

Designing Interactive Systems 2

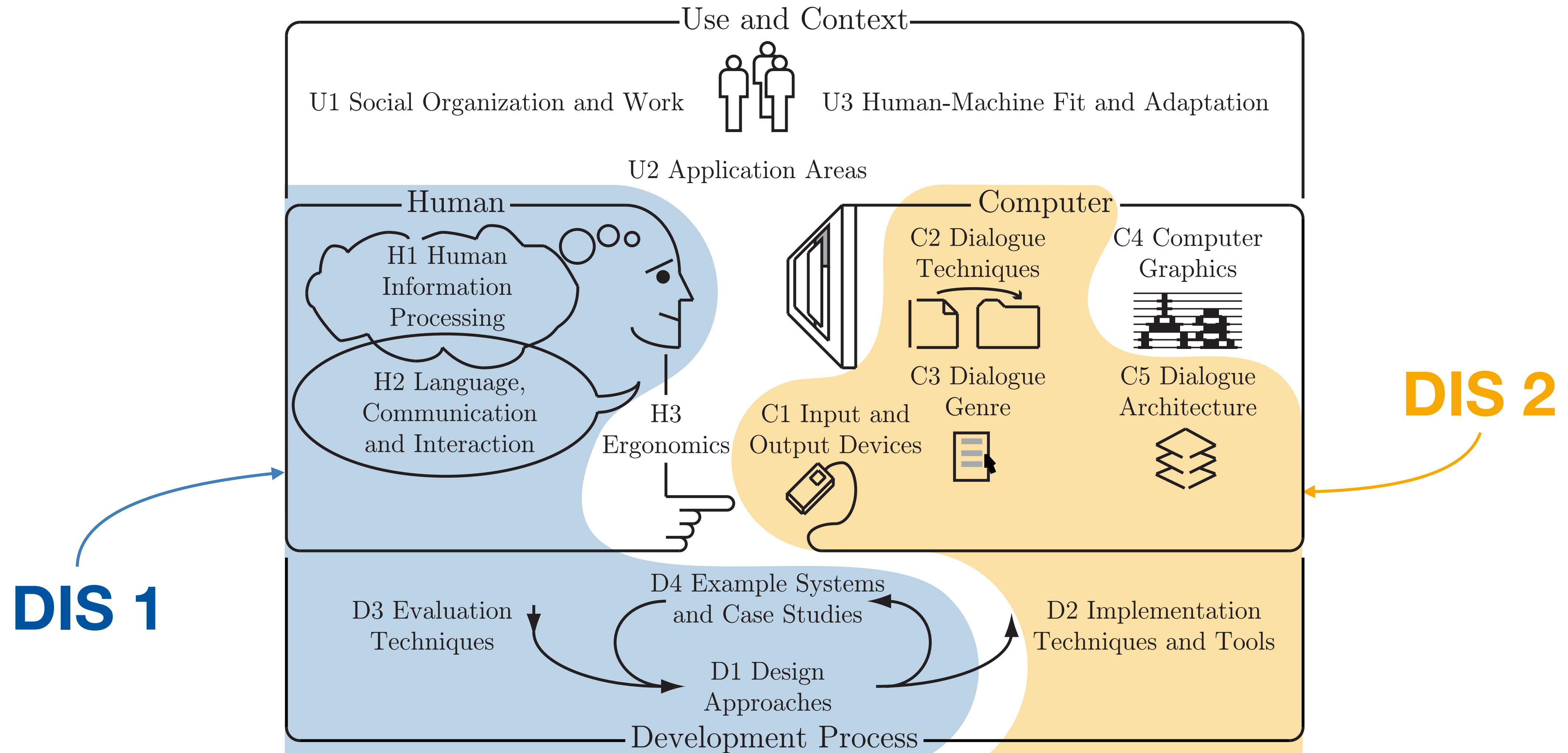
Lecture 1: Introduction, History, Design Space of Input Devices

Prof. Dr. Jan Borchers

hci.rwth-aachen.de/dis2

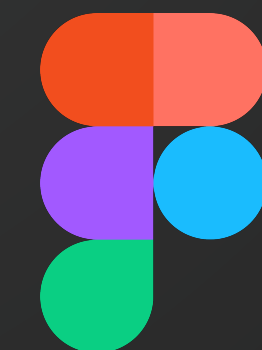
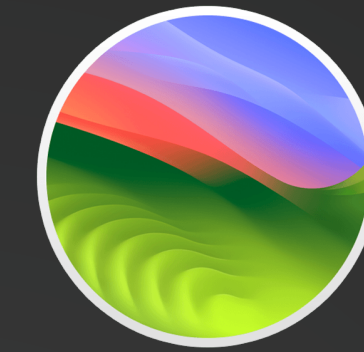


How DIS1 and DIS2 Cover HCI



DIS2 Class Syllabus

- Part 1
Key Concepts
- Part 2
Usage and Design of UI Toolkits and Design Systems
- Part 3
UIs Beyond the Desktop
- Part 4
Prototyping Process



Administrivia

- Format: Lecture+Lab (V3/Ü2)
- 6 ECTS credit points
- Class times
 - **Lecture** on Wednesdays (9:30–12:00)
 - **Lab** on Mondays (14:30–16:00)



Team



**Prof. Dr.
Jan Borchers**



Kevin Fiedler

kfiedler@cs.rwth-aachen.de
E-Mail Subject: [DIS 2]



Oliver Nowak

nowak@cs.rwth-aachen.de
E-Mail Subject: [DIS 2]

The Question Flow Chart

Does it contain matters of personal concern?

No (Default)

Yes

RWTHmoodle Forum

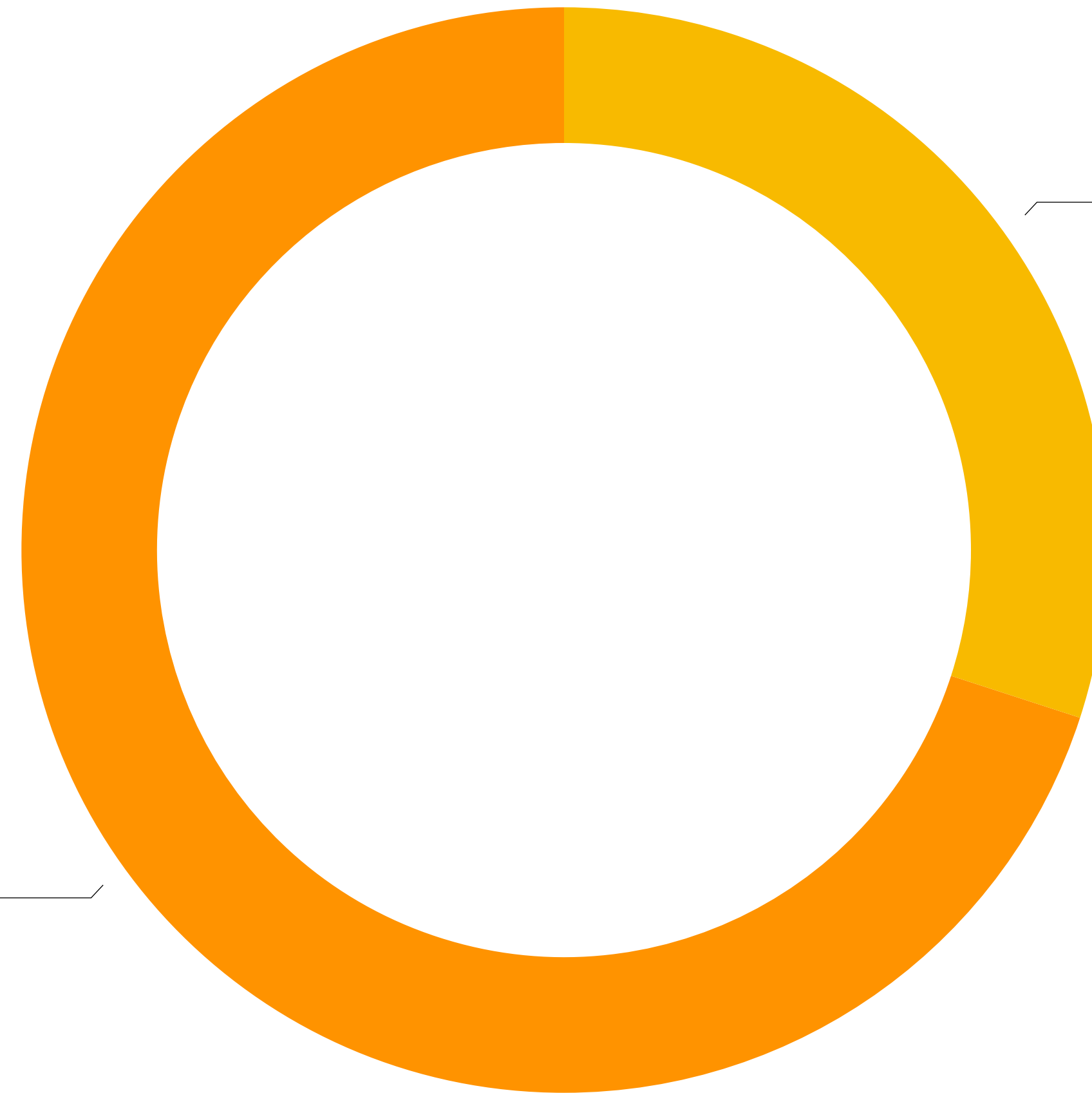
Email with the subject prefix **[DIS 2]**
to Oliver & Kevin (Not Jan 😊)

Alternatively: A quick chat after the lecture ☕

Your Final Grade

Final Exam (60 min)
July, 24th
or August, 27th

70%



30%

Project (5 weeks)
2 people



Note: You can only deregister from the course up to three days before the final project milestone is due (May 26).

Assignments & Project

- Assignments distributed weekly
 - You present your approaches in the lab
 - For some assignments you will need a **Mac**
 - No Mac? Contact Kevin and Oliver
- Project starts on April 22nd
 - 5 weeks, groups of two
 - You receive milestone documents instead of assignment sheets
 - Weekly milestone deadlines
 - Milestones discussed in the lab

You cannot deregister from the exam after three days before the final project milestone is due (May 26).

To pass the course, you have to pass the final exam and the project.

Who Are You?

- Audience
 - M.Sc. Computer Science / Media Informatics / Software Systems Engineering
 - B.Sc. / M.Sc. Technical Communication
 - B.Sc. / M.Sc. Human-Technology Interaction and Communication
 - B.Sc. Computer Science, ...
- Prerequisite: **Designing Interactive Systems (DIS1)** strongly recommended



Limited Seats

- **42 seats** available
- Register in RWTHonline by the end of **today**
- Seats will be assigned **tomorrow**
- **Sign the Declaration of Compliance document** and upload it to the Sciebo folder (all on the class website) as a PDF using this naming scheme:

Lastname-Firstname.pdf
(Example: Nowak-Oliver.pdf)

Deadline: Today, 10.04.24, 23:59

Course Website

- All information about this course can be found online
- hci.rwth-aachen.de/dis2

The screenshot shows a web browser window displaying the course website for 'Designing Interactive Systems 2 (2024)'. The page features a header with the RWTH Aachen logo and the course title. Below the title, there is a detailed description of the course, its objectives, and contact information for the instructors: Prof. Dr. Jan Borchers, Kevin Fiedler, and Oliver Nowak. On the right side, there are three summary boxes: 'Class Information' (Lecture: Wed, 9:30-12:00; Lab: Mon, 14:30-16:00; Language: English; Credits: 6), 'Exams' (Final PT1: Jul 24; Final PT2: Aug 27), and 'Resources' (RWTHonline, Moodle, YouTube Playlist).

Designing Interactive Systems 2 (2024)




This course builds on the foundations of [Designing Interactive Systems I](#) and provides an understanding of how interactive systems are built from a computer science point of view. It covers the principles of event-based operating systems, window system architectures, input and output device technology for multiple modalities, including audio and haptics, User Interface Management Systems, and UI toolkits and their relative merits. In our labs, you will develop a simple window system yourself but also learn to develop user interfaces with various real-life platforms, including Windows, macOS, Qt, and embedded platforms like Arduino.

After this class, you will know how the technology behind interactive systems works. You can analyze, design, and implement graphical and other user interfaces for existing and emerging technologies, both for the desktop and beyond and including interfaces for multimedia content. Group-based, project-centered assignments and lab activities convey hands-on experience in building user interfaces and foster project management and teamwork skills.

The class consists of weekly lectures, labs, reading assignments, project-based group assignments, and written examinations.

This course has limited seating. You need to [register](#) to obtain a seat in this course.

Contact

  
Prof. Dr. Jan Borchers Kevin Fiedler Oliver Nowak

For any questions about the class, please contact both [Kevin Fiedler](#) and [Oliver Nowak](#) (in English or German).

Class Information

Lecture	Wed, 9:30-12:00
Lab	Mon, 14:30-16:00
Language	English
Credits (BSc/MSc CS)	6

Exams

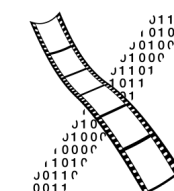
Final PT1	Jul 24
Final PT2	Aug 27

Resources

- [RWTHonline](#)
- [Moodle](#)
- [YouTube Playlist](#)

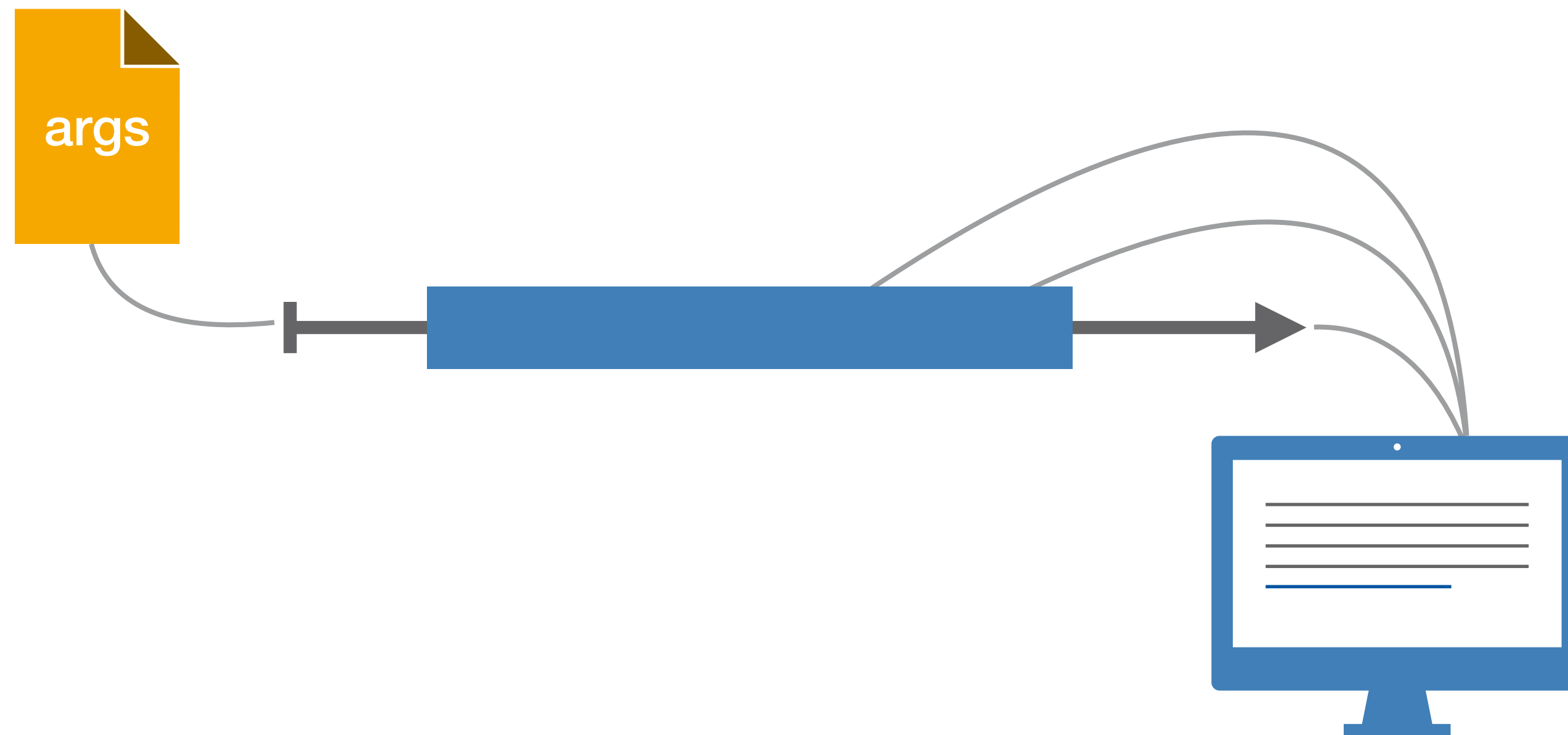
CHAPTER 1

Evolution of User Interface Programming Paradigms



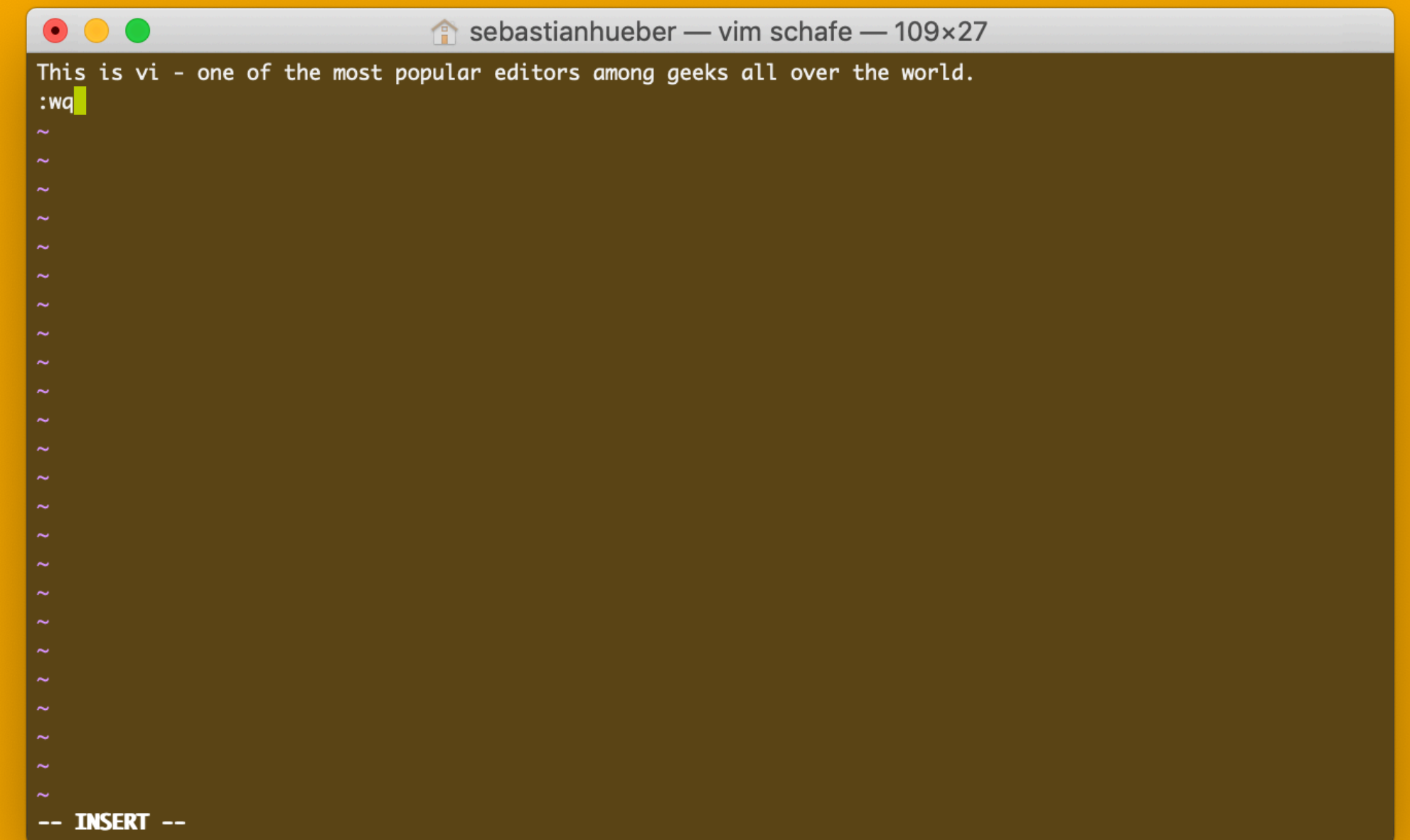
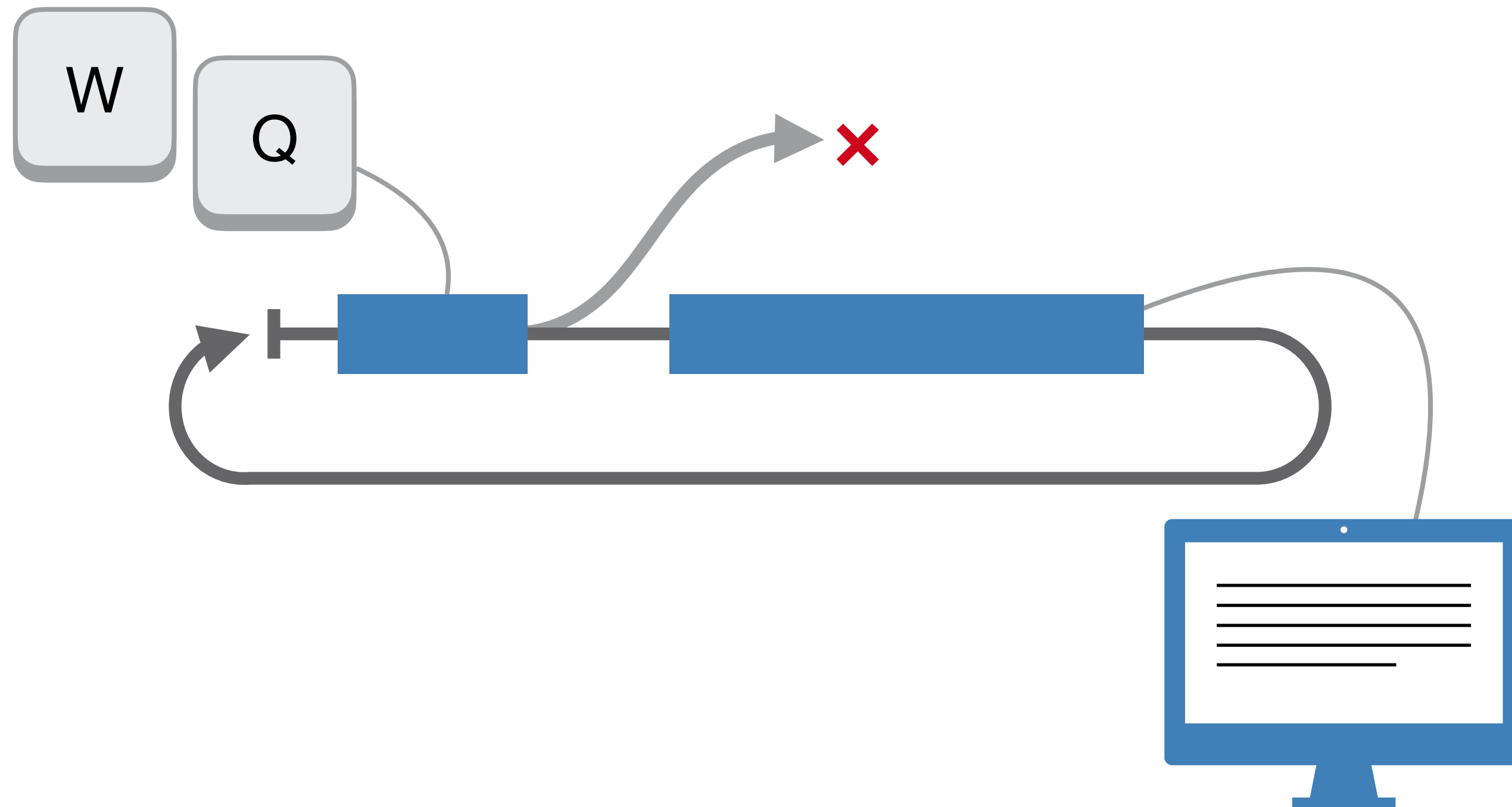
Time-sharing Systems

- Command line interaction
- Shorter turnaround (per line)



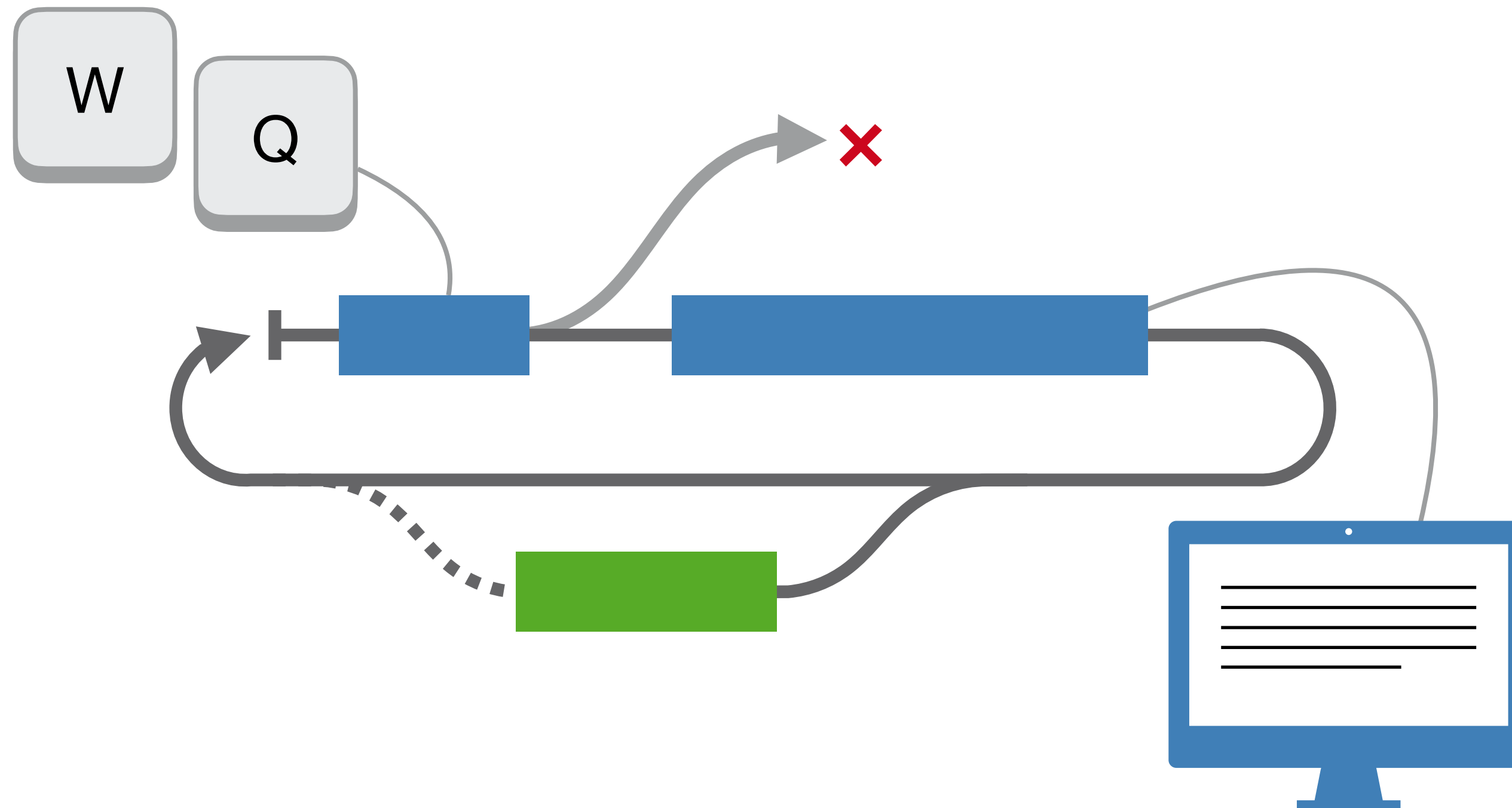
Full-screen textual UIs

- Turnaround per character
- Interaction starts to feel “real time”



Menu-based Systems

- Apps have UI component
- But: menu hierarchy, app still "in control"
- **Threading** becomes important



```
GNU nano 2.8.6 File: /Library/Python/2.7/site-packages/vboxapi/_init__.py
# -*- coding: utf-8 -*-
# $Id: vboxapi.py 101359 2015-06-30 22:28:00Z bird $
"""
VirtualBox Python API Glue.
"""

__copyright__ = \
"""
Copyright (C) 2009-2015 Oracle Corporation

This file is part of VirtualBox Open Source Edition (OSE), as
available from http://www.virtualbox.org. This file is free software;
you can redistribute it and/or modify it under the terms of the GNU
General Public License (GPL) as published by the Free Software
Foundation, in version 2 as it comes in the "COPYING" file of the
VirtualBox OSE distribution. VirtualBox OSE is distributed in the
hope that it will be useful, but WITHOUT ANY WARRANTY of any kind.
"""
__version__ = "$Revision: 101359 $"

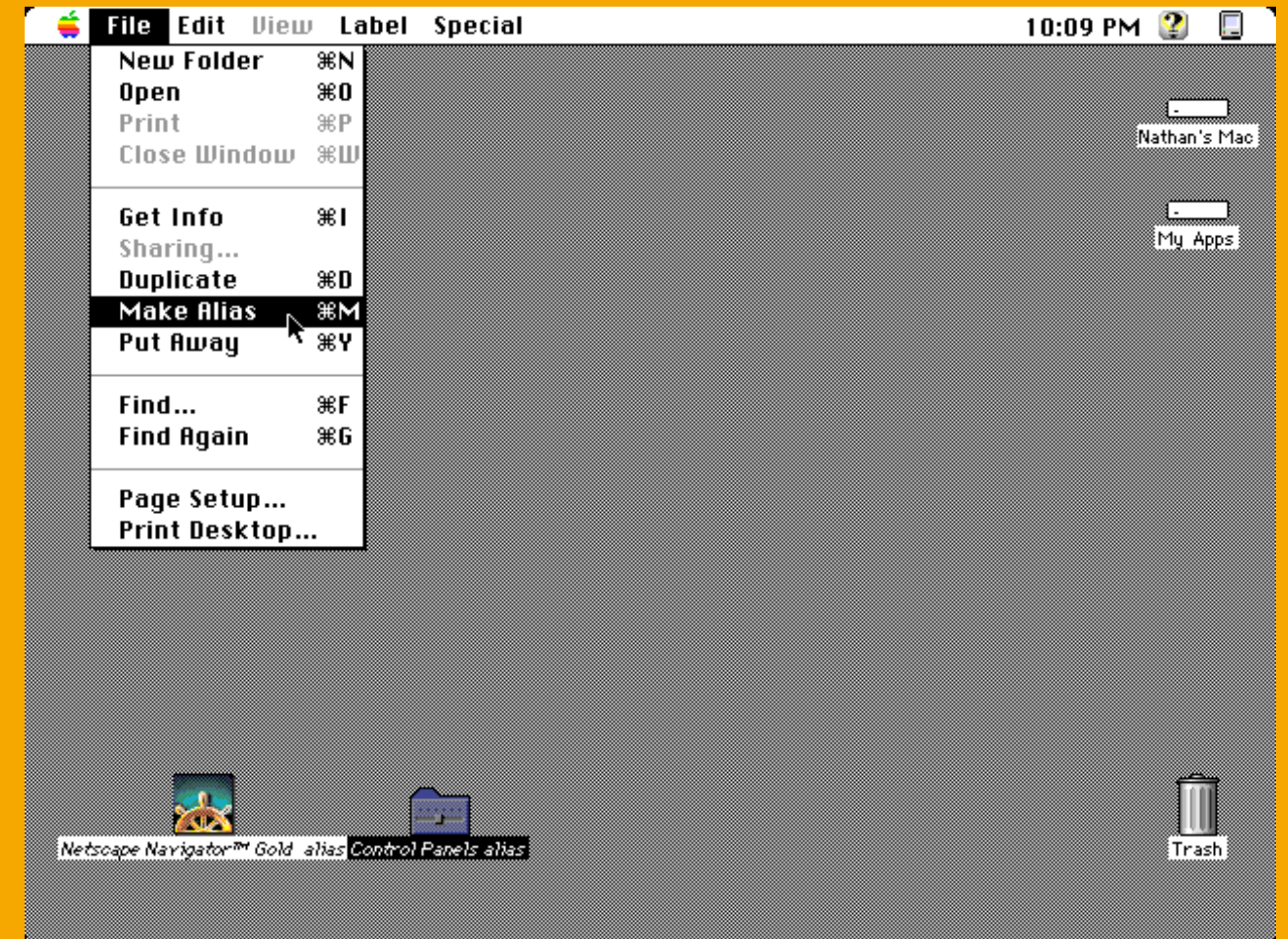
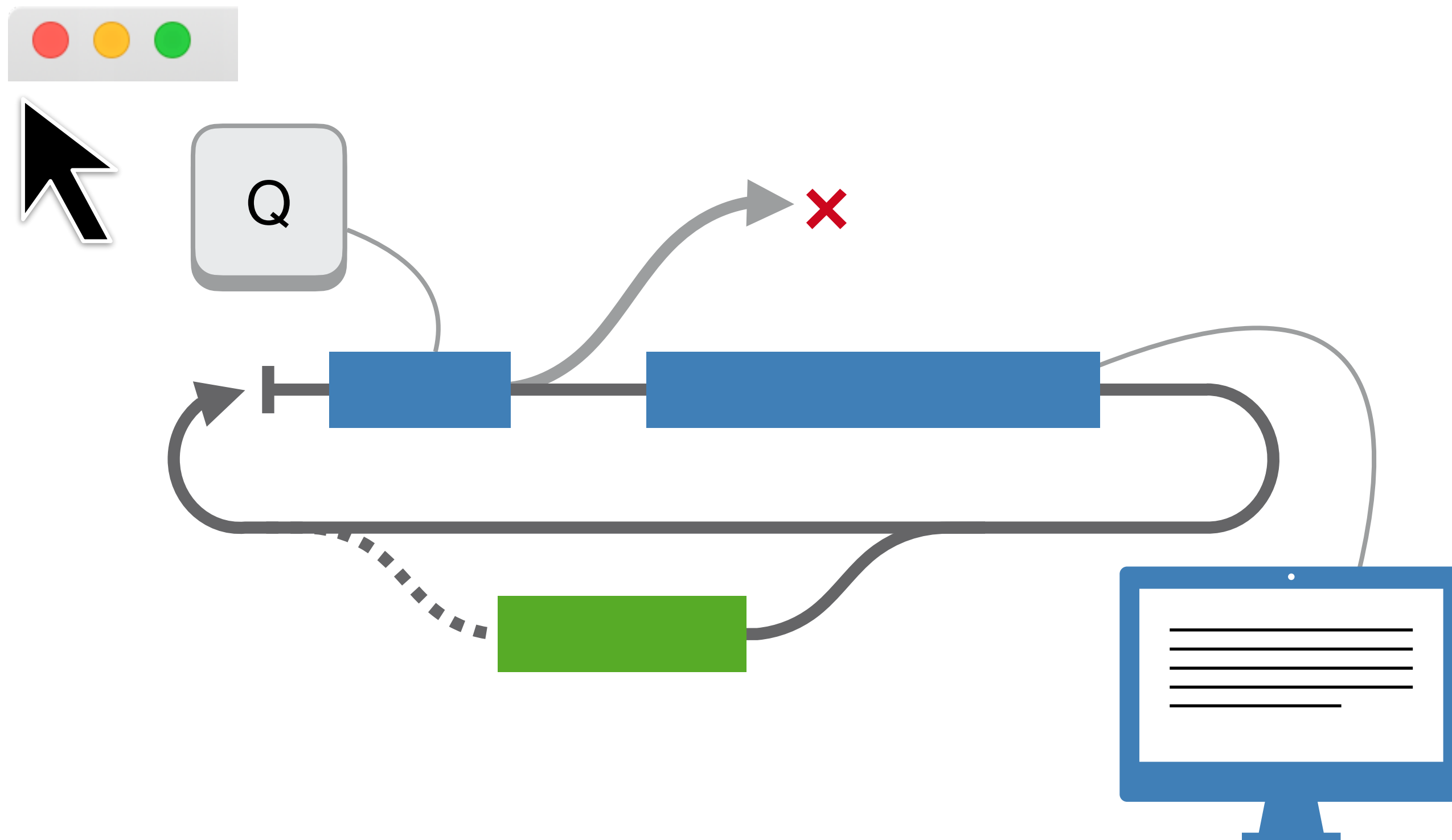
# Note! To set Python bitness on OSX use 'export VERSIONER_PYTHON_PREFER_32_BIT=yes'

^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit          ^J Justify      ^W Where Is    ^V Next Page    ^U UnCut Text   ^T To Spell
```



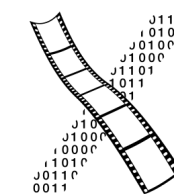
Graphical User Interfaces

- Pointing devices in addition to keyboard
- Event-based program structure
- User in control, application reacts in **callbacks**



CHAPTER 2

Design Space of Input Devices



Design Space of Input Devices

- Card, Mackinlay, Robertson 1991
- Categorization of input devices according to physical, mechanical and spatial properties
- **Why?**
 - Compare input devices
 - Identify new input modalities

Reading assignment
in Moodle



Movement Primitives

Manipulation operator

Device state

Output domain

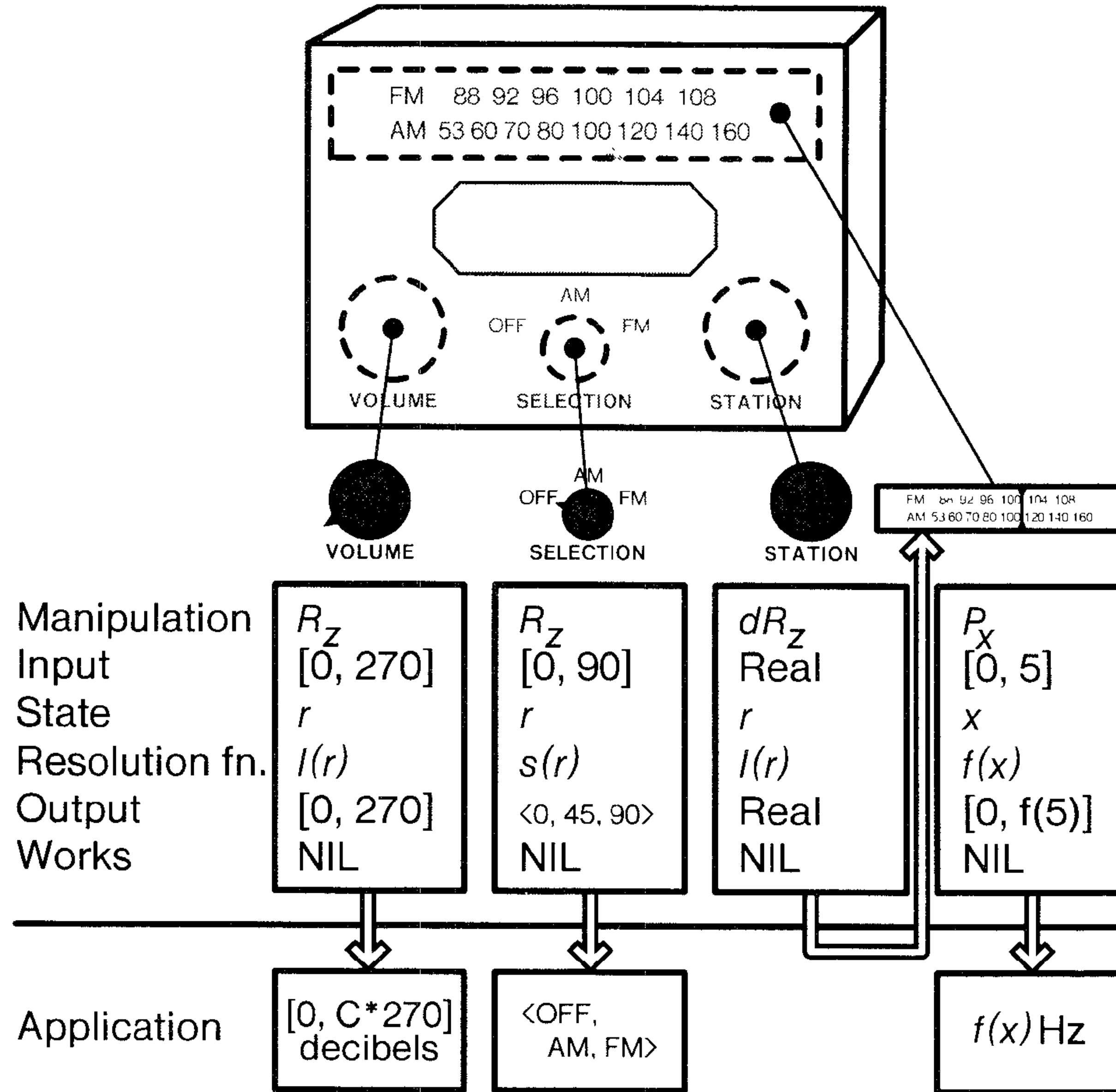
Input device := $\langle \mathbf{M}, \mathbf{In}, \mathbf{S}, \mathbf{R}, \mathbf{Out}, \mathbf{W} \rangle$

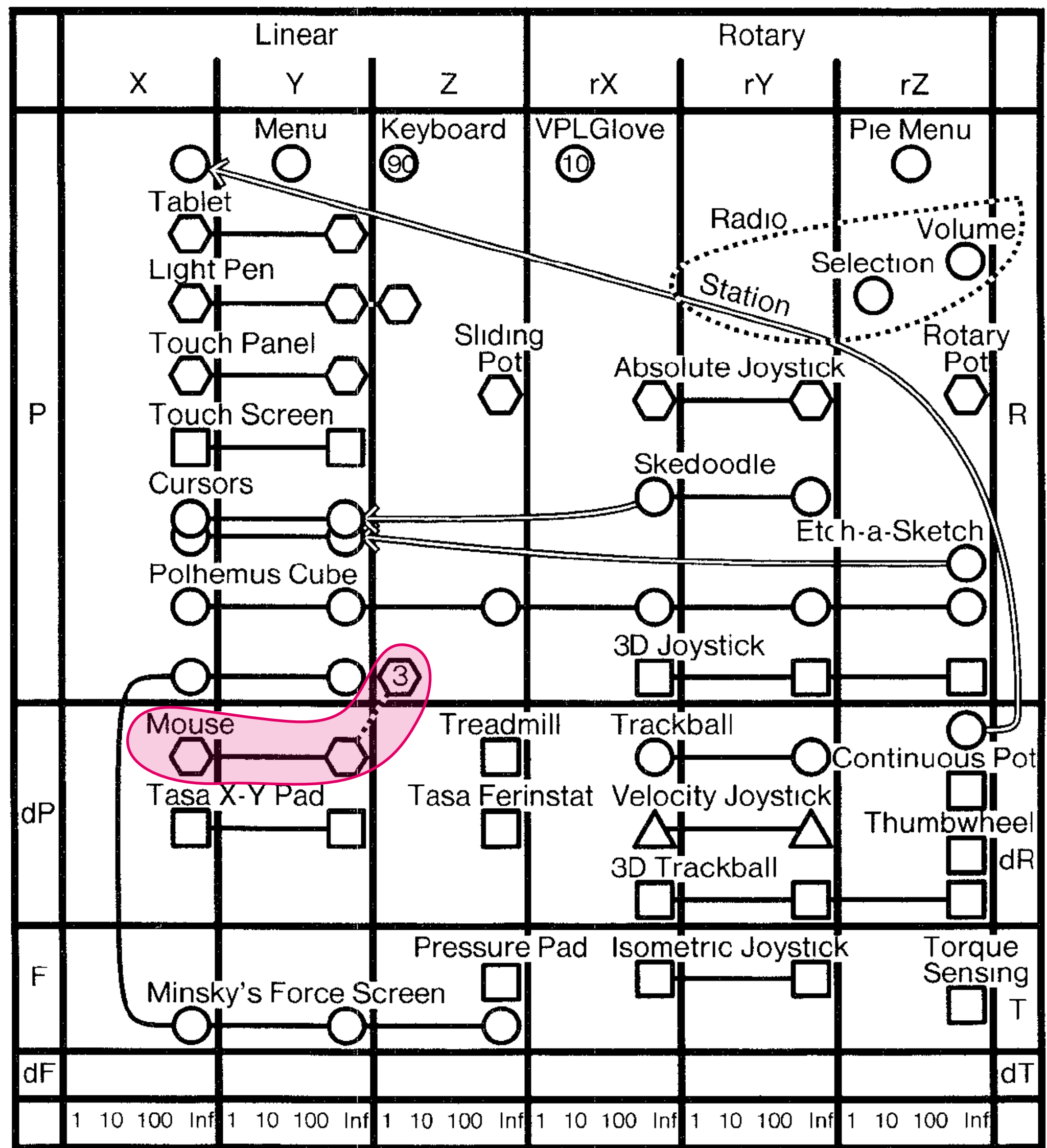
Input domain

Resolution function

Work properties

Example





△ Only Foley □ Only Buxton ⬡ Both Foley & Buxton ○ Other



In-Class Exercise

- Plot out the input capabilities of the Ferrari Racing Controller on the Card Design Space of Input Devices.
- The controller consists of a **steering wheel** with **8 buttons** and a **rotary switch** with 5 states, as well as **2 pedals**.
- Assume that the steering wheel can only have one full rotation.



In-Class Exercise

	X	Linear			Rotar							
		Y	Z	rX	rY	rZ						
P			8				R					
dP							dR					
F							T					
dF							dT					
	1	Inf	1	Inf	1	Inf	1	Inf	1	Inf	1	Inf

1 switch with 5 states

8 buttons

2 pedals with an infinite number of states each

1 steering wheel with an infinite number of states

buttons (8 in total)

--- Layout



Is This Space Complete?



Testing Points

- **Expressiveness** describes how precisely the meaning is conveyed
- For input devices, expressiveness suffers if $|In| \neq |Out|$
 - $|In| < |Out|$: Cannot specify all legal values
 - $|In| > |Out|$: Can specify illegal values

Testing Points

- **Effectiveness** describes how well the intention can be communicated

	Linear												Rotary												
	X				Y				Z				rX				rY				rZ				
P																									R
dP																									dR
F																									T
dF																									dT
	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	1	10	100	Inf	

CHAPTER 3

Window System Architecture



Window Systems: Basic Tasks

- **Input handling**
Pass user input to appropriate application
- **Output handling**
Visualize application output in windows
- **Window management**
Manage and provide user controls for windows



Window Systems: Requirements

- **Independent** of hardware and operating system
- No noticeable **delays** (few ms) for basic operations, e.g. moving window, redrawing cursor
- **Customizable** look&feel for user preferences
- Input & Output in **parallel**
- **Multimedia** support: Graphics, audio, ...
- Support for various **input devices** and modalities



Window Systems: Evaluation Criteria

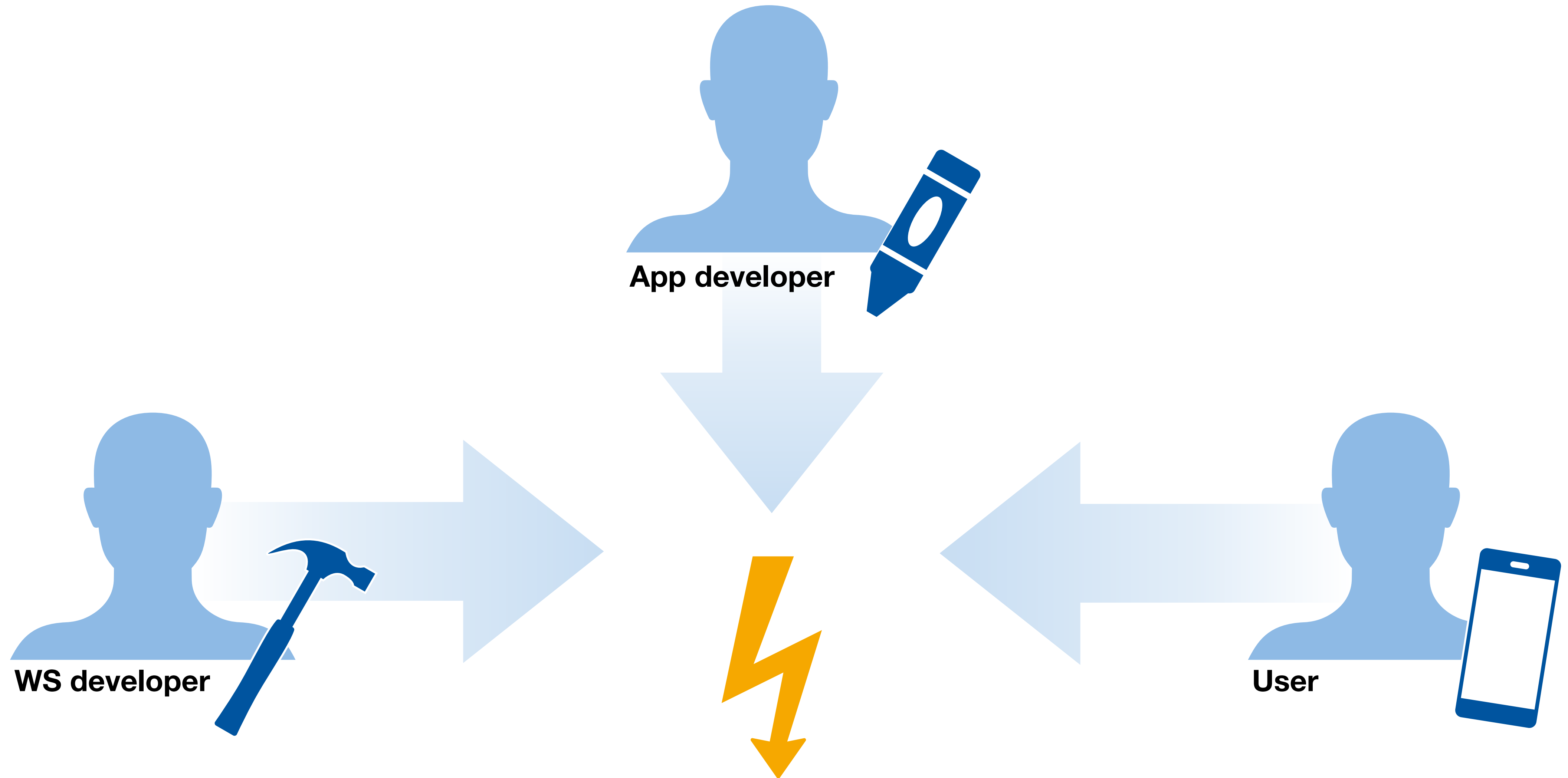
- **Availability**
Platforms supported
- **Productivity**
For application development
- **Parallelism**
External and internal
- **Performance**
Usage of resources and latency
- **Graphics model**
RasterOp vs. vector
- **Appearance**
Look & Feel, exchangeable?



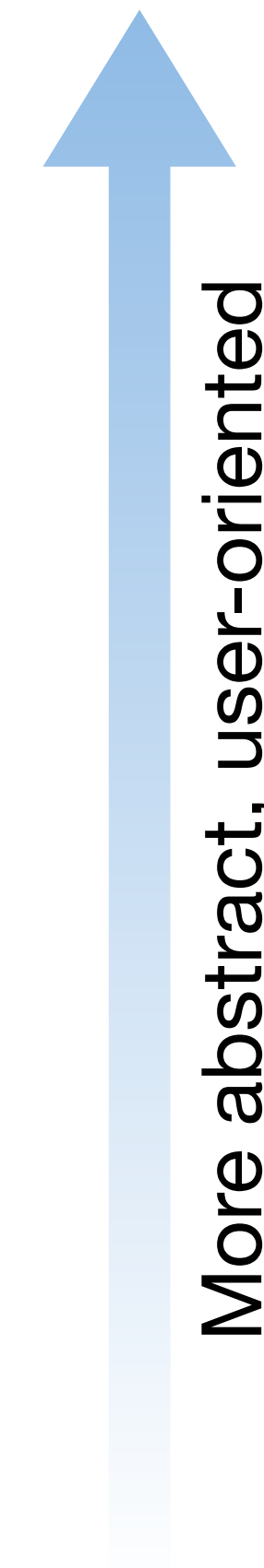
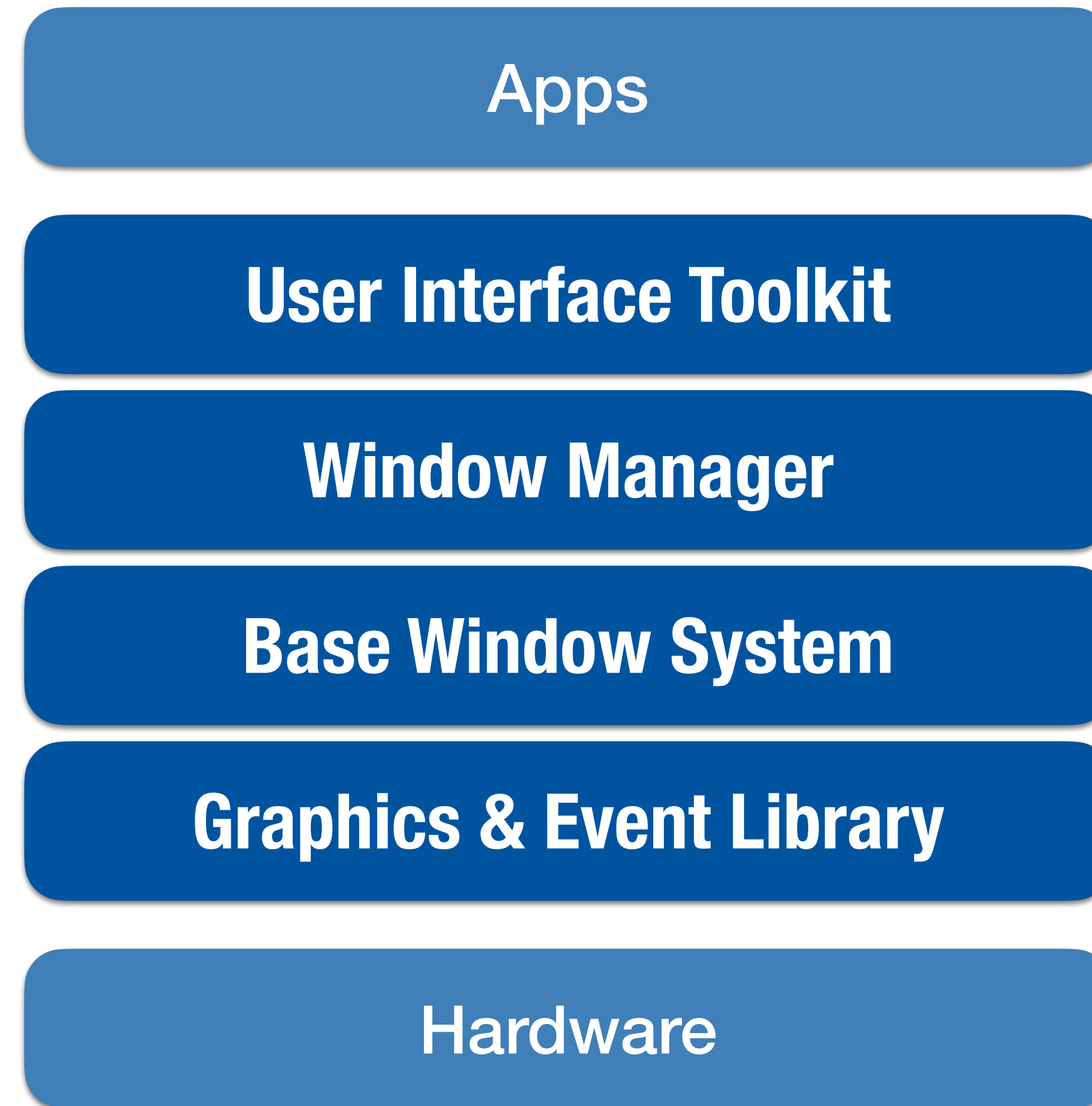
Window Systems: Evaluation Criteria

- **Extensibility**
In source code or at runtime
- **Adaptability**
Localization and customization at runtime
- **Resource sharing**
E.g., fonts
- **Distribution**
Over network
- **API**
Structure and comfort
- **Independence**
Of application and interaction logic inside programs written for the WS
- **Inter-Application Communication**
Copy & Paste, Drag & Drop

Window Systems: Conflict



Window System Architecture



What's Next?

- **Today**
 - Register in RWTHonline
 - Submit your signed Declaration of Compliance
 - Link on the website
- **This week**
 - Read the Design Space paper
 - Start with the first assignment
- We will do the seat allocation tomorrow

