

IN4MATX 133: User Interface Software

Lecture 17:
Small & Large Displays

Announcements

- A4 due Friday

- A5 is posted

- Intentionally less programming than previous assignments
- Timezone converter, can choose from three frameworks to implement it

Q1: How, if at all, does this framework support package and library management?

[Your answer here]

Q2: How, if at all, does this framework support principles for code separation, like Model-View-Controller? Would the separation principles effectively support creation of a larger application? Why or why not?

[Your answer here]

Q3: In lecture, we discussed a few design recommendations for each respective device. How, if at all, does the framework support the recommendations for that device? What recommendations are left to the developer to decide how or whether to implement?

[Your answer here]

Q4: What appear to be benefits to using web technologies for development on this device? Conversely, what advantages might a native framework have over the framework you used?

[Your answer here]

Q5: What did you find easy and challenging about development in this framework?

[Your answer here]

Announcements

- A3 grades will be posted later today
 - Average 8.62, stdev 2.67
- Remember regrade policy
- We are keeping an eye on overall course grades
 - We are likely to curve up very low assignment grades, we will check at the end of the quarter

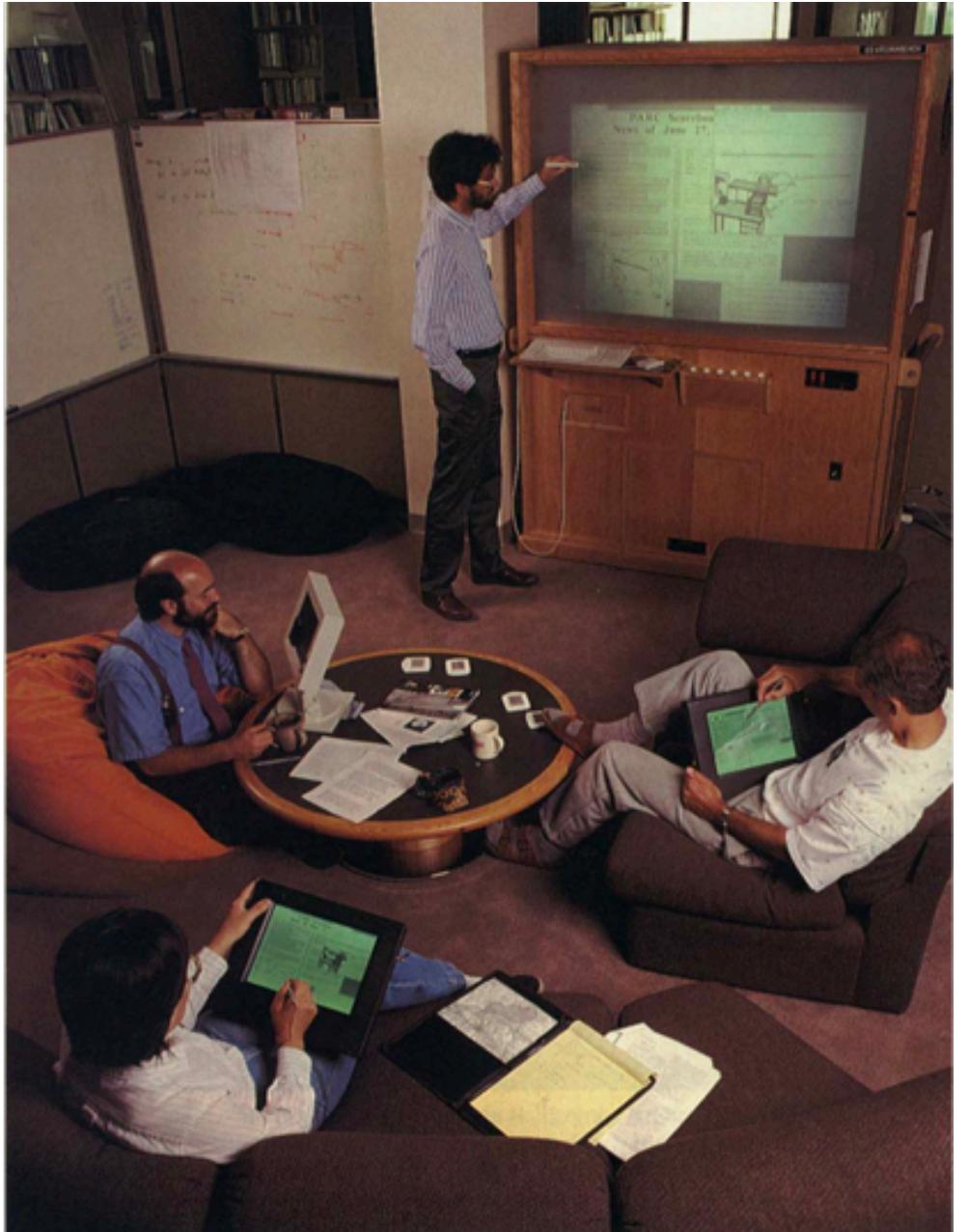
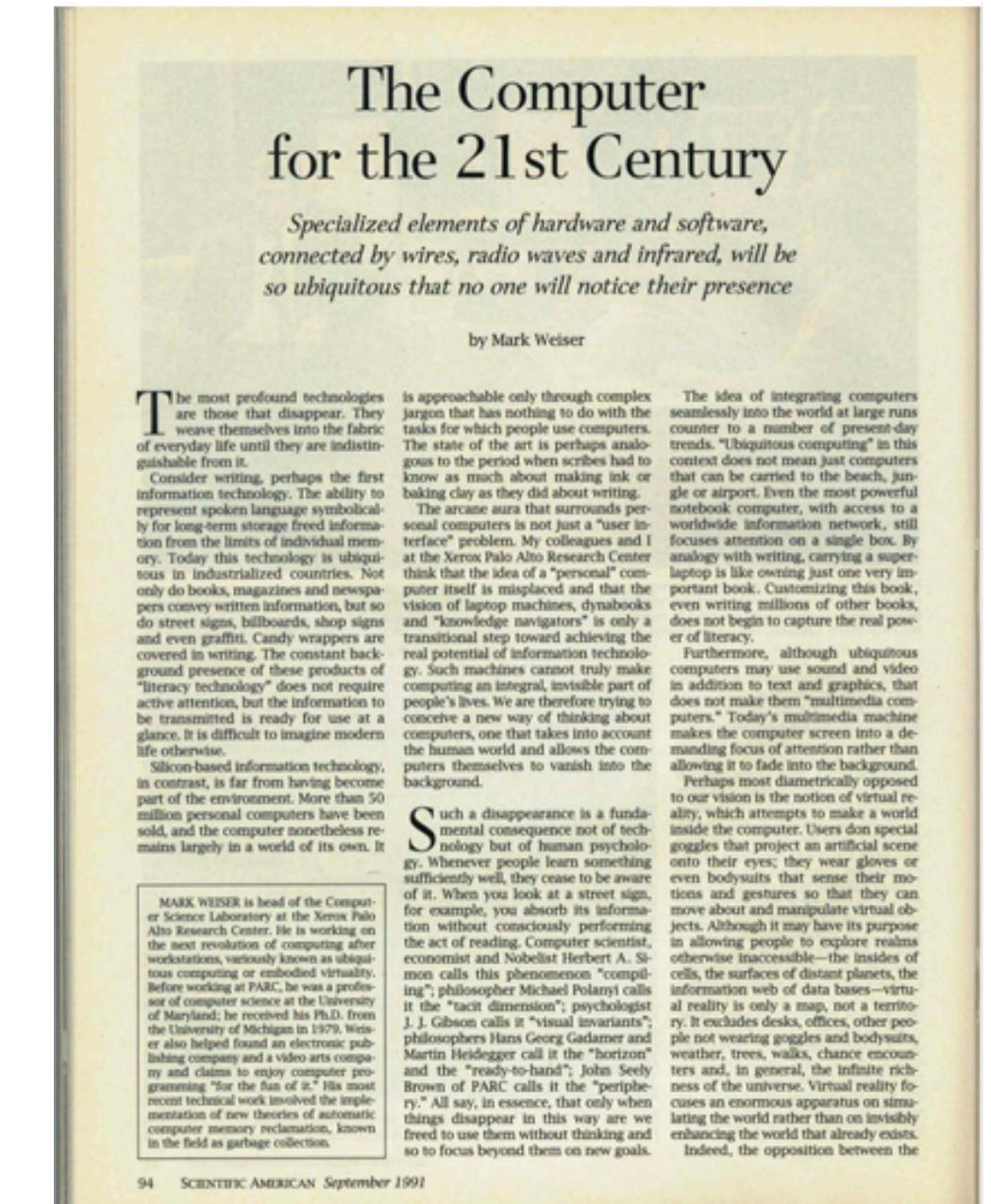
Today's goals

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

- Describe some of the history of wearable computing and large scale displays
- Explain key principles for designing a good wearable experience
- Articulate some principles for creating experiences with large displays
- Identify a few potential libraries for developing wearable and TV apps

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



What is a wearable computer?

A MUCH More Diversified Market Than Investors Realize



CREDIT SUISSE

What is a wearable computer?

- A computer on the body that is:
 - Always on
 - Always accessible
 - Always connected
- Other actions:
 - It augments user actions
 - Is aware of the user and their surroundings

Rhodes, B.J. 1997. The wearable remembrance agent: a system for augmented memory.
Personal Technologies, 1(4), 2018-224.

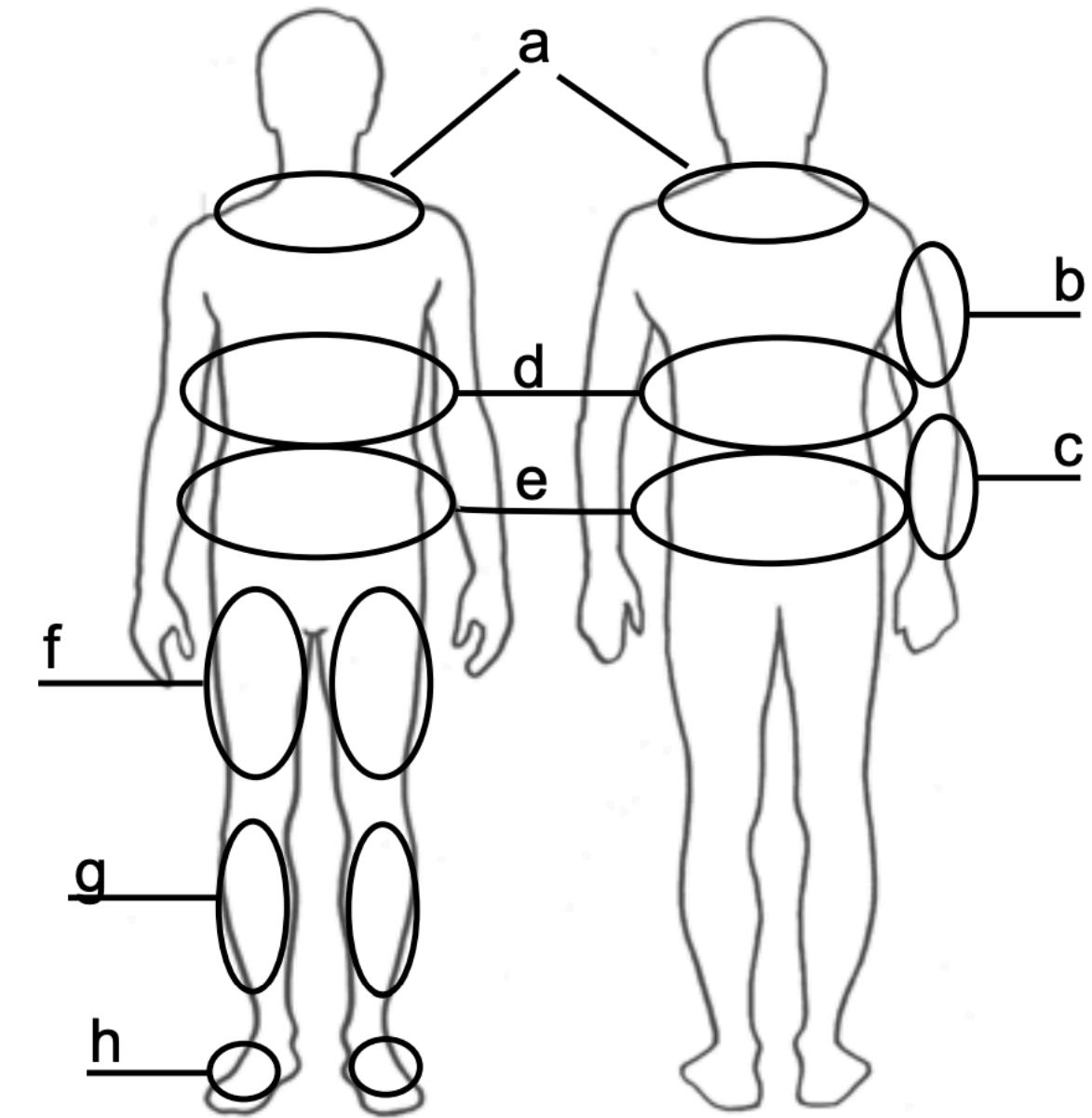
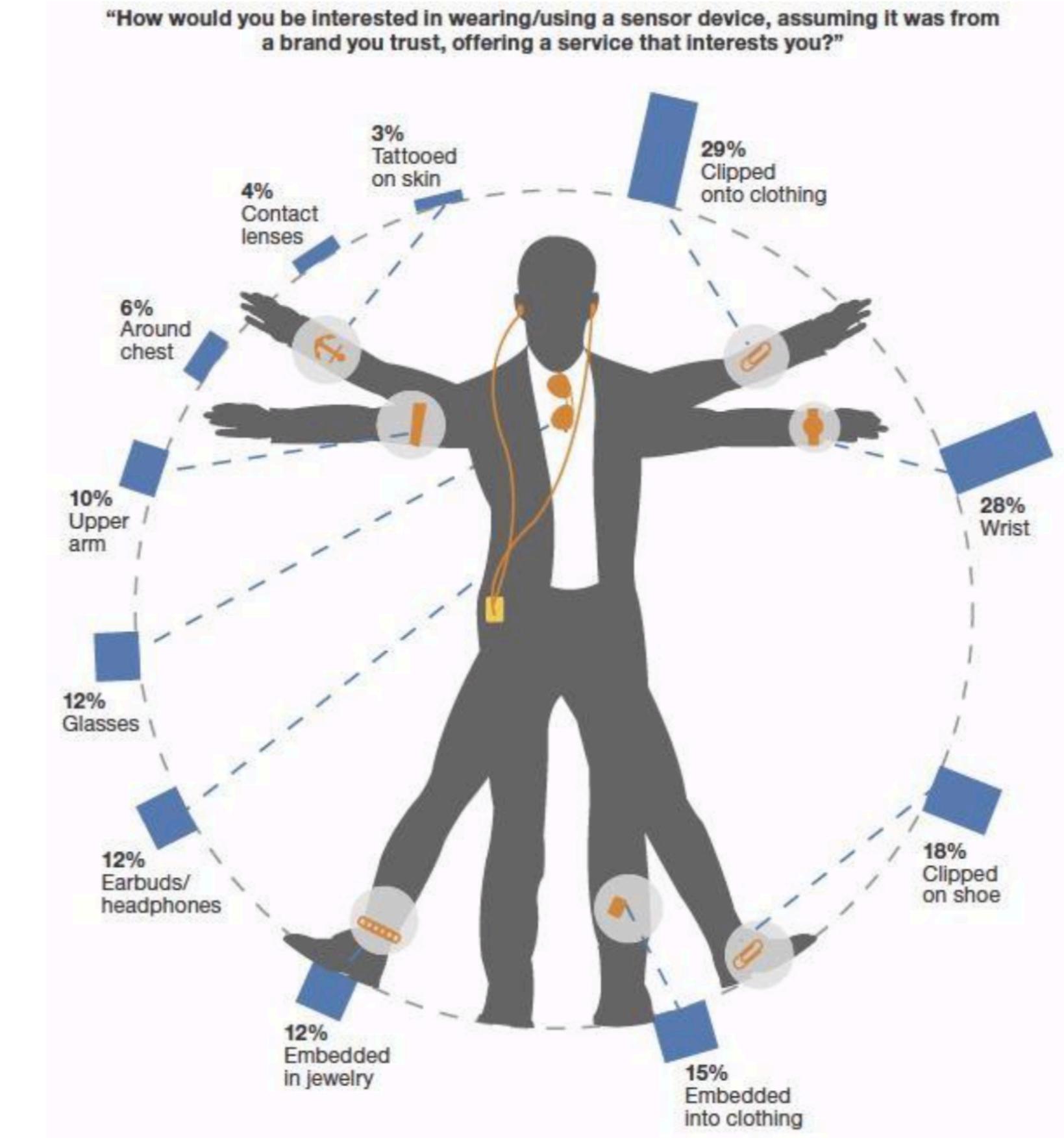


Fig. 1, The general areas we have found to be the most unobtrusive for wearable objects are: (a) collar area, (b) rear of the upper arm, (c) forearm, (d) rear, side, and front ribcage, (e) waist and hips, (f) thigh, (g) shin, and (h) top of the foot.

Gemperle, F. Kasabach, C., Stivoric, J. Bauer, M. Martin, R. Design for Wearability. ISWC 1998.



North American Technographics
Consumer Technology Survey, 2013.

Body and head-mounted wearables

MIT Wearable Computing (1996)



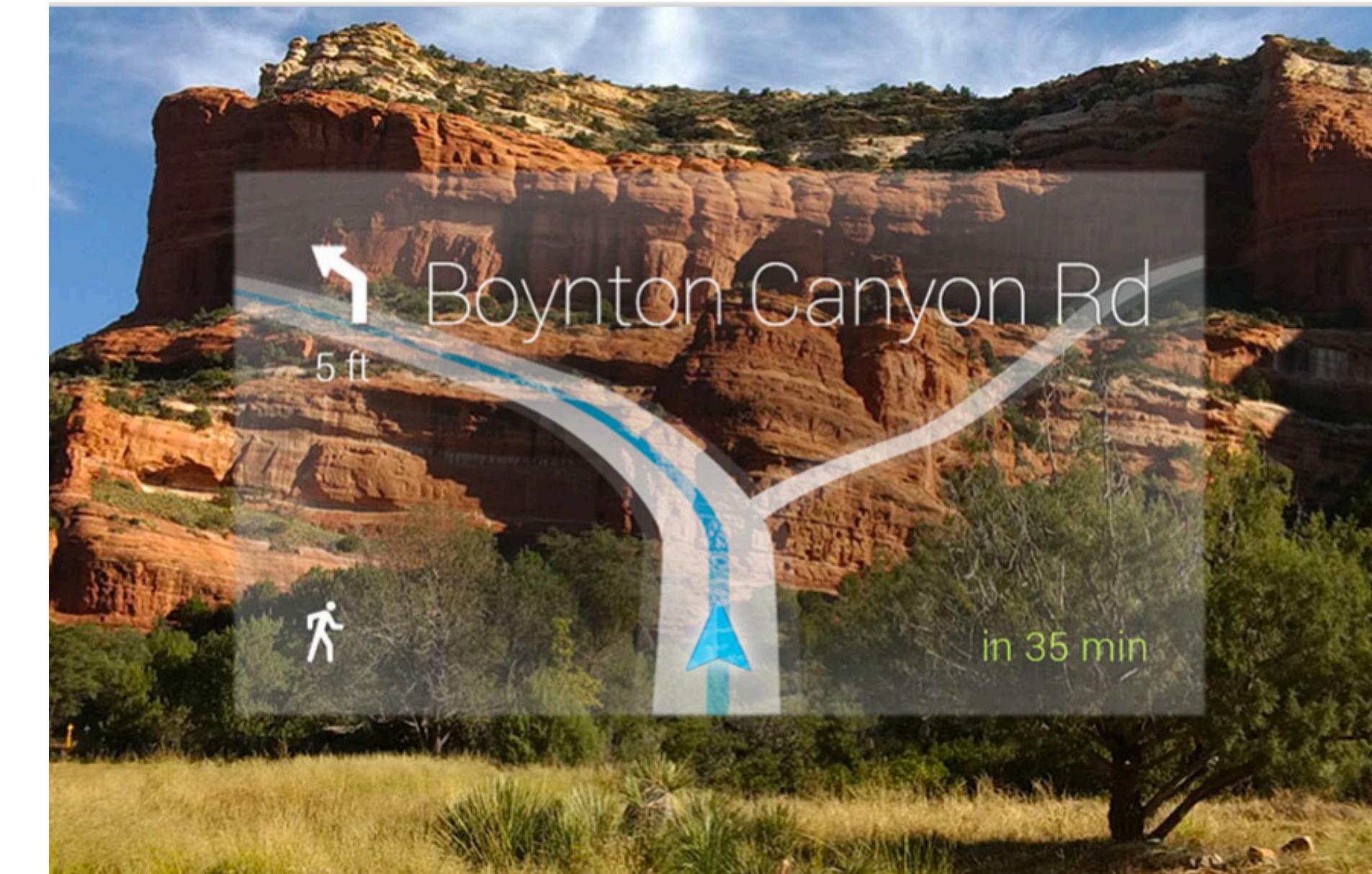
Google glass

- Commercial smart glasses, released in 2013
 - Technology lead by Thad Starner, part of the MIT group
- Front-facing camera, rear-facing display
- Gyroscope/accelerometer/magnometer
- Natural language input capabilities



<https://www.x.company/glass/>

Google glass



<https://www.x.company/glass/>

Google glass

- Privacy and safety concerns prevented take-off in the consumer space
- Lives on in enterprise spaces
 - New version released in 2017
 - Used in manufacturing, healthcare



<https://www.x.company/glass/>

Wrist-worn wearables

Fitbit (2011)

- One of the first commercially successful digital pedometers
- Early versions were hip-worn, now almost exclusively wrist-worn
- Current models are “fitness-first” smartwatches
 - Activity prominently included on the home screen
- Acquired by Google in 2019



Pebble (2013)

- Arguably the first commercially successful smartwatch
 - Two of the most funded Kickstarter projects ever
- E-ink display led to high battery life (a week vs. a day)
- Paired with a phone via Bluetooth
 - Could retrieve email, control music, receive notifications, etc.
- Acquired by Fitbit in 2016



Apple Watch (2015)

- From the onset, it was intended to be a “second screen” companion to iOS devices
- Original versions could do almost nothing without pairing to an iOS device
- Apps add a secondary component to an existing iOS app



Design recommendations for (wrist-worn) wearables

One visual thought per screen

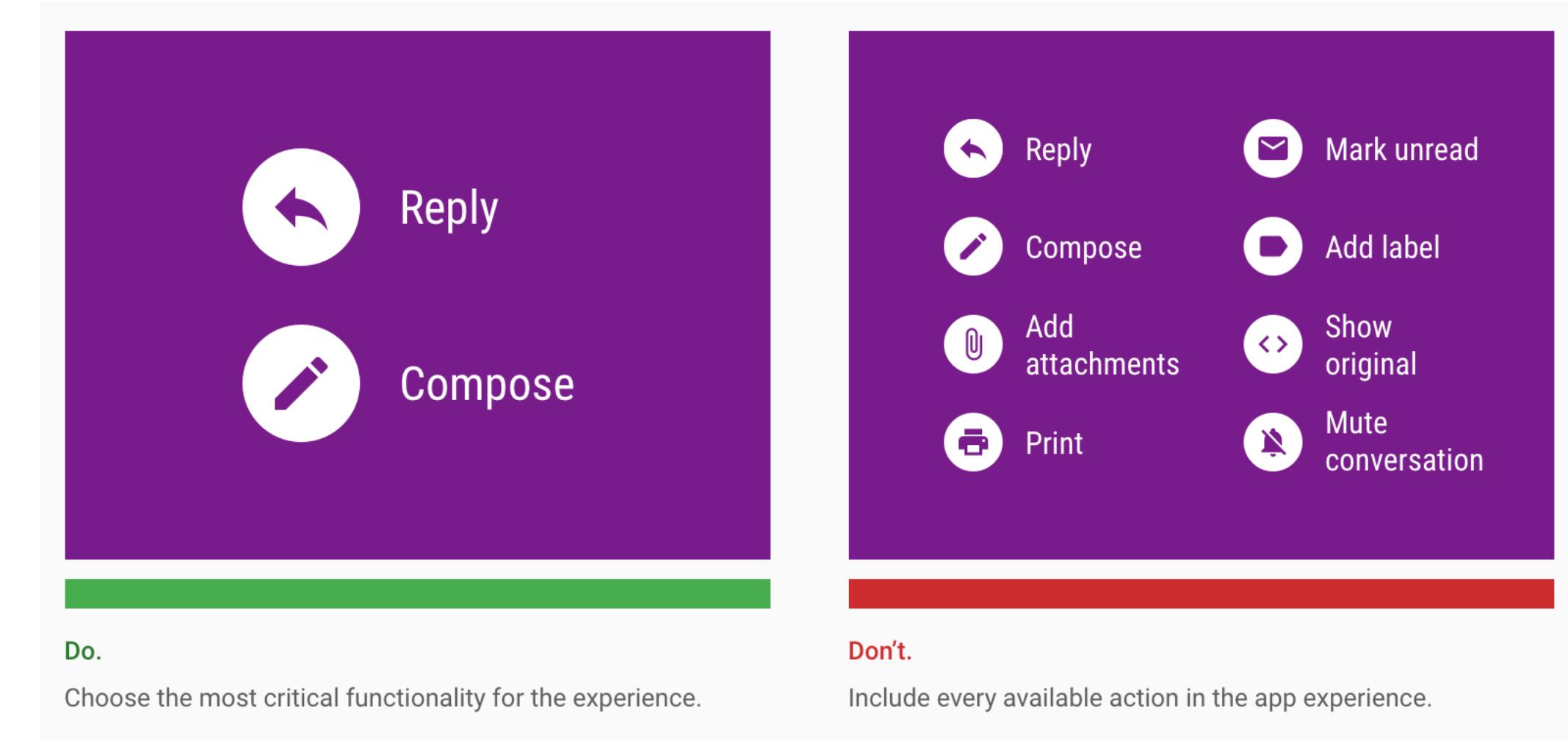
- Real estate is extremely valuable on watches
- Shrinking a mobile or desktop app will create a bad experience
- Keep words and interactions to a minimum



<https://mayvendev.com/blog/10-tips-for-designing-for-wearables-and-watches>

Reduce input options

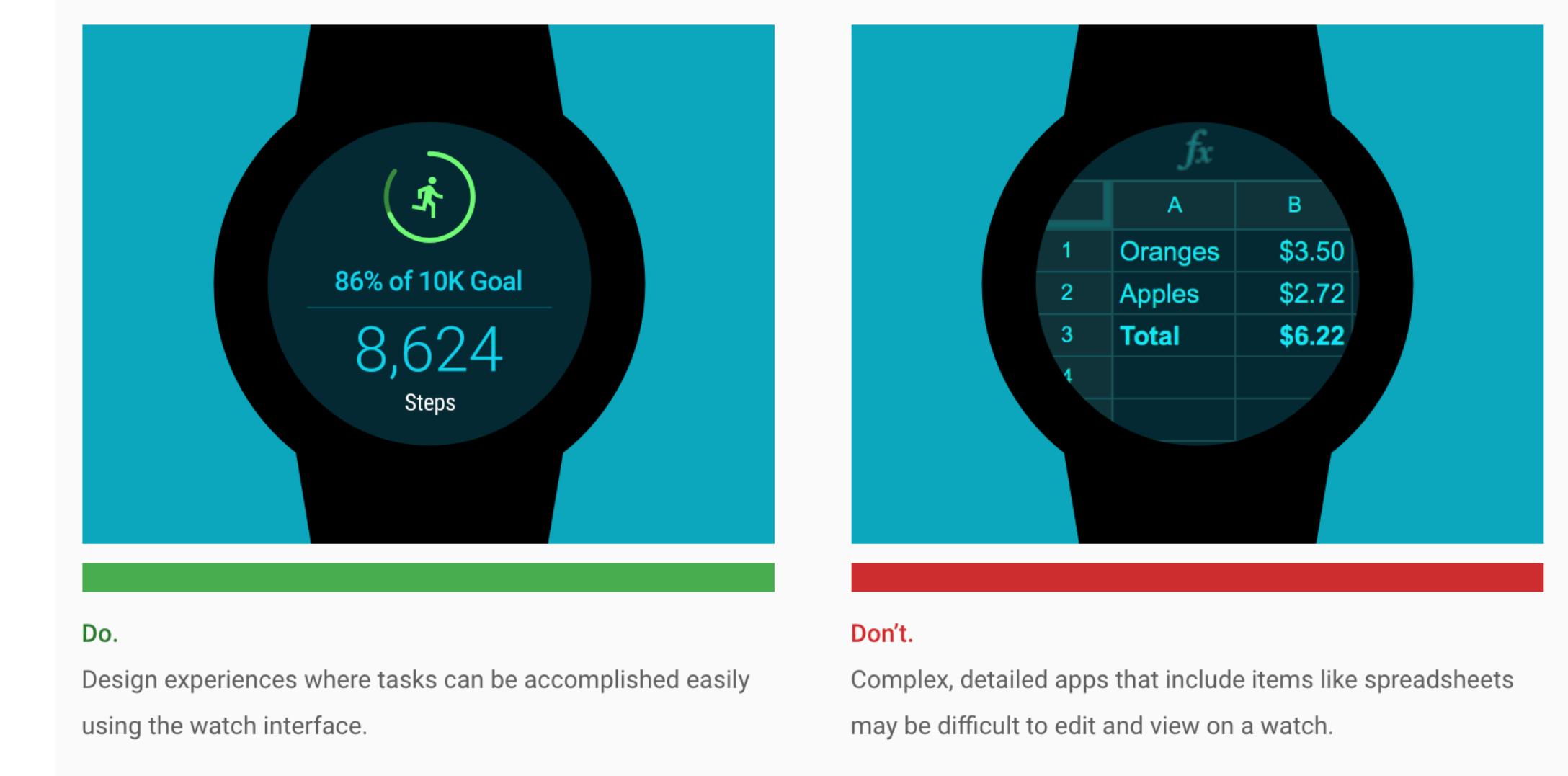
- Have only a couple of buttons per screen
- This may mean a watch app has fewer features than a mobile app
 - That's okay!
 - The watch augments the experience
- Consider voice input when longer instruction is needed



<https://desingguidelines.withgoogle.com/wearos/wear-os-by-google/designing-for-watches.html>

Some apps don't need a watch interface

- For some apps, a watch app may not add to the experience
- Focus on use cases which make sense
 - Quick input
 - Glanceable feedback



<https://designguidelines.withgoogle.com/wearos/wear-os-by-google/designing-for-watches.html>

Minimize depth of navigation

- Minimize the information hierarchy in your app
 - Number of pages/subpages
- Physical buttons can help with scrolling/switching between screens



<https://developer.apple.com/design/human-interface-guidelines/designing-for-watchos>

Question



These are all Apple Watch apps.
Which have a compelling use case?

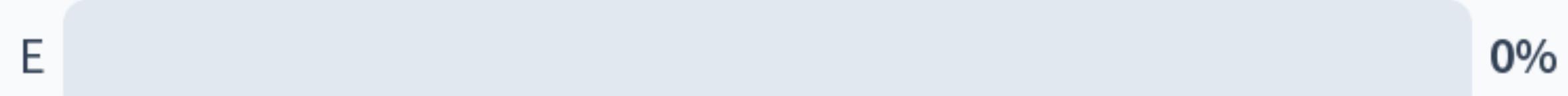
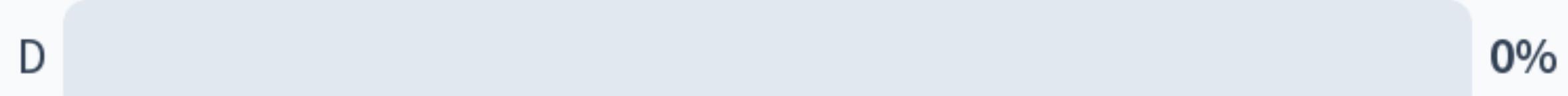
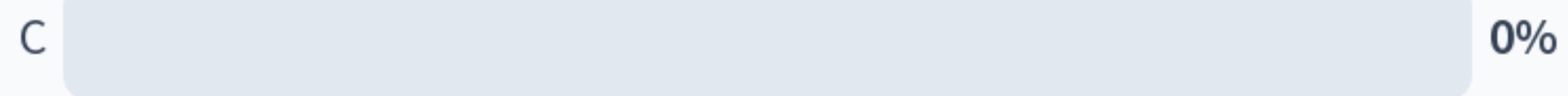
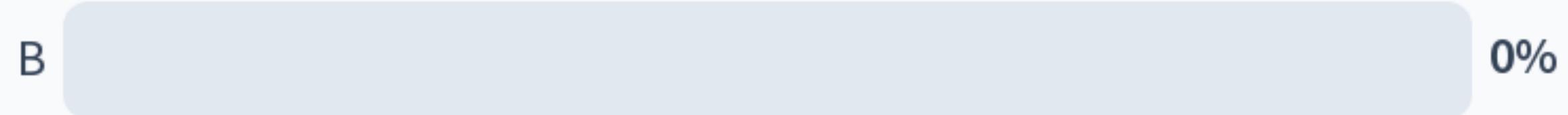
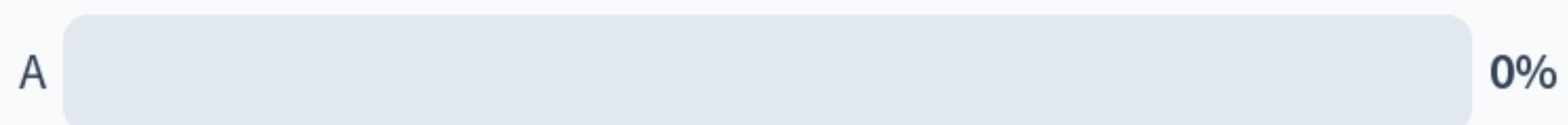
New York Times (news)
The Weather Channel (weather)
Fandango (movie purchase)
Strava (fitness)
OneNote (notetaking)

- A None of them have a compelling use case
- B One of them
- C Two of them
- D Three or four of them
- E All of these have a compelling use case

These are all Apple Watch apps. Which have a compelling use case?

- A None of them have a compelling use case
- B One of them
- C Two of them
- D Three or four of them
- E All of these have a compelling use case

New York Times (news)
The Weather Channel (weather)
Fandango (movie purchase)
Strava (fitness)
OneNote (notetaking)



Question



These are all Apple Watch apps.
Which have a compelling use case?

- A None of them have a compelling use case
- B One of them
- C Two of them
- D Three or four of them
- E All of these have a compelling use case

New York Times (news)
The Weather Channel (weather)
Fandango (movie purchase)
Strava (fitness)
OneNote (notetaking)

Open to interpretation,
I think NYT and Fandango
are a stretch but there are
potential uses

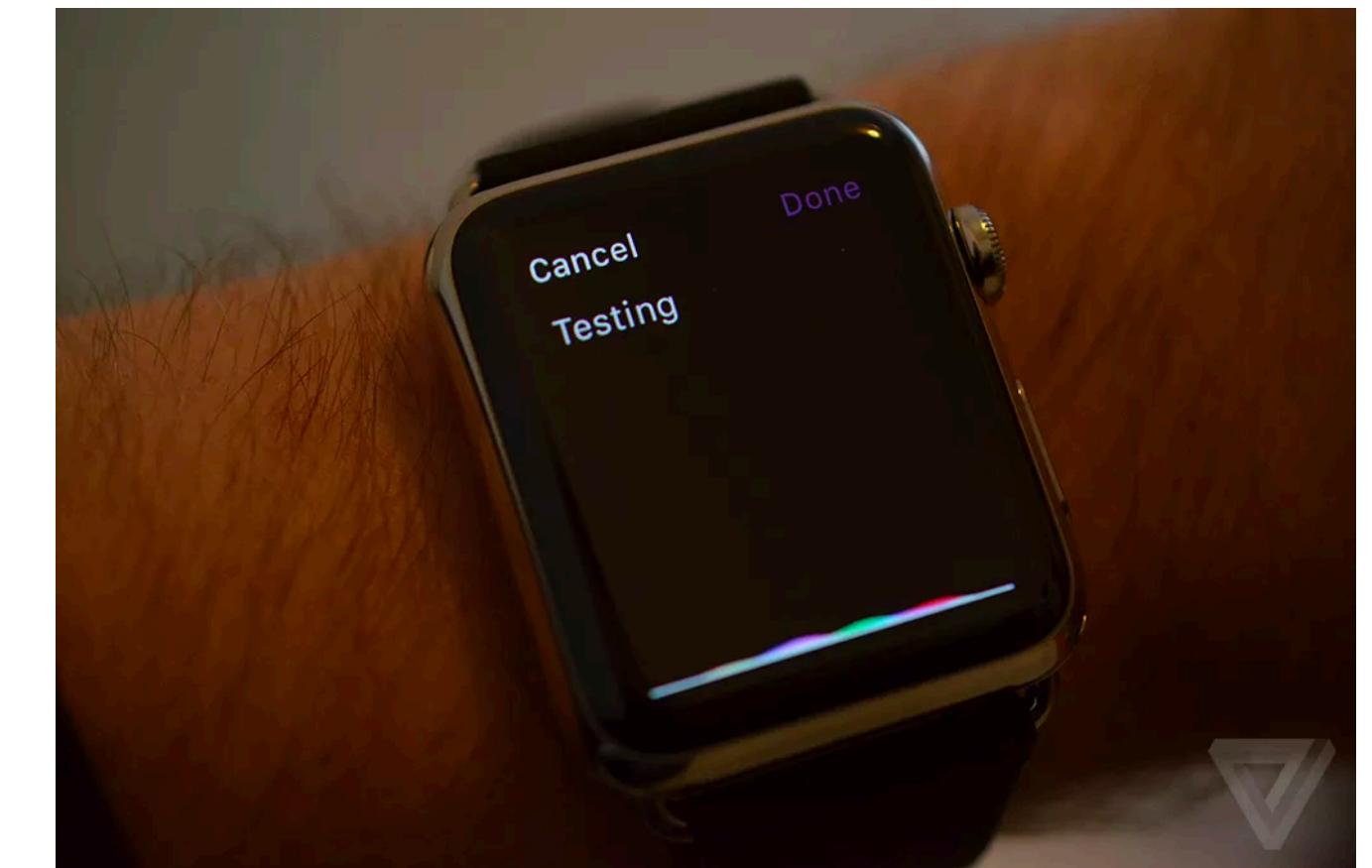
Microsoft apps for Apple Watch



Powerpoint remote



Skype messaging



OneNote voice memos

Are these compelling use cases?

<https://www.theverge.com/2015/4/28/8508445/microsoft-apple-watch-apps-hands-on>

Questions to consider

- Would a watch app add anything to my mobile app?
 - Is there timely information the app needs to provide?
 - Can it be shown in a very small format?
 - Are there simple controls to the app that would be added to a watch?
- Do I have the resources/time to do this?
 - Currently limited market impact, but growing
- What type of interaction do you want the user to have?

Implementing watch apps

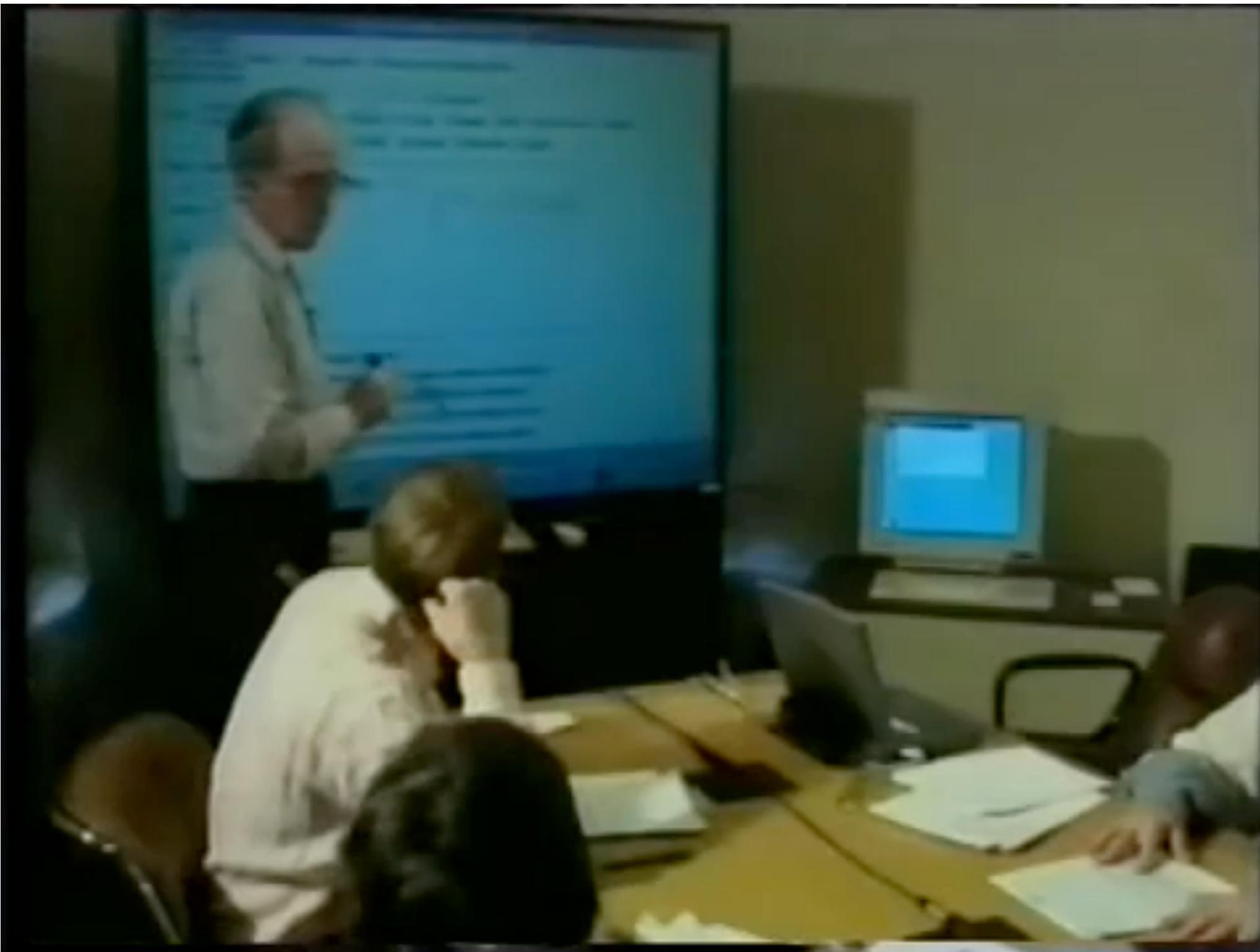
- Requires native development for many watches
 - WatchKit for iOS, Wear OS for Android
- Requires a companion iOS or Android app for building/deploying, though may be able to run as a standalone
- However, you can develop a hybrid mobile app and connect it to a native watch app

<https://developer.android.com/training/wearables/apps>

<https://developer.apple.com/documentation/watchkit>

Large-scale displays

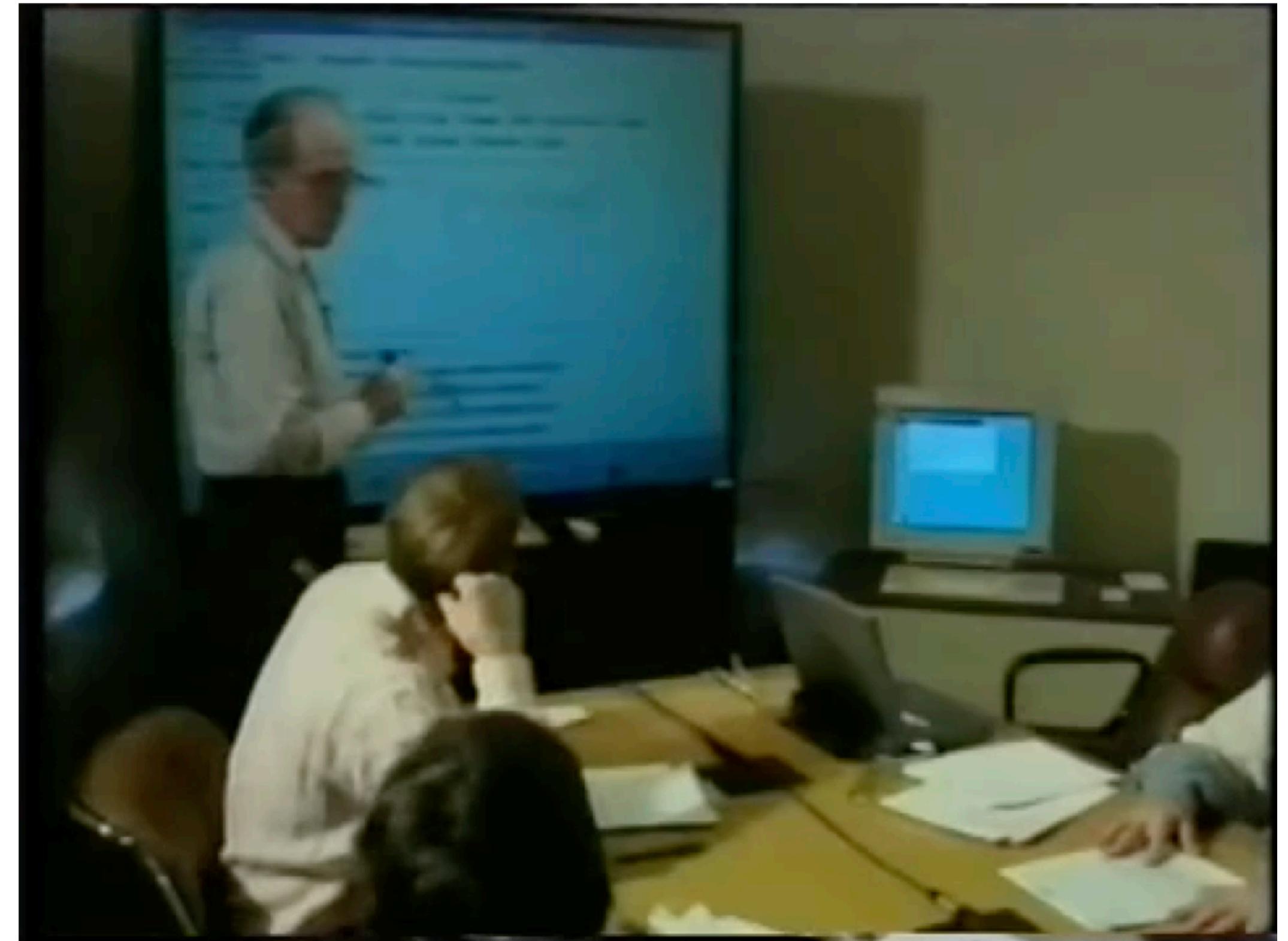
Project Tivoli



Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. INTERCHI, 1993.

Project Tivoli

- Pen-based gestures
 - Writing
 - Increasing size
 - Pick up and drag
- Inherently collaborative use
- Value in connecting information to other computers



Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings. INTERCHI, 1993.

Interactive Whiteboards (~2005)

- Widely introduced for educational purposes
- Mostly served as external monitors for computers
- Operated as a projector with a resistive touchscreen



Public Displays

- Often focused on immersion
- Close integration with the environment is important, from both a technical and a design perspective
- Very few libraries or toolkits to help with development



Large-scale displays

- Rarely for consumer use
- Often blend personal with interpersonal interaction



Question



Would touching a button be easier on a large display or a typical monitor, assuming the button is scaled?

- A Large display
- B Typical monitor
- C They would be about the same
- D
- E

Would touching a button be easier on a large display or a typical monitor, assuming the button is scaled?

- A Large display
- B Typical monitor
- C They would be about the same
- D
- E

A

0%

B

0%

C

0%

D

0%

E

0%

Question



Would touching a button be easier on a large display or a typical monitor, assuming the button is scaled?

- A Large display
- B Typical monitor
- C They would be about the same
- D
- E

Designing for large displays

Scale everything up

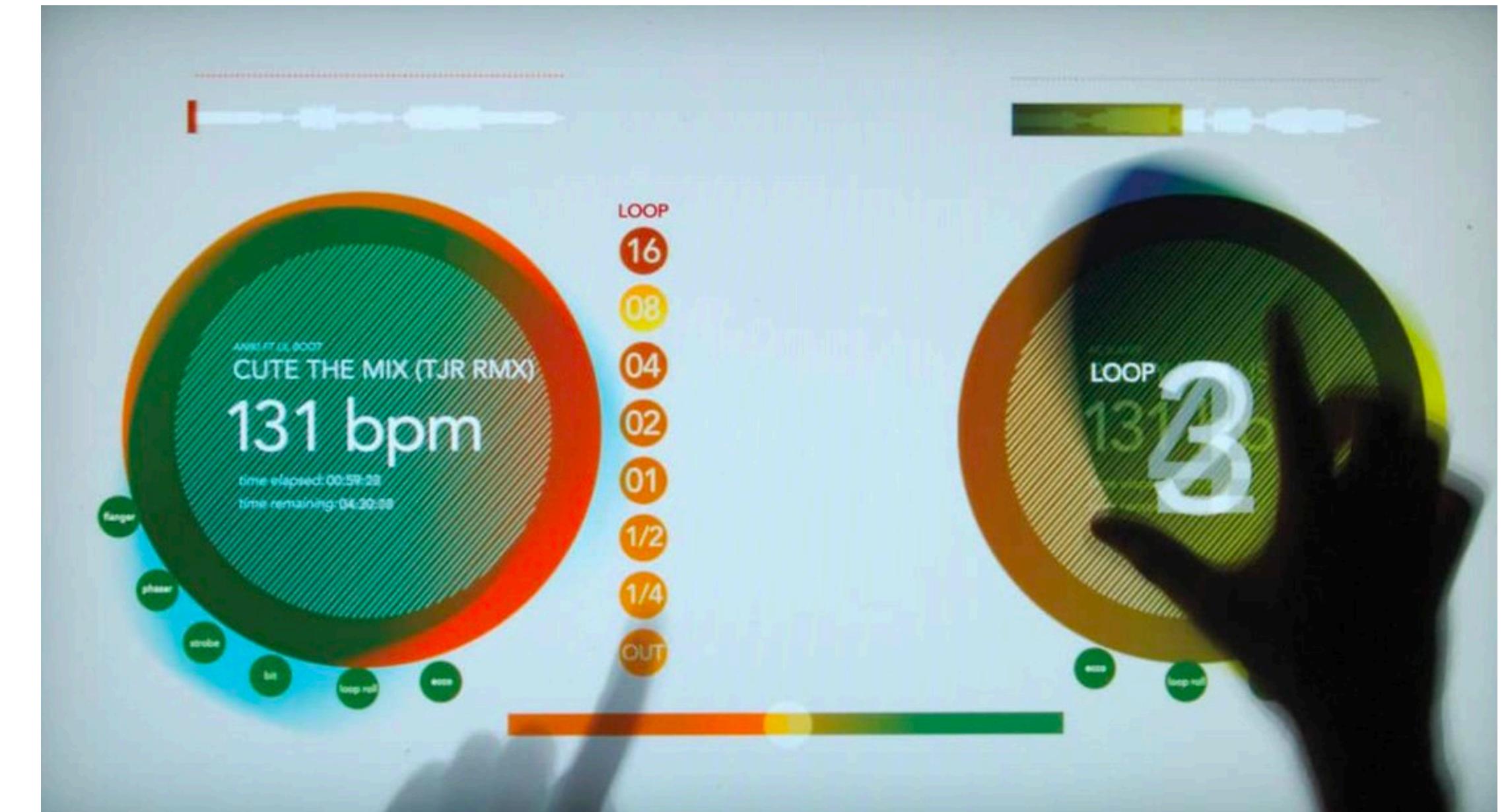
- A person is likely to be viewing the screen from further away
- Displays are often in public spaces, so large elements can encourage someone to come interact
- Often in new or unfamiliar environments, so interaction needs to be obvious

<https://designshack.net/articles/ux-design/tips-for-designing-oversized-touchscreens/>

<https://uxdesign.cc/designing-for-large-touch-screen-always-have-the-user-context-in-mind-878b6d2e02a9>

Enable more complex gestures

- Consider using a person's entire hand
- Movement will be exaggerated, avoid requiring high precision
- Minimize overall arm movement, avoid long swipes



<https://designshack.net/articles/ux-design/tips-for-designing-oversized-touchscreens/>

<https://uxdesign.cc/designing-for-large-touch-screen-always-have-the-user-context-in-mind-878b6d2e02a9>

Respect privacy

- The bigger the screen, the more likely someone else can see the interaction
- Large screens will often have smaller pop-ups for data entry
- Consider risks of requiring internet connection

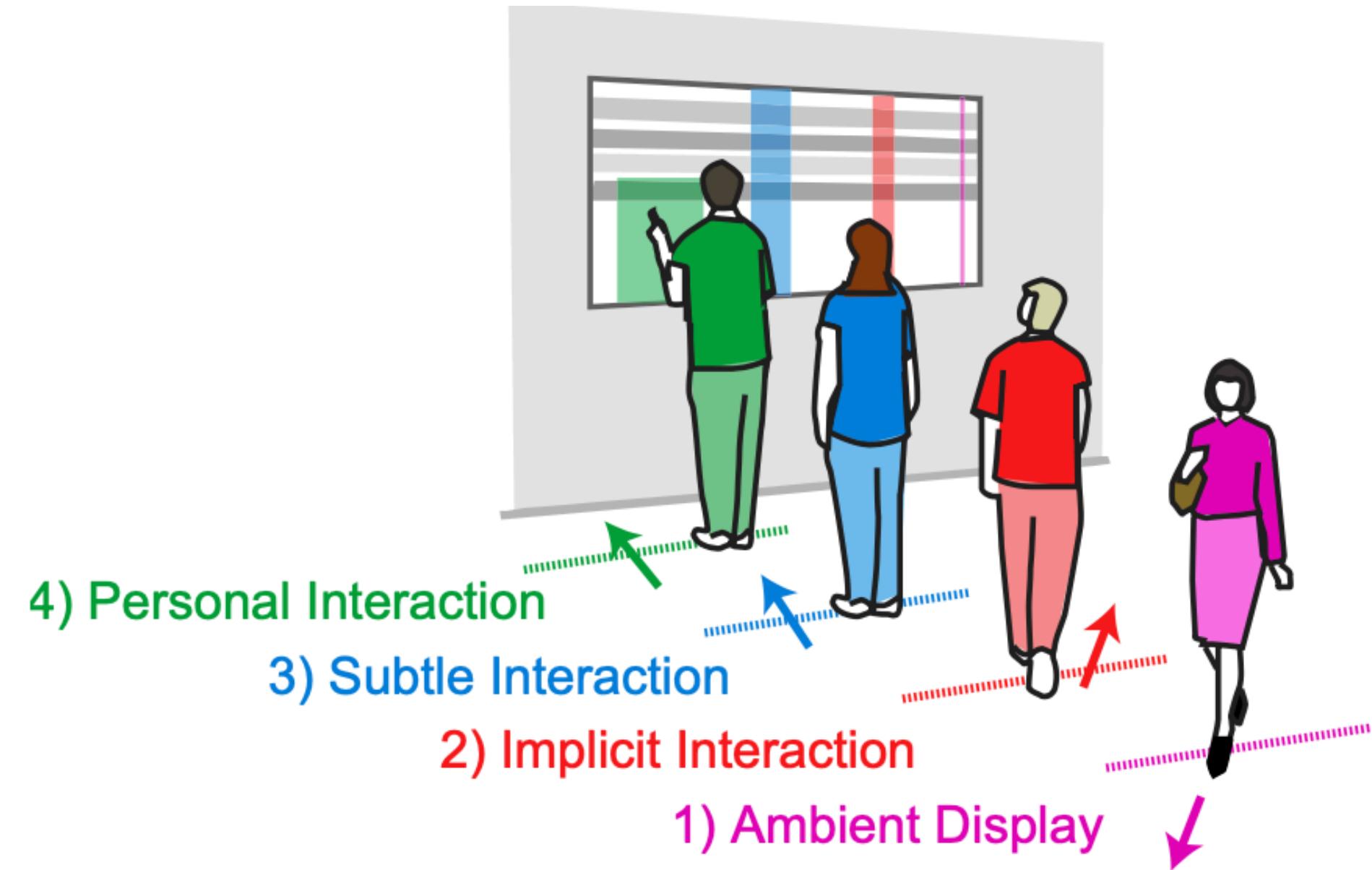


<https://designshack.net/articles/ux-design/tips-for-designing-oversized-touchscreens/>

<https://uxdesign.cc/designing-for-large-touch-screen-always-have-the-user-context-in-mind-878b6d2e02a9>

Respect interaction phases

- There are different “phases” of interaction in public spaces
- Layout needs at each phase are different
- Can draw people in with large elements, then start presenting more personal information

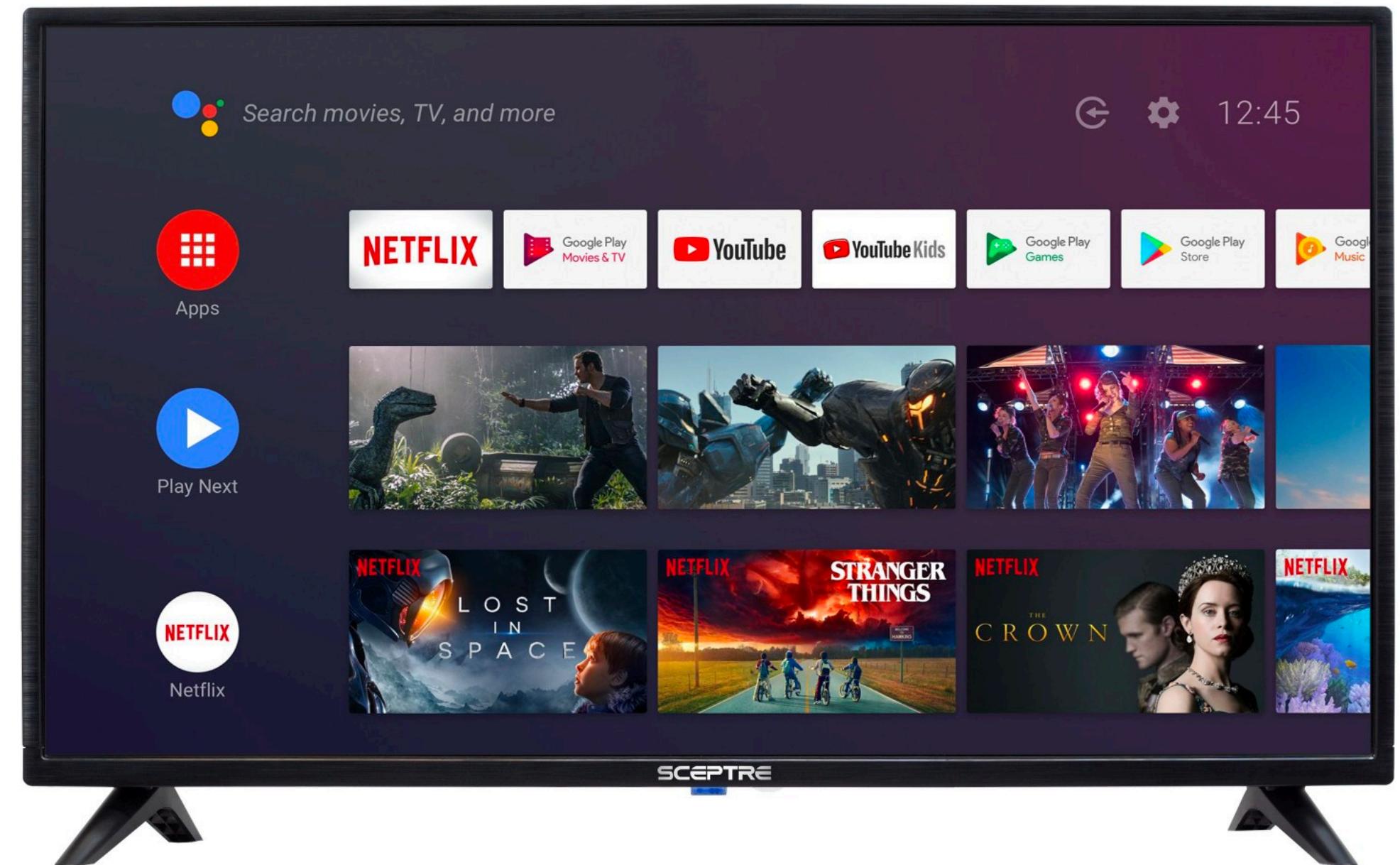
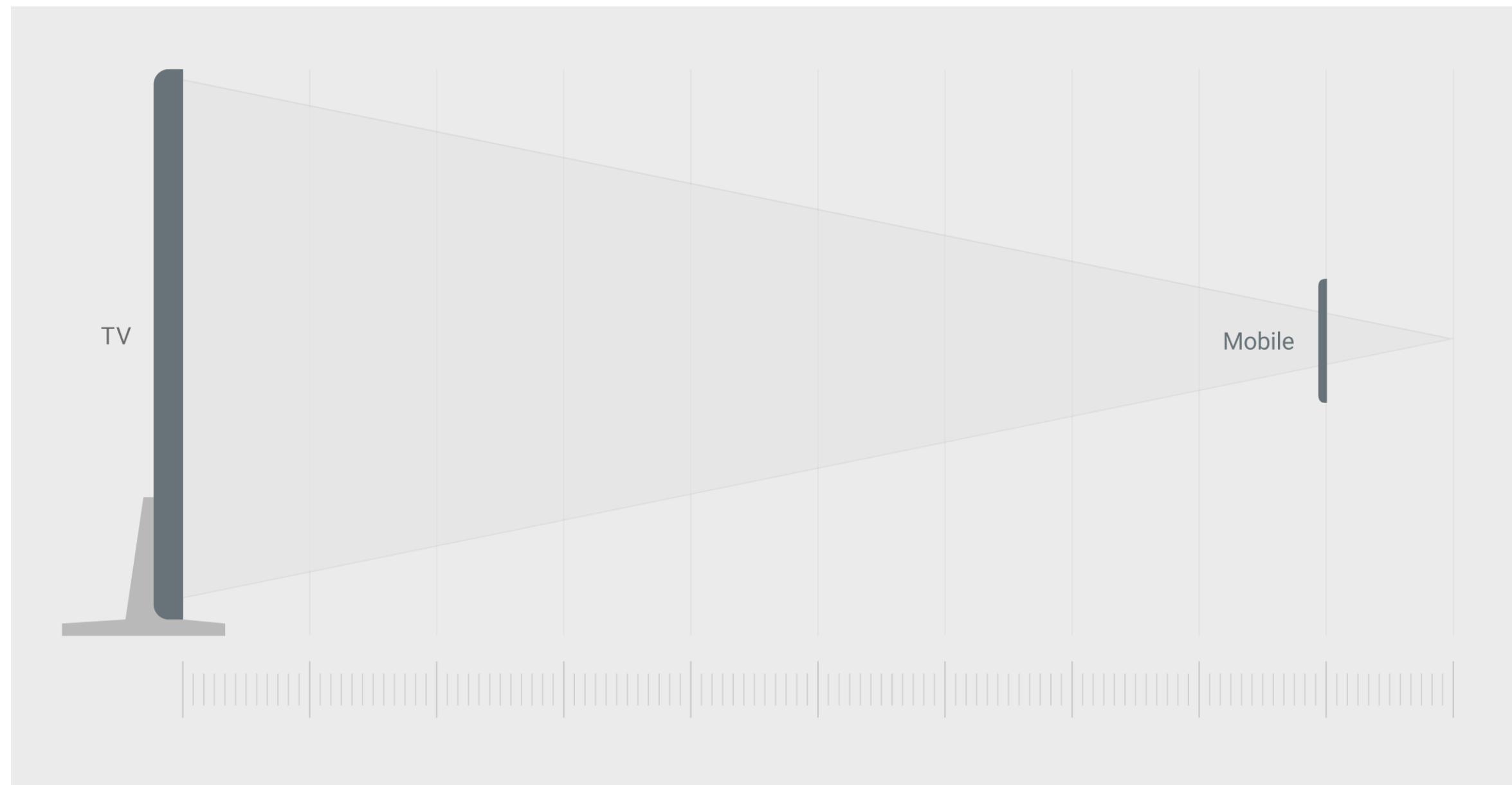


<https://designshack.net/articles/ux-design/tips-for-designing-oversized-touchscreens/>

<https://uxdesign.cc/designing-for-large-touch-screen-always-have-the-user-context-in-mind-878b6d2e02a9>

Smart TVs

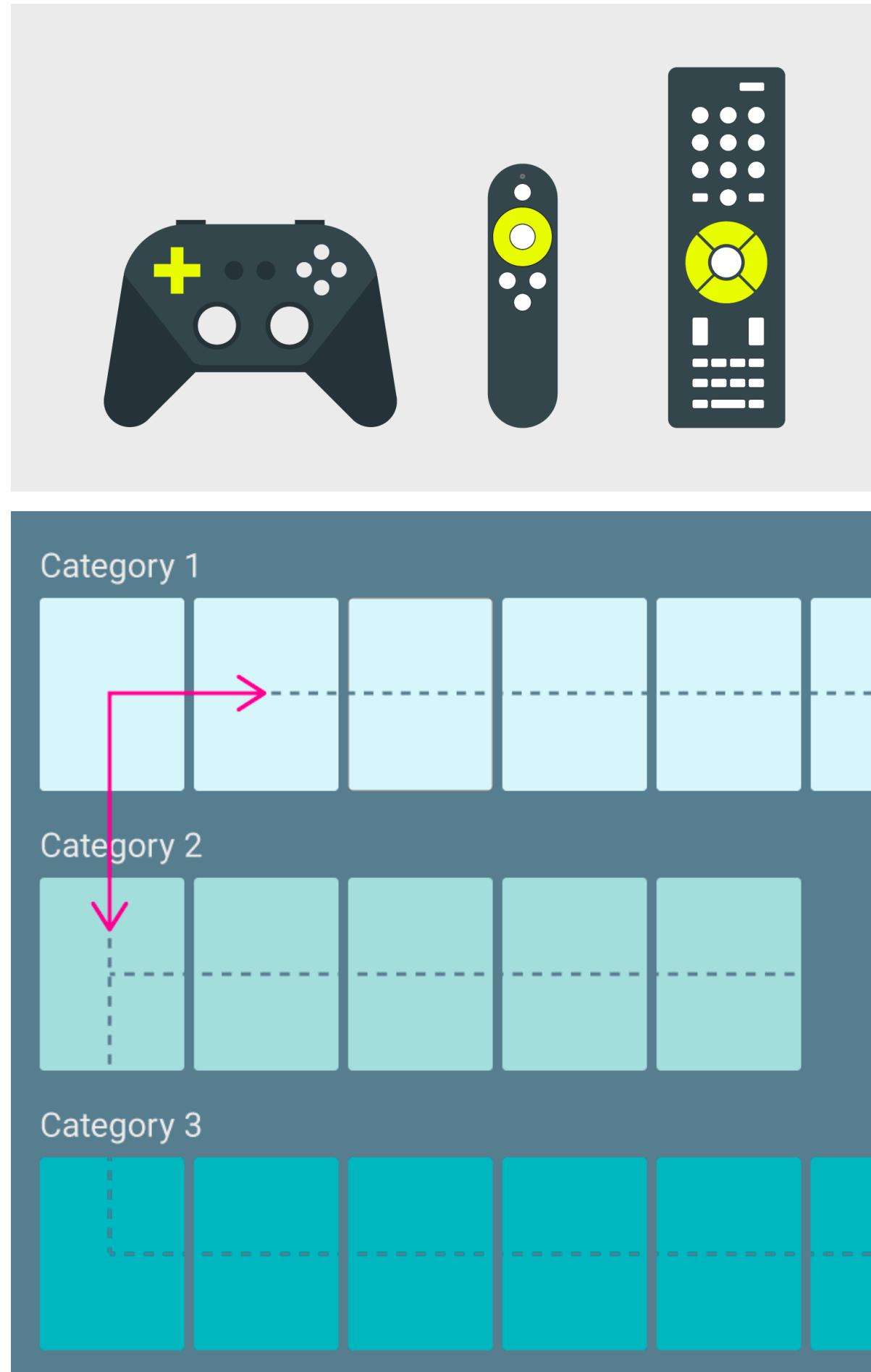
- The “smart” implies internet access
- Can safely assume that people are far away



<https://developer.android.com/design/ui/tv/guides/foundations/design-for-tv>

Smart TVs

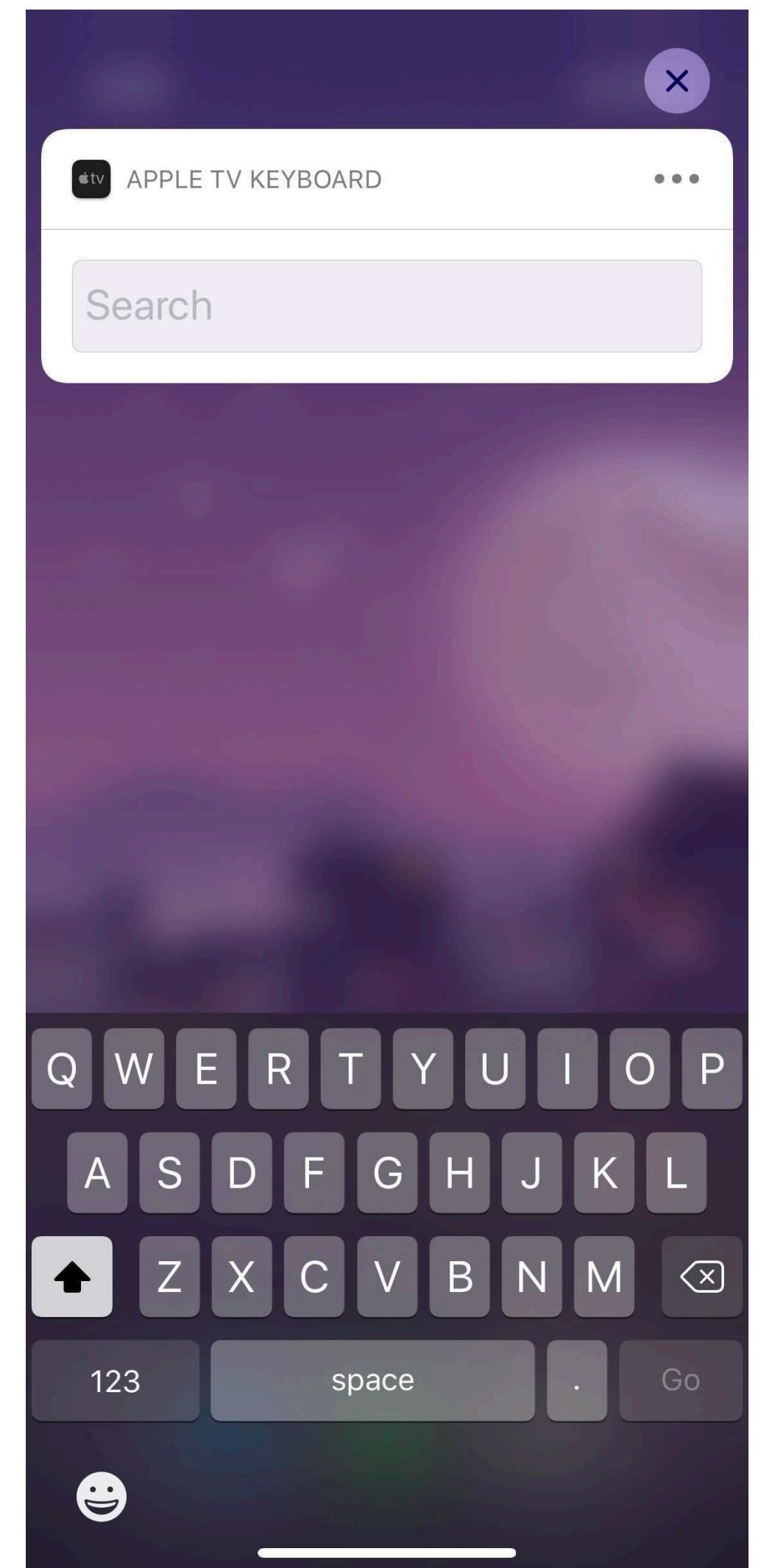
- Interaction is often via directional pads, so axis-based navigation is important



<https://developer.android.com/design/ui/tv/guides/foundations/design-for-tv>

Smart TVs

- Text input via D-pads is slow
 - Voice-based interaction can help
 - Can offload to mobile devices, if paired



<https://developer.android.com/design/ui/tv/guides/foundations/design-for-tv>

Developing for large displays

Developing for large displays

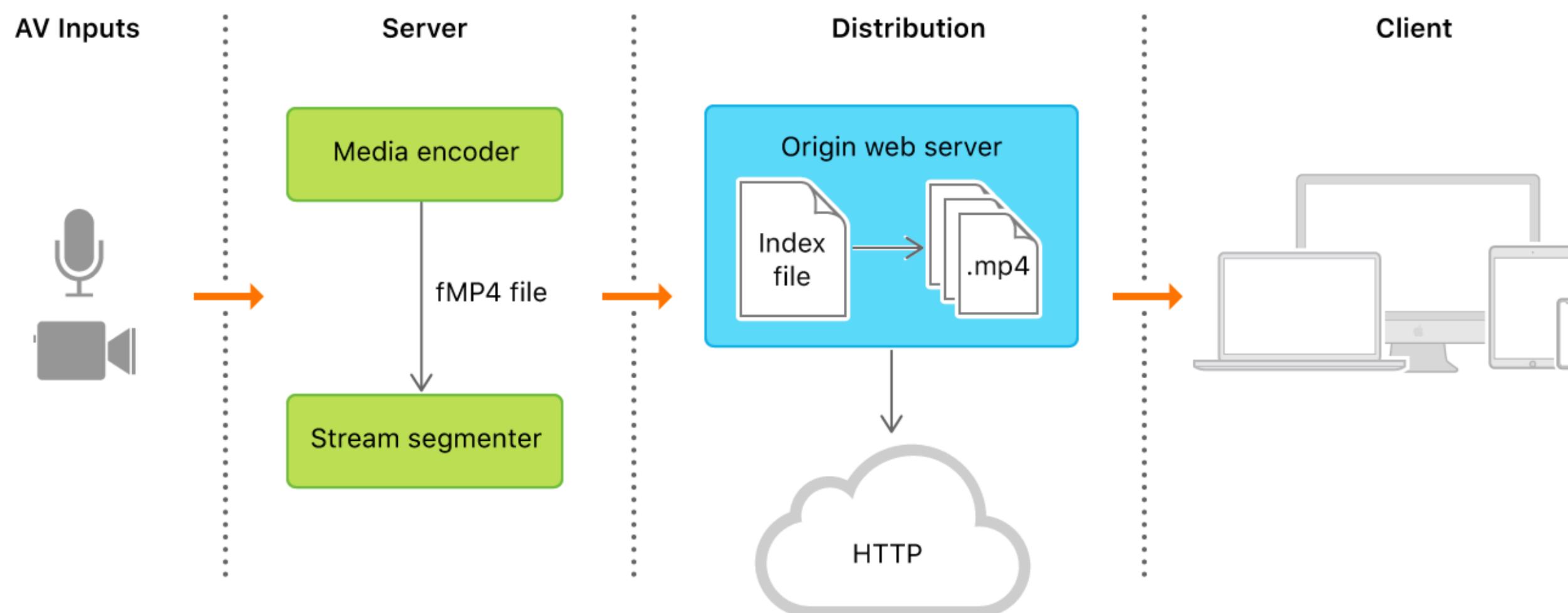
- Displays often have access to browsers, even offline
- Can use plain old HTML, CSS, and JavaScript
 - Bootstrap has an `xl` class for extra large displays
- More complex environments require custom software for gesture recognition, integrating other sensors

Developing for large displays

- Android TV
 - <https://developer.android.com/tv>
- tvOS (Apple)
 - <https://developer.apple.com/tvos/>
- Both provide native apps for interaction
 - Swift for tvOS, Java for Android TV

Developing for large displays

- Most media is still streaming, so falling back to HTTP and JavaScript is useful
 - Webservers host media content to play



https://developer.apple.com/documentation/http_live_streaming

Today's goals

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

- Describe some of the history of wearable computing and large scale displays
- Explain key principles for designing a good wearable experience
- Articulate some principles for creating experiences with large displays
- Identify a few potential libraries for developing wearable and TV apps

IN4MATX 133: User Interface Software

Lecture 17:
Small & Large Displays