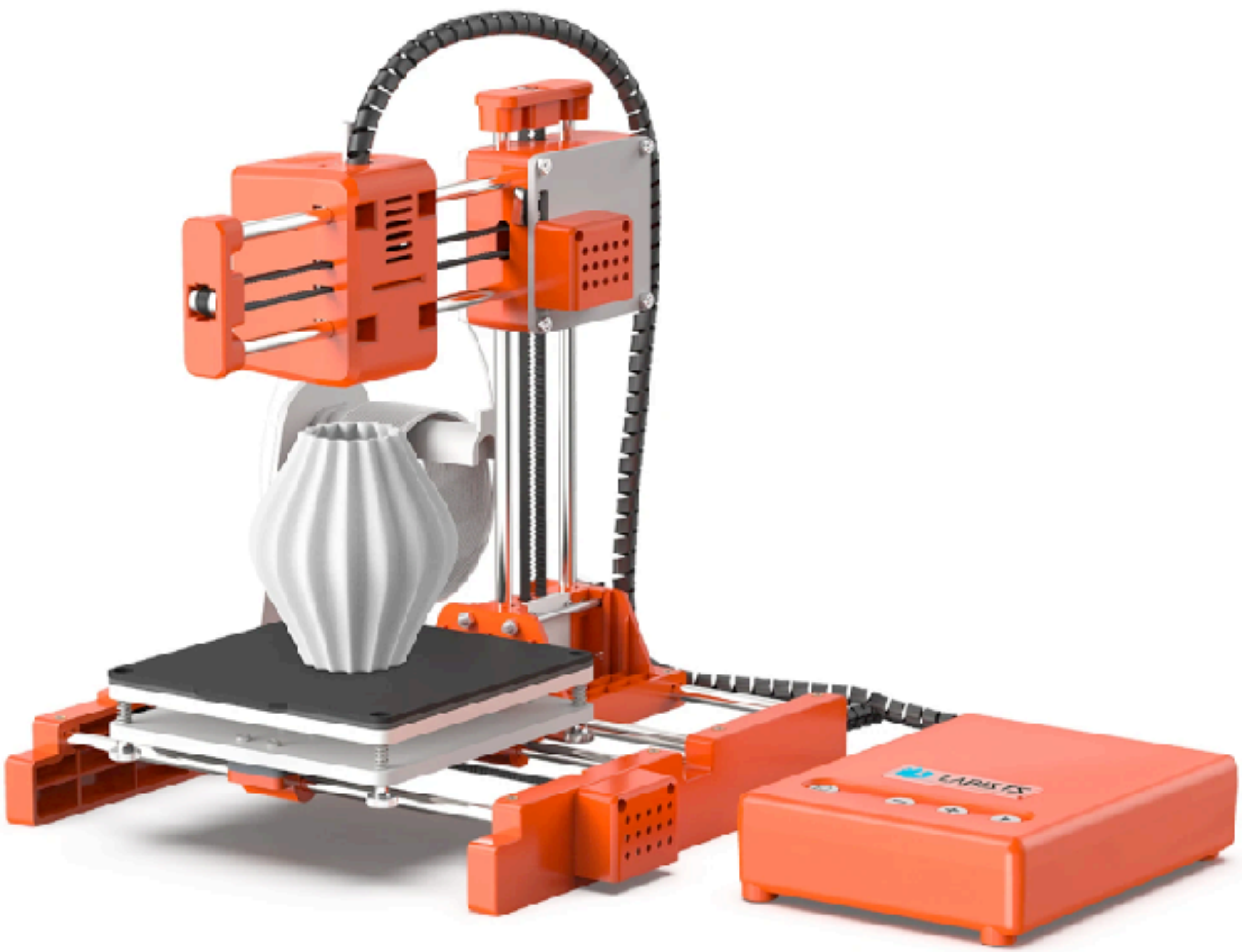
FDM (the “standard”)



SLA (ultra-high resolution)

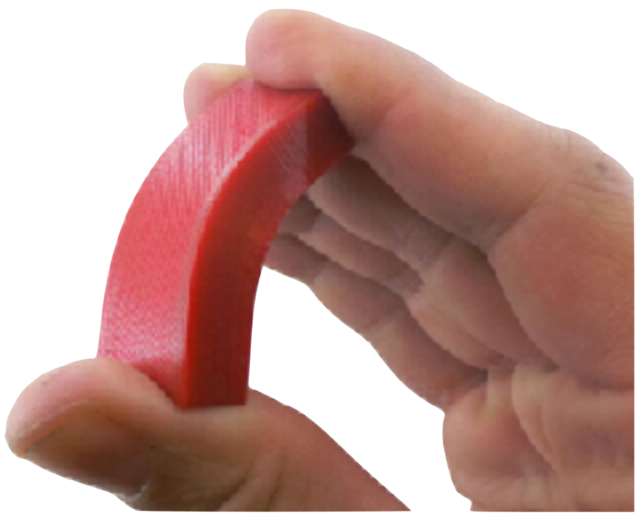
3D printers

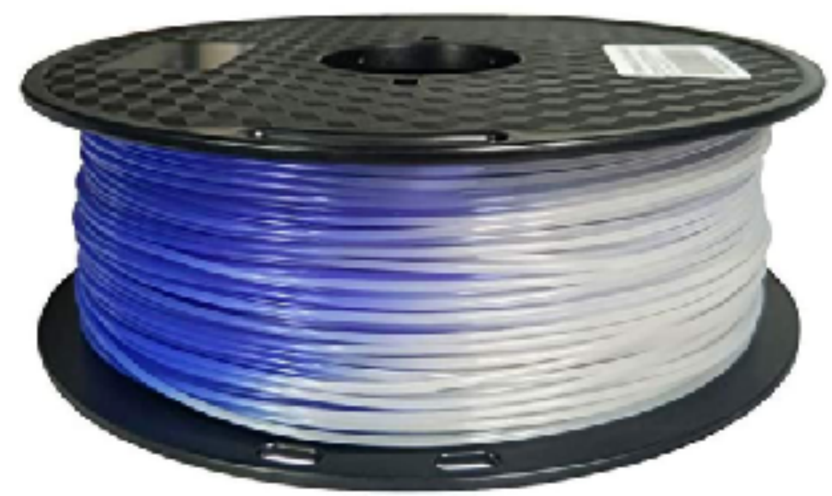
Kinda crappy tiny printers to borrow or modify



FDM (the “standard”)

Fancy filaments





Laser cutter

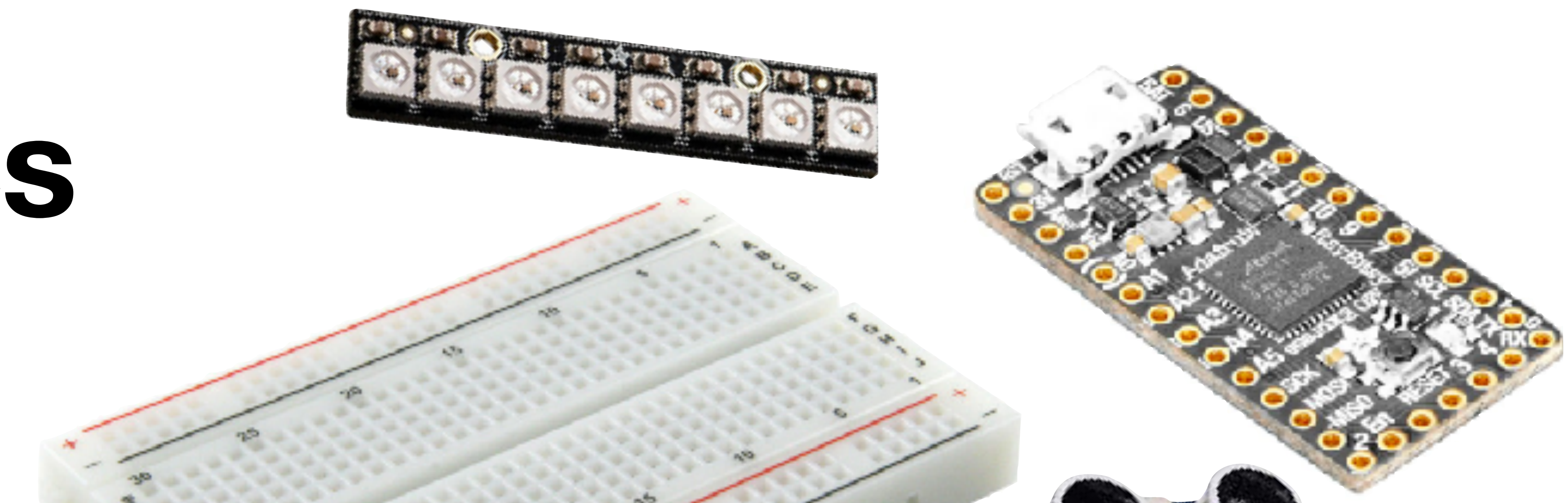
It can burn down the building

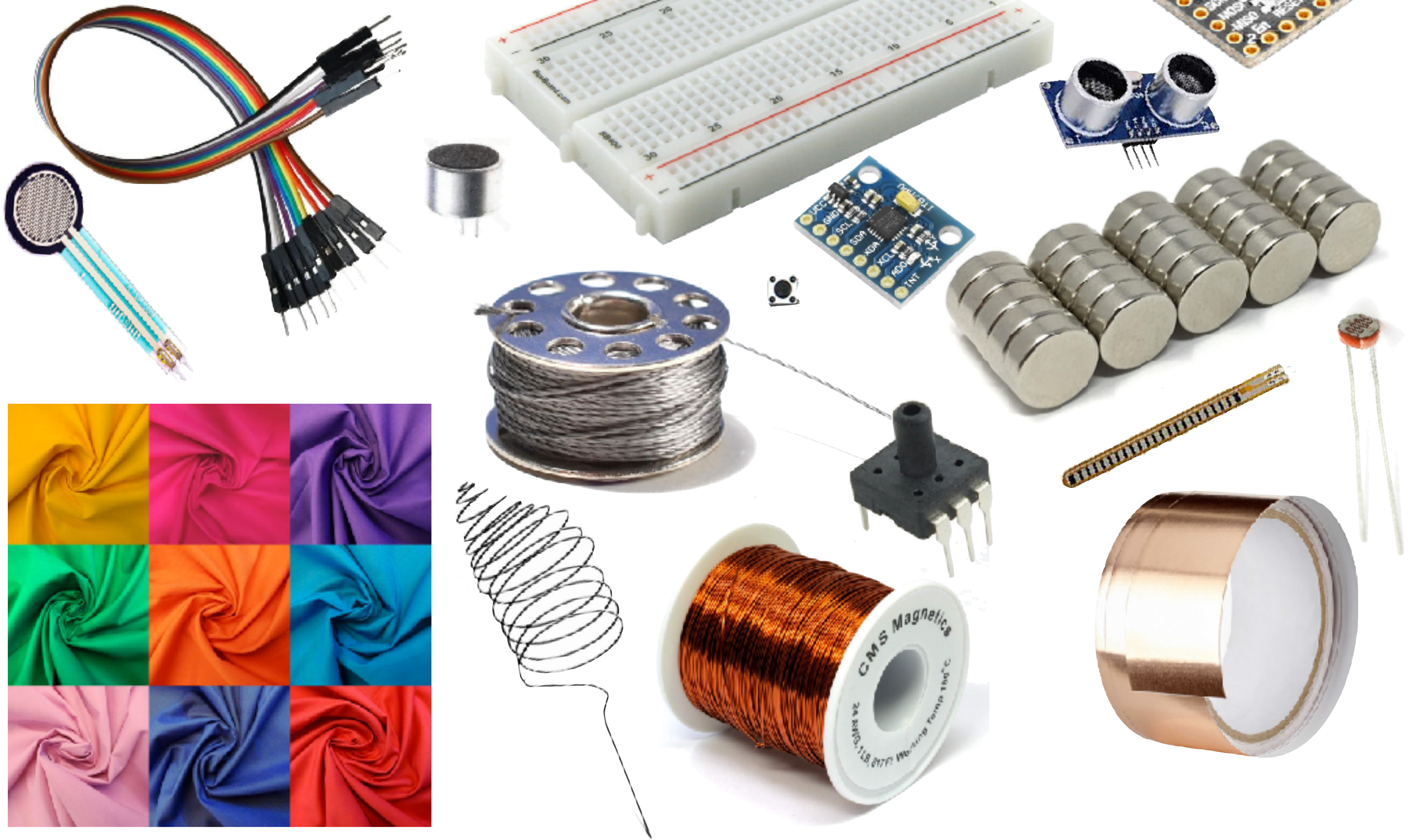
Cutting machines

Cricut Maker

Materials to cut

Electronics





Tools

and other stuff

- Electronics lab
  - Soldering irons
  - Oscilloscope
  - Multimeters
- Sewing machine
- “Hard” tools like drills, saws, screwdrivers
- Probably a bunch of other stuff—ask if unsure

Lab rules

Please don't make our colleagues angry at us

1. **Clean up after yourself**
   - Plan your work to **end 15 minutes early** so you have time to do a thorough job
   - Put your stuff in your bin
   - Put the equipment and tools away
   - Throw away your trash

2. **Clean up after others**
   - If someone left a mess:
     take a photo, post to Slack, then clean it up!
     Post a follow-up photo.
   - Don't make someone clean up after you!

3. **Be respectful:** lots of people use the labs
   - **Clean up**
   - Don't be super loud

4. **Get help if you need it**
   - Don't know how to use something? Ask on Slack!
   - Don't know where something goes? Ask on Slack!
   - Emergency and have to leave a mess? Post on Slack including a **concrete time** you will return to **clean up**!

Nathalia will be in the lab
10-12 on Tuesdays

Exercise

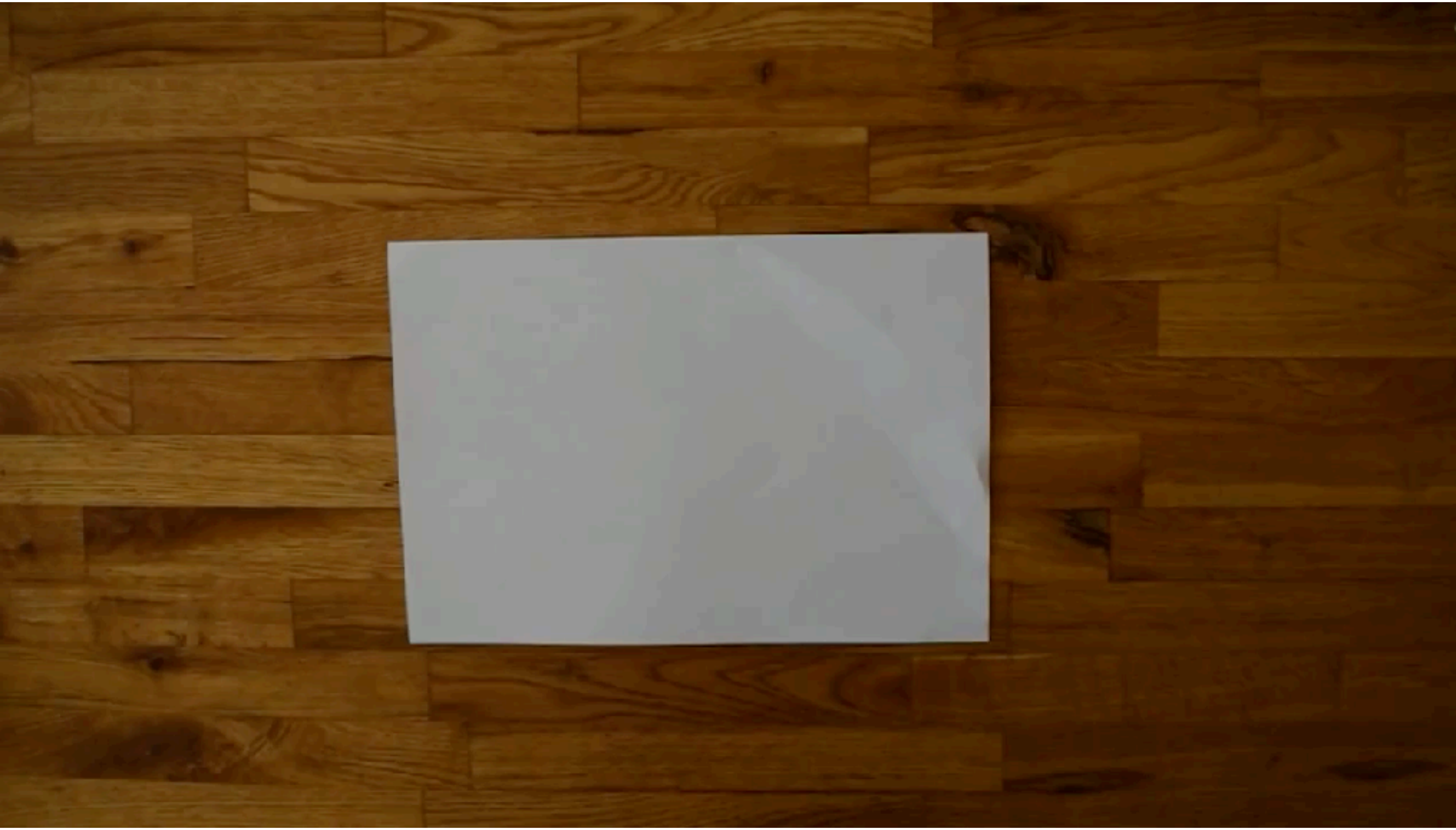
In-class exercises

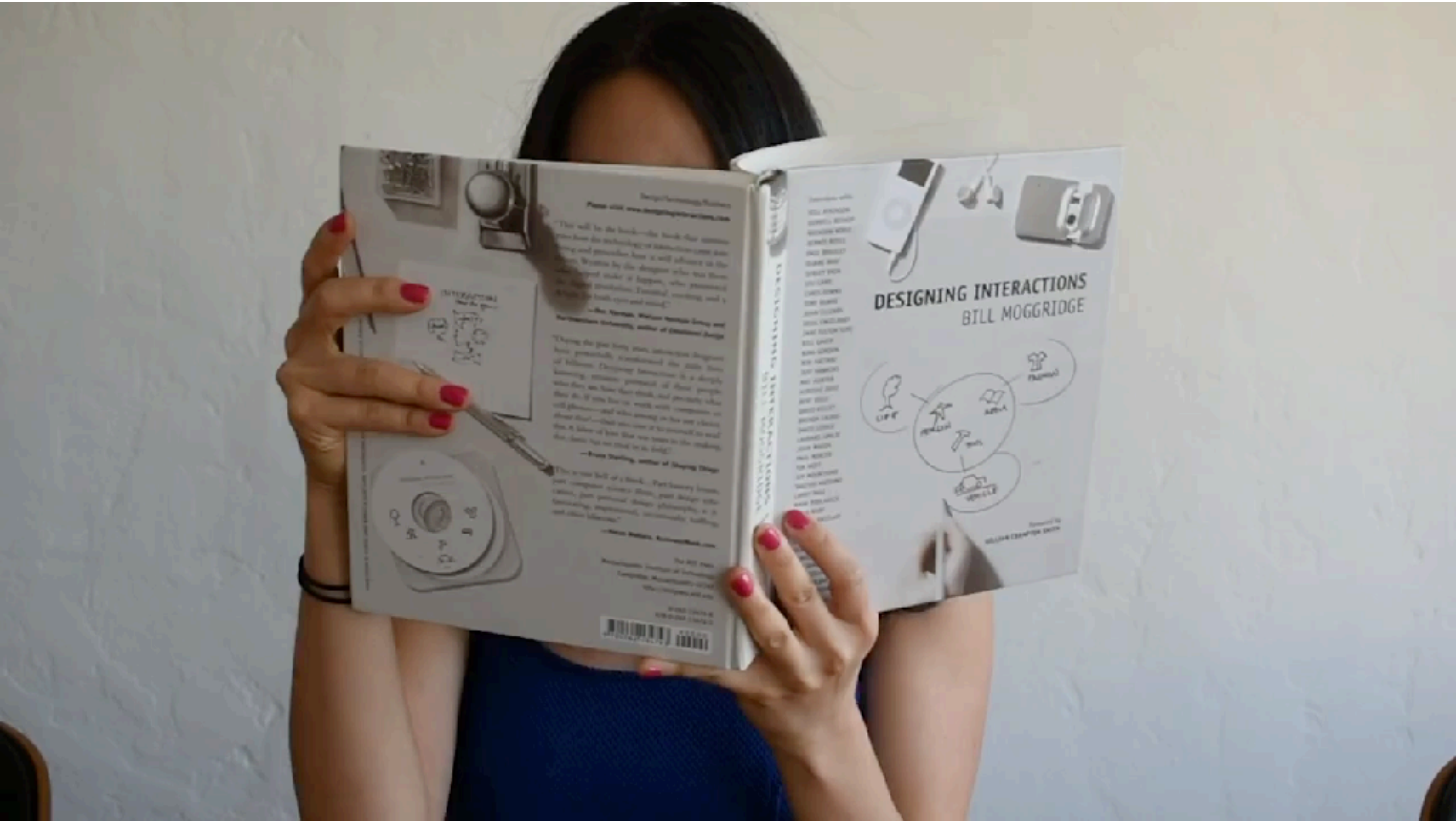
General flow

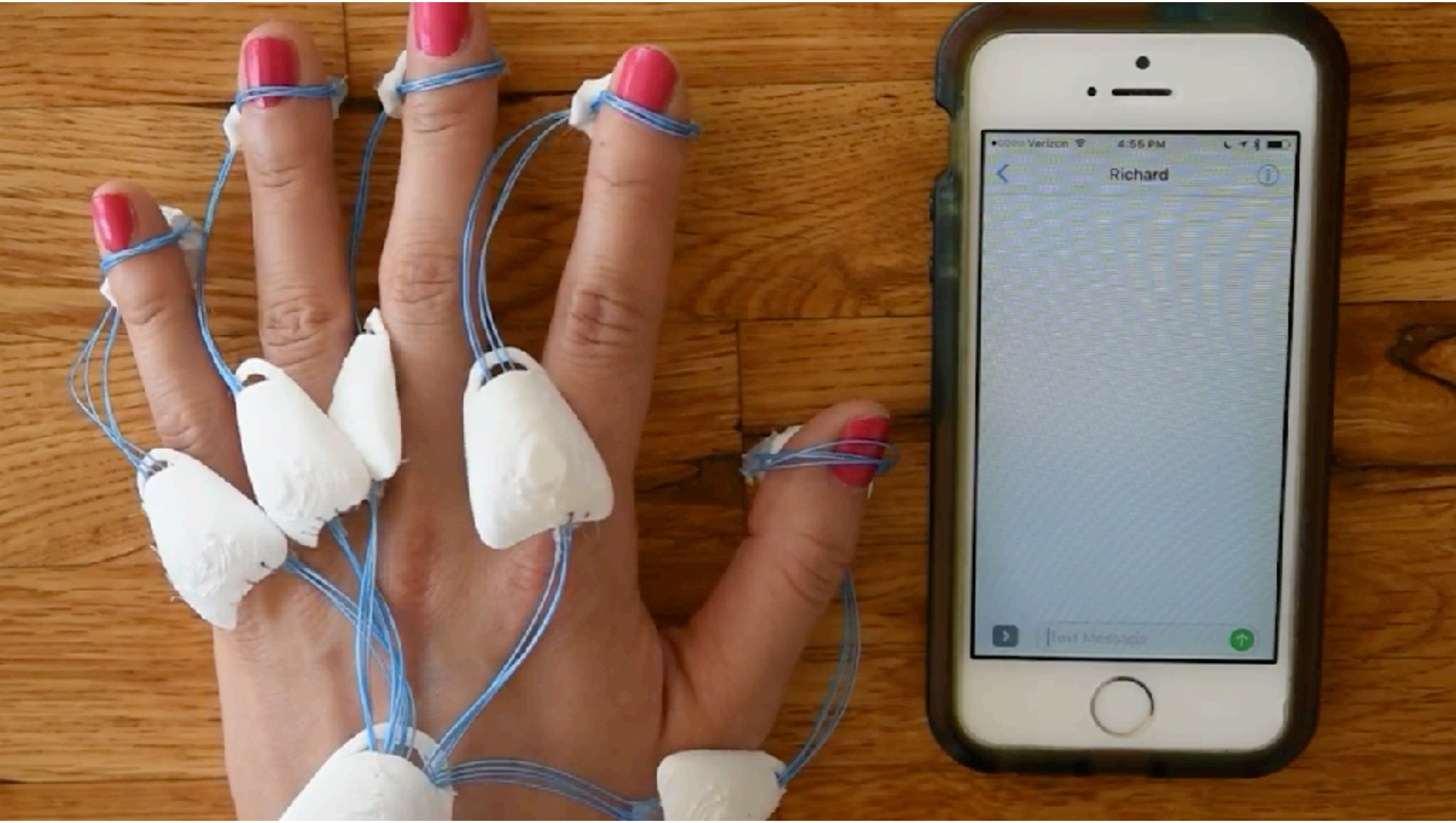
- We’ll introduce it (that’s about to happen!)
- You’ll work on it for ~1hr
- You’ll wrap up by making a 1-minute video that discusses your prototype (and upload to Absalon):
  - what does it do?
  - how does it work?
  - what would you do differently?
  - what went well?

Exercise

Make a chording keyboard prototype!




DESIGNING INTERACTIONS
BILL MOGGRIDGE

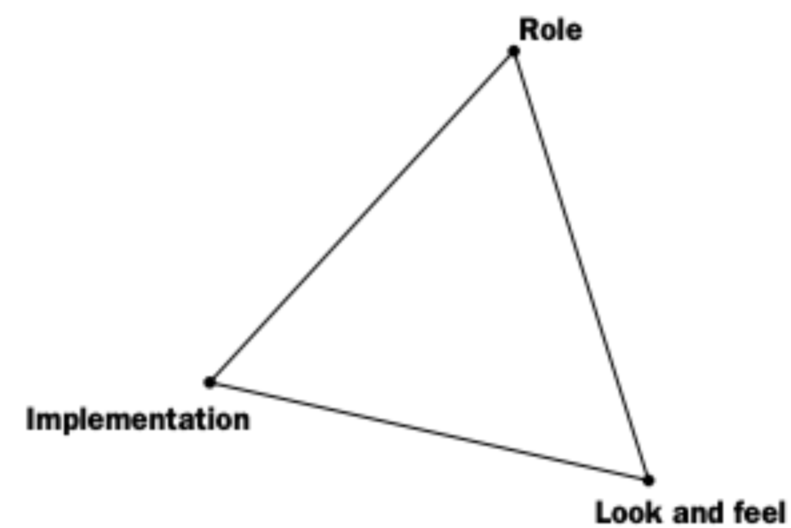

Verizon
4:56 PM
Richard
Text Message

Chording Keyboards

- 5-8 buttons
- Entry of text while on-the-go
- Lots of research on them going back to the 60s

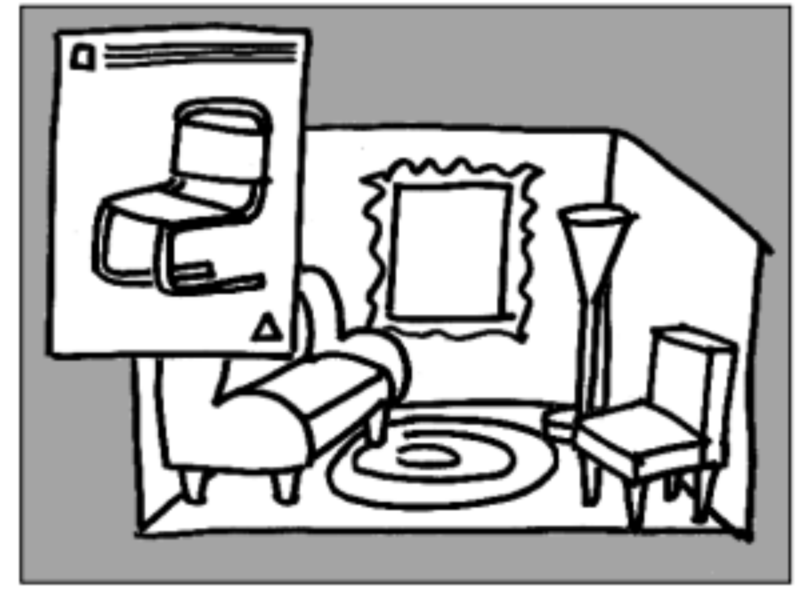
Prototyping

- What is a prototype?

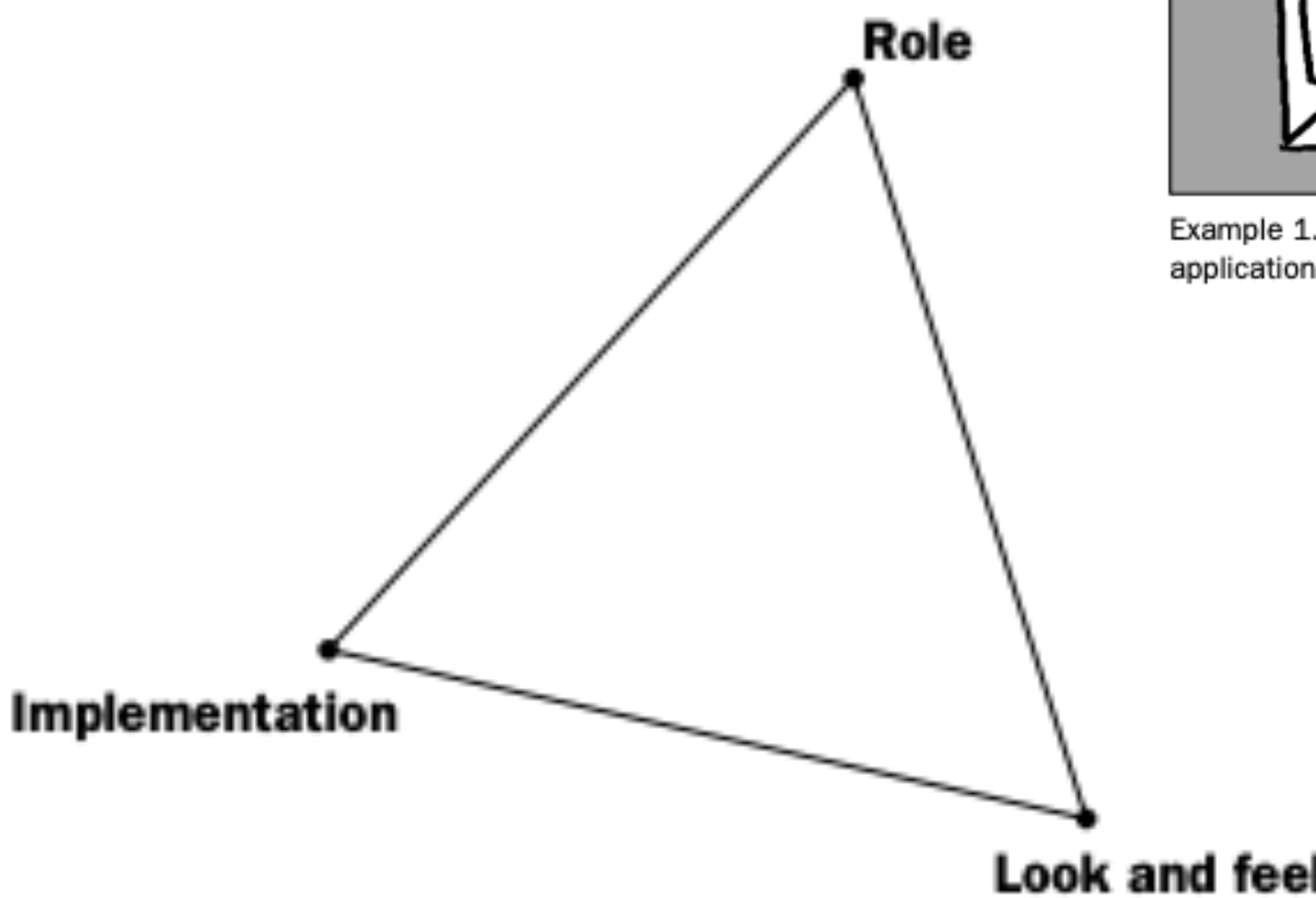


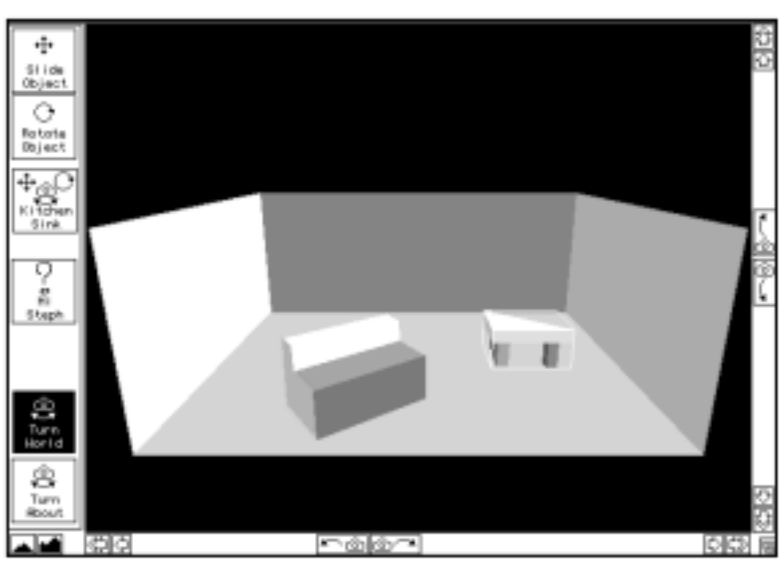
- What do prototypes prototype?

Houde, S., and Hill, C., What Do Prototypes Prototype?, in Handbook of Human-Computer Interaction (2nd Ed.), M. Helander, T.␣ Landauer, and P. Prabhu (eds.): Elsevier Science B. V: Amsterdam, 1997.

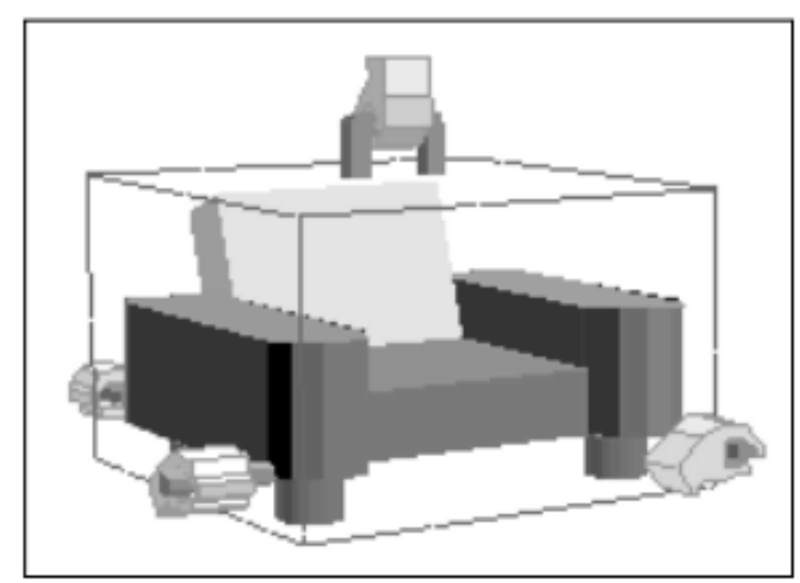


Example 1. Role prototype for 3D space-planning application [E1 Houde 1990].

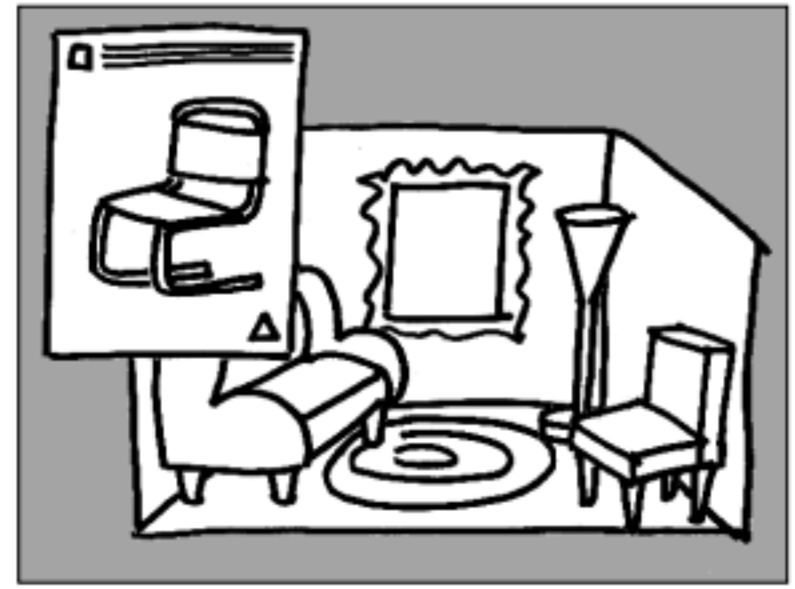




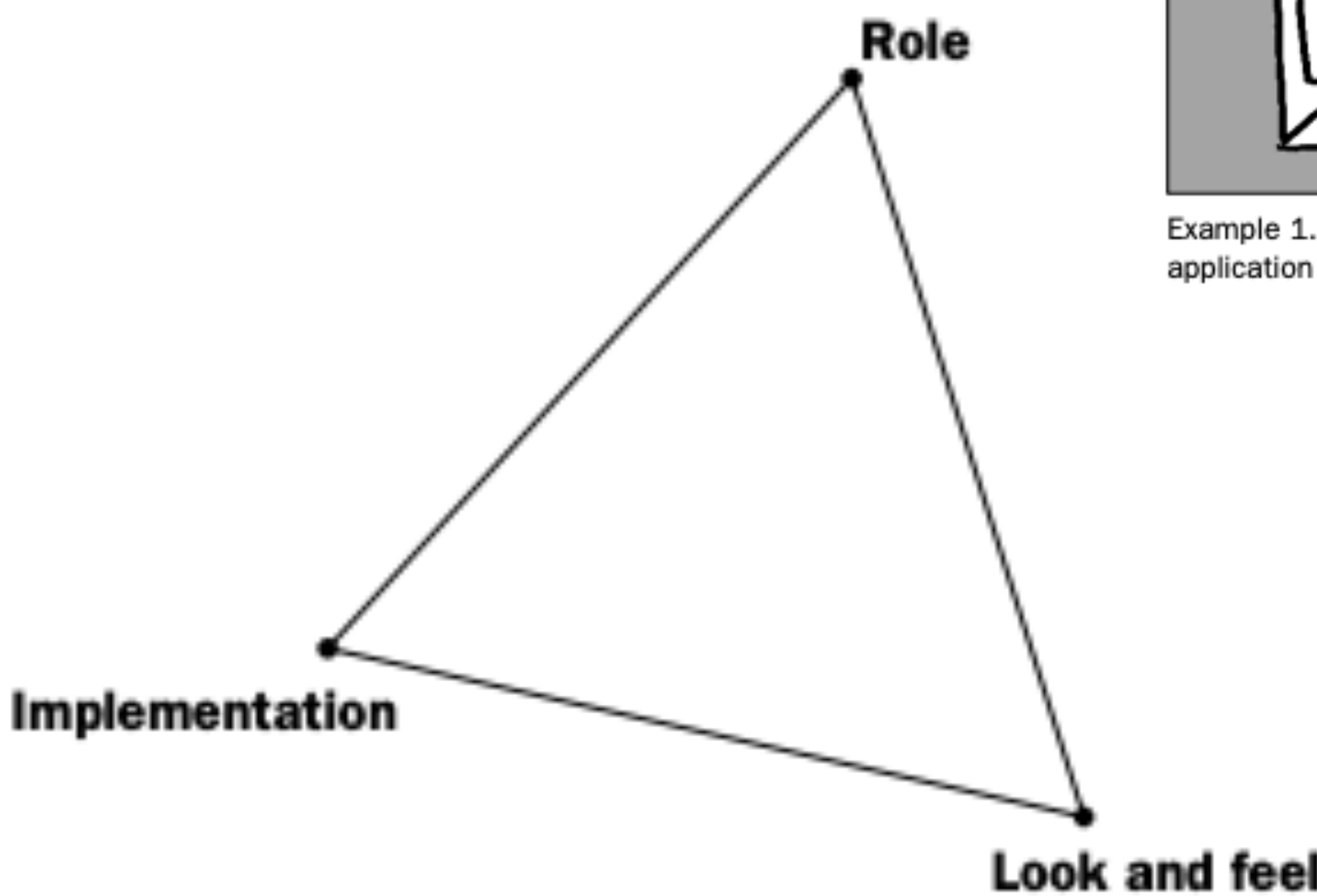
Example 3. Implementation prototype for 3D space-planning application [E3 Chen 1990].

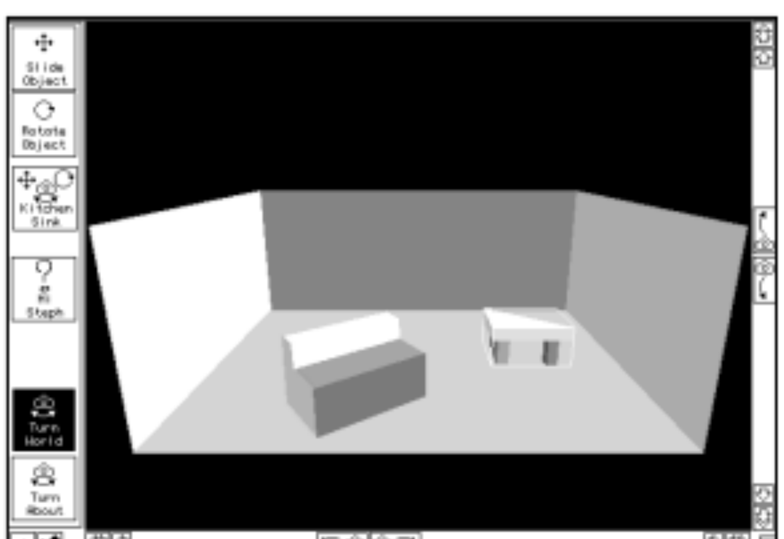


Example 2. Look-and-feel prototype for 3D space-planning application [E2 Houde 1990].

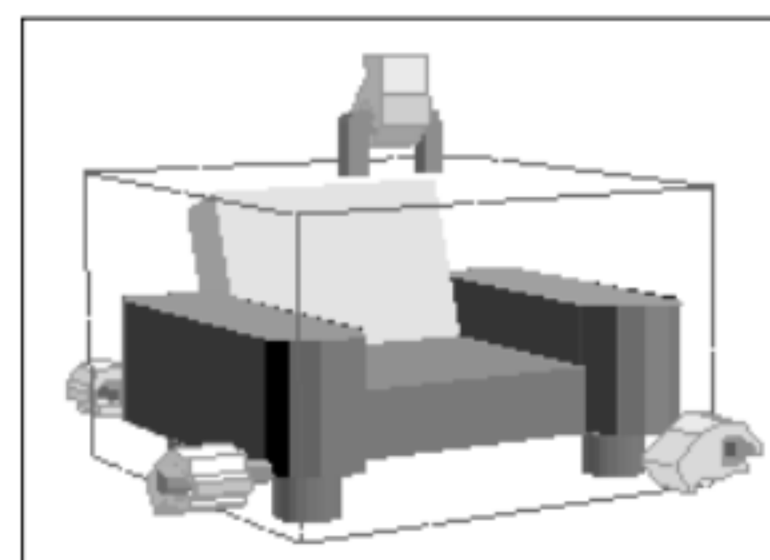


Example 1. Role prototype for 3D space-planning application [E1 Houde 1990].

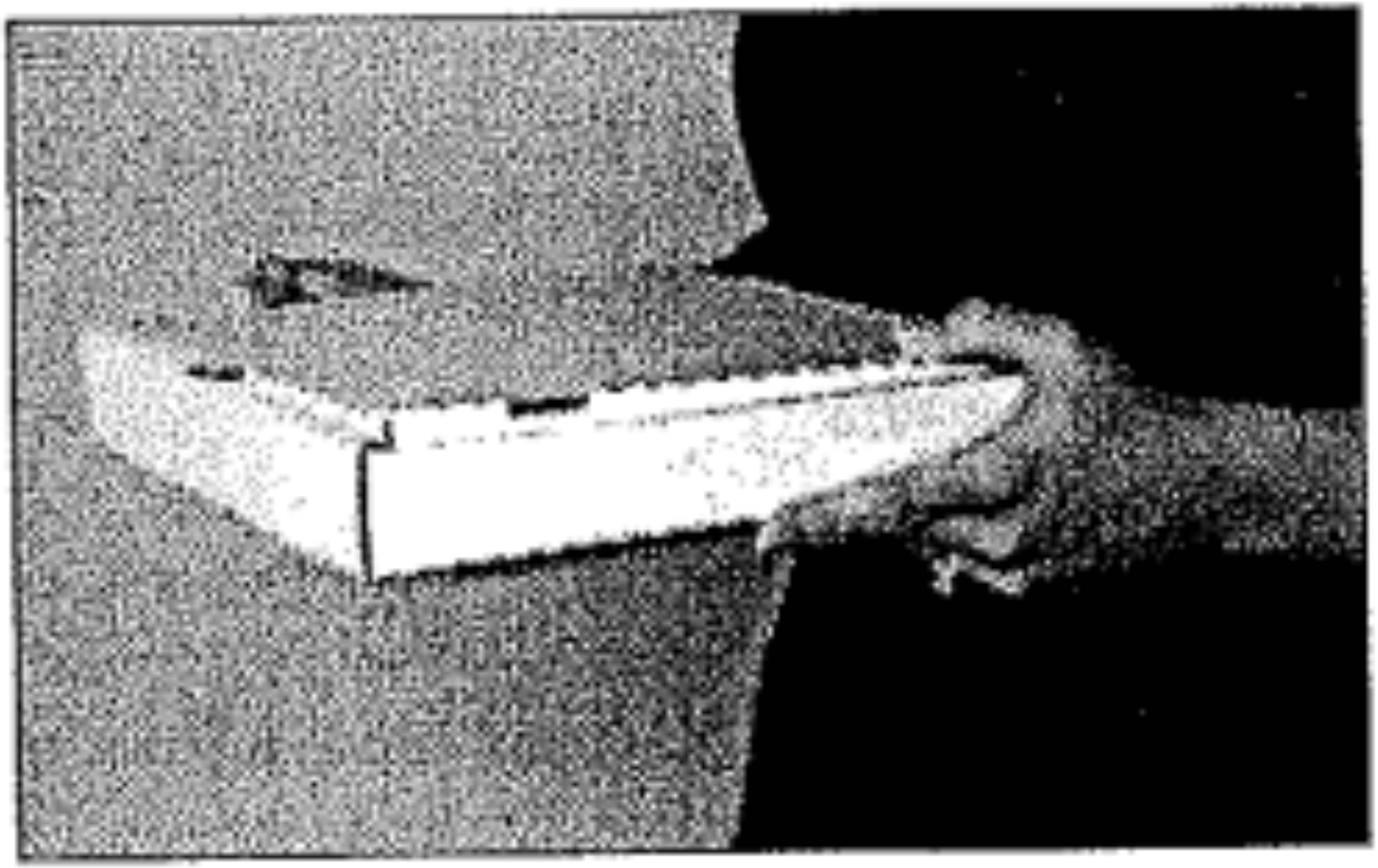




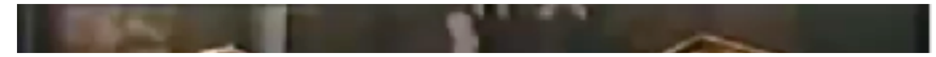
Example 3. Implementation prototype for 3D space-planning application [E3 Chen 1990].

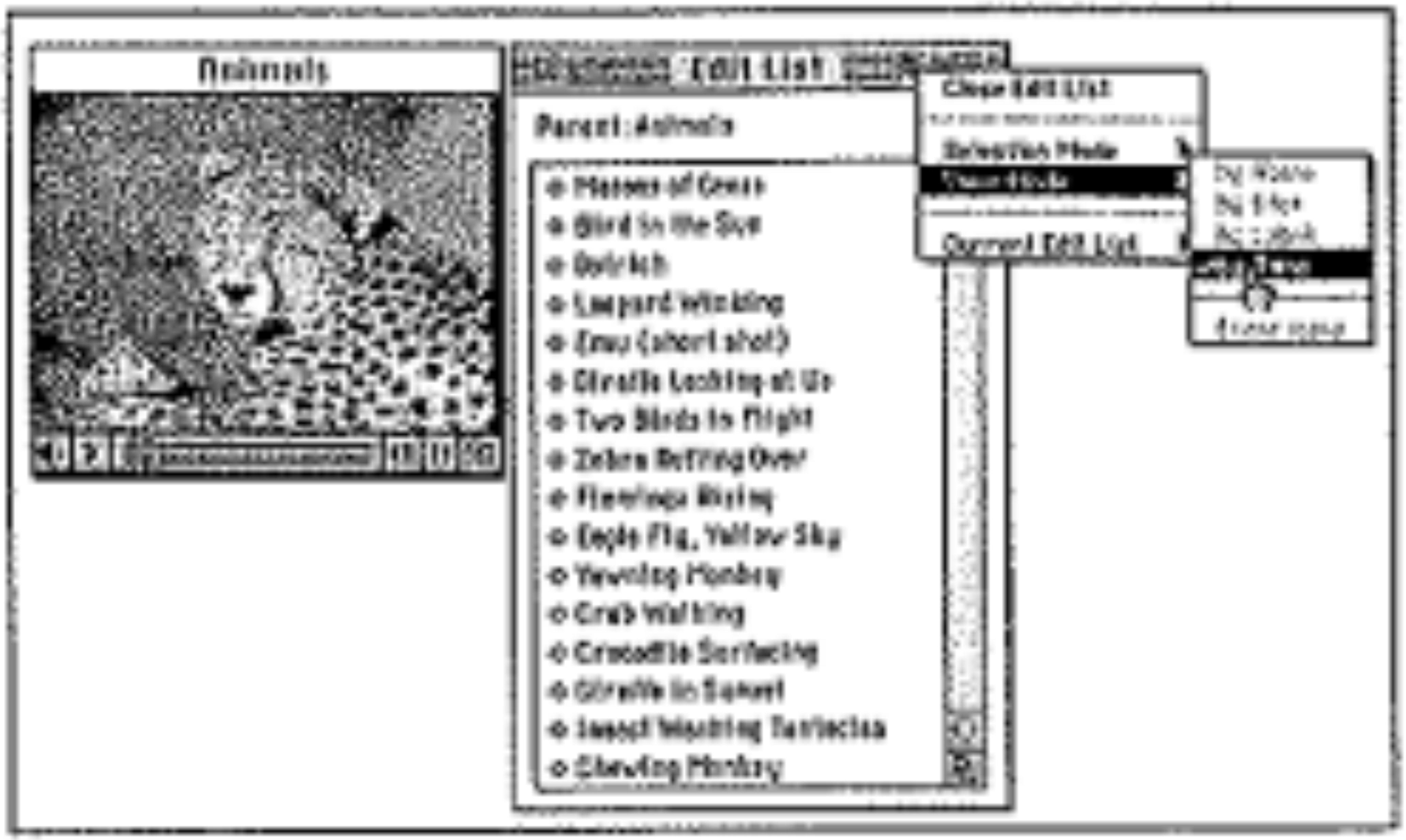


Example 2. Look-and-feel prototype for 3D space-planning application [E2 Houde 1990].



Example 10. Pizza-box prototype of an architect's computer [E10: Apple Design Project, 1992].





Example 11. Working prototype of a digital movie editor [E11: Degen, 1994].

Example 10. Pizza-box prototype of an architect's computer [E10: Apple Design Project, 1992].

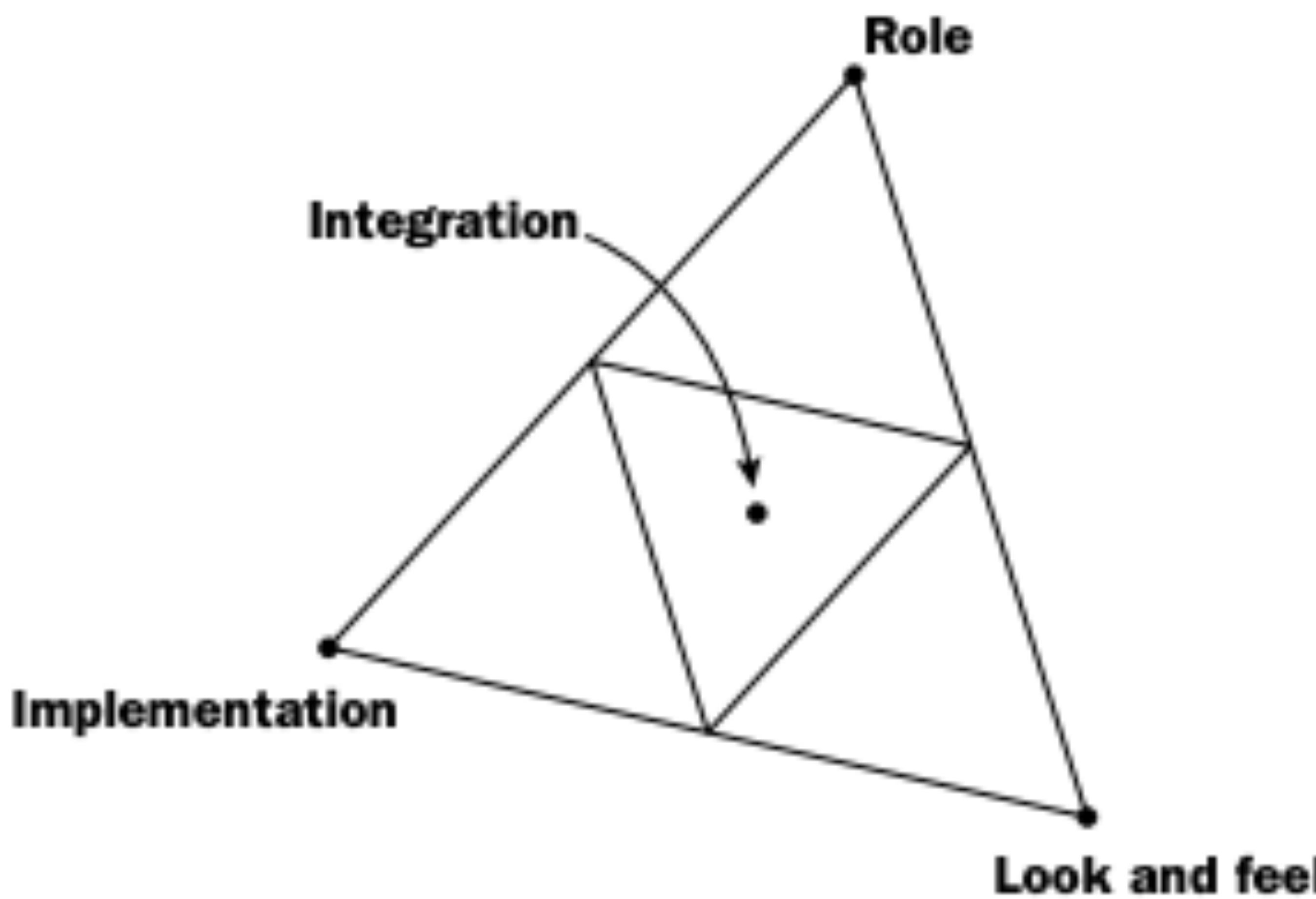


Figure 3. Four principal categories of prototypes on the model.

Today: focus on look and feel and role

Lab tour!

+ making chorded keyboards

Project work time

Project work time

- Work on your project or group assignment for next week
- Finish the in-class exercise if you didn’t get there yet
- Work on individual homework for next week

Project work time

- Work on your project or group assignment for next week
  - This week: watch videos from others to get ideas for your project. Talk with others in the class about their videos! Form groups.
- Finish the in-class exercise if you didn't get there yet
- Work on individual homework for next week
  - Get set up for microcontroller development
  - Practice soldering by soldering headers onto your boards
  - Practical sensing activities