

Accessibility and the Web

Input, Interaction, and Accessibility
Spring 2019

Monday

- Briefly:
 - Other laws of human perf
 - Beating Fitts' Law
- Web accessibility prep
- Intro to Project 1 (due in 3 weeks)

Updates for Wednesday



- Graduate Project info posted
- **Today:**
 - Brief intro to accessible design thinking
 - Making web pages accessible

Related laws of human performance

Choice reaction time

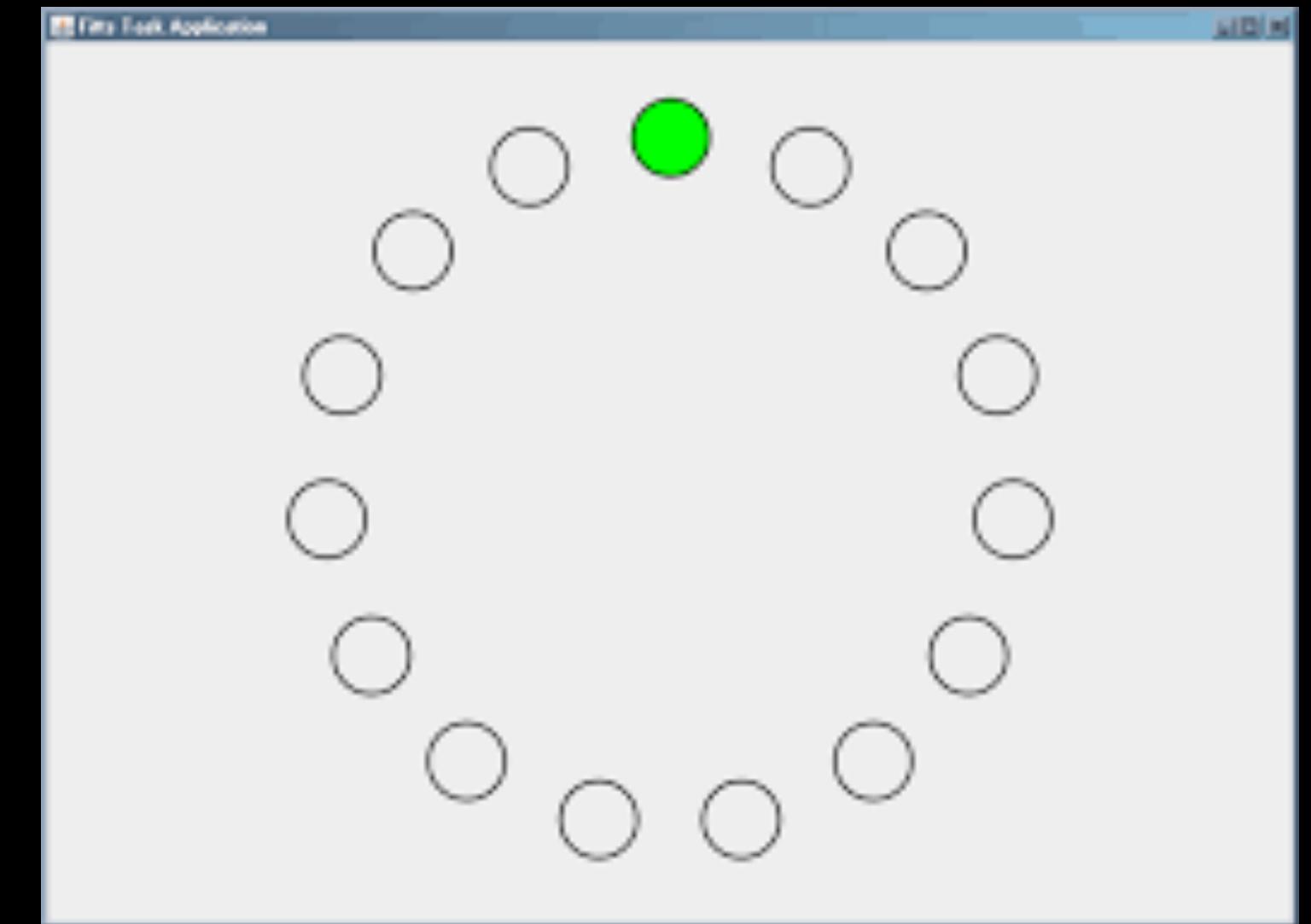
- Fitts' Law assumes that the target location is known
- What if the user needs to find the target first?

Hick-Hyman Law (1952-3)

- Given n equally probable choices, time to choose among them is approx.
 - $RT = a + b \log_2 (n+1)$
- $1/b$ is often 5-7 bits for adults
- This assumes all options are equally probable (but variation exists for unequal probabilities)

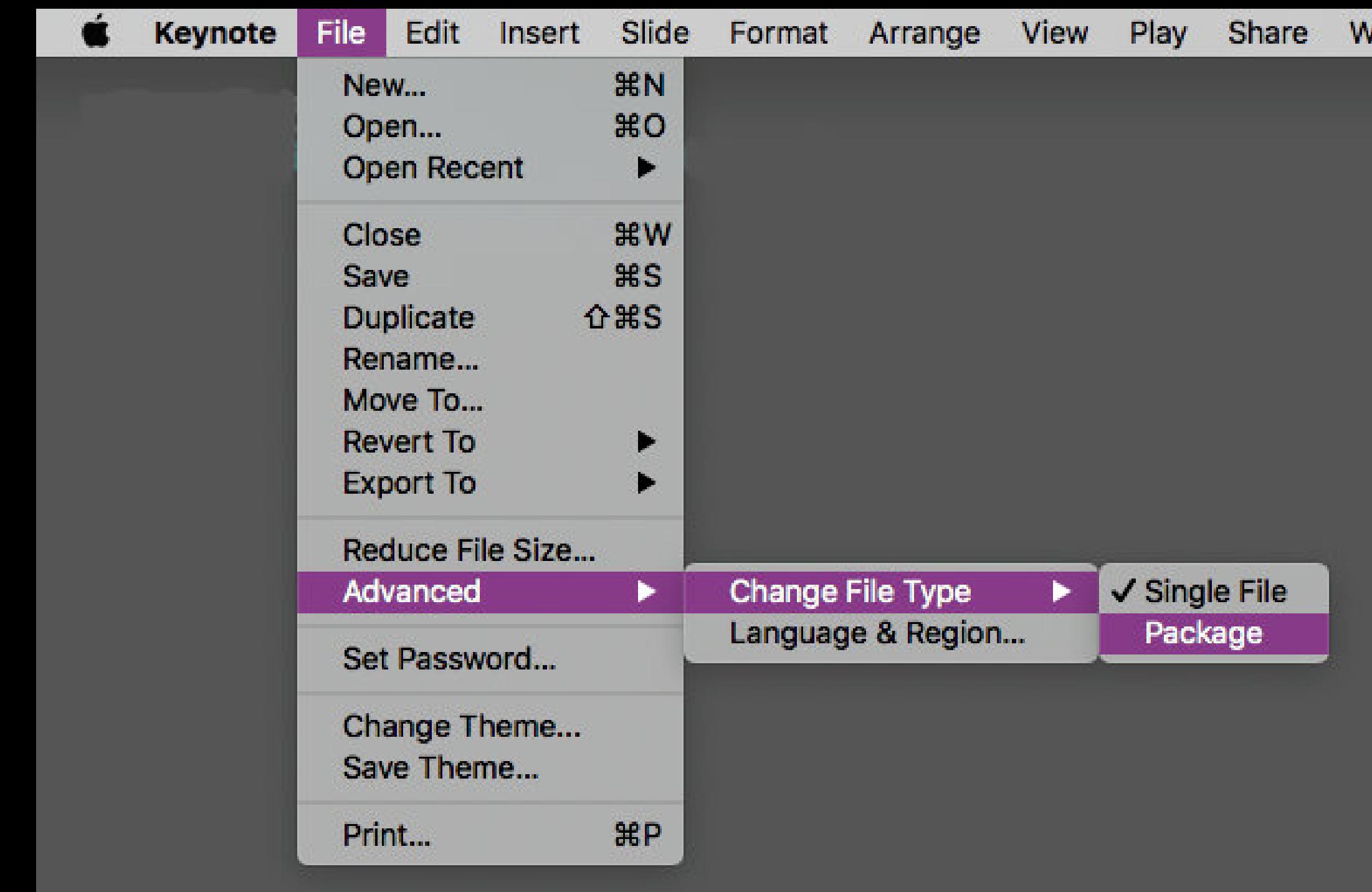
Using Hick-Hyman

- N buttons; one becomes highlighted
- Can combine with Fitts' to calculate *time to locate and then click*
- Used by Soukoreff and MacKenzie to calculate theoretical upper bound for stylus and soft keyboard typing



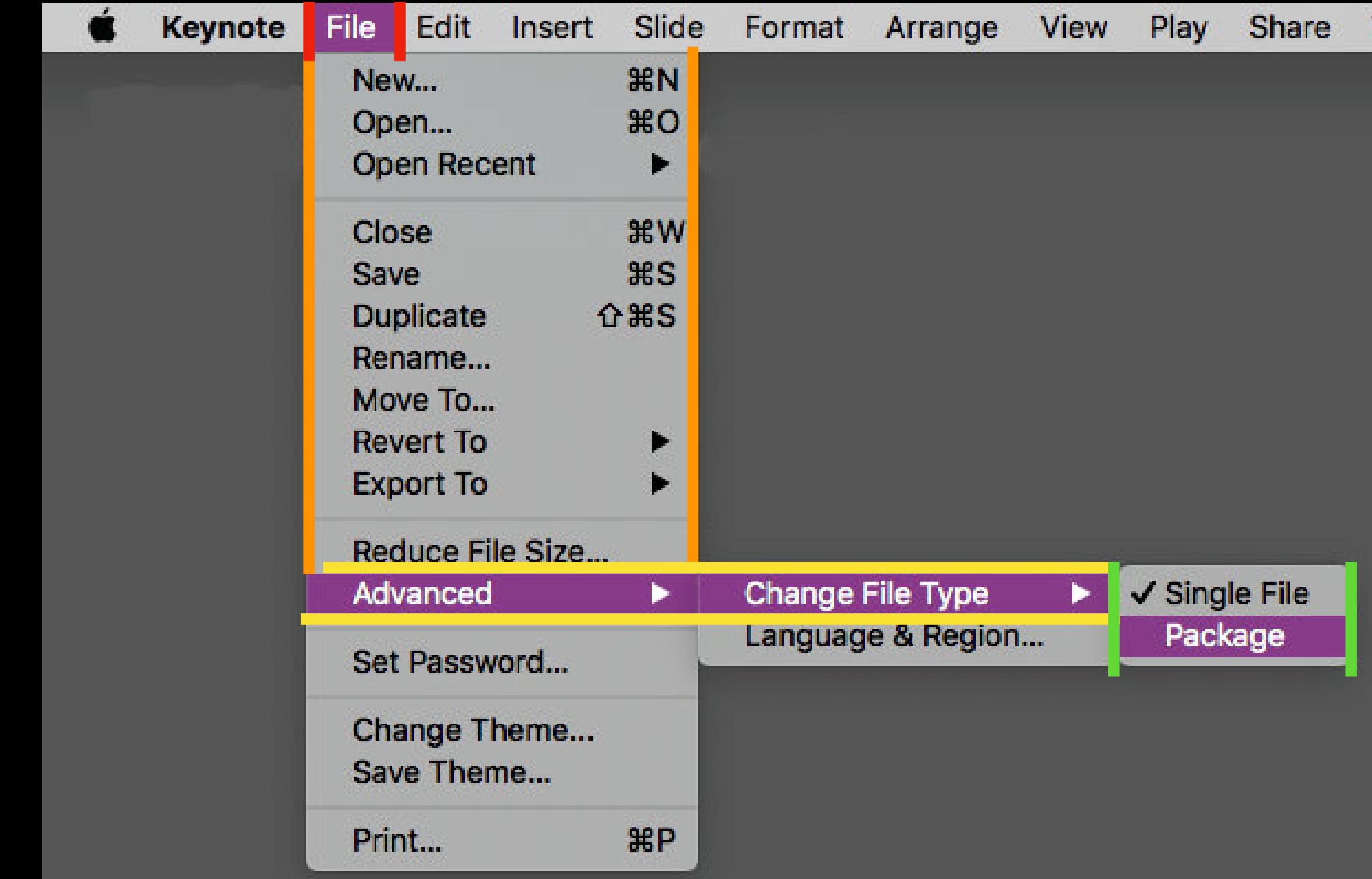
Steering law

- Fitts' Law only models a single target (essentially a 2D line)
- What about more complicated pointing tasks?
- These are *steering tasks*



Steering law

- User is navigating a 2D tunnel
- We essentially integrate along the curve of movement



Steering law (Accot and Zhai 1997)

$$T = a + b \int_C \frac{ds}{W(s)}$$

- T - time to navigate the path
- C - path
- W(s) - width of path at s
- a and b - index of performance (as in Fitts' Law)

Error Law (Wobbrock et al. 2008)

- Fix movement time (in studies, use a metronome)
- Derived the error law:
- For more details,
see [the paper](#)

$$P(E) = 1 - \text{erf} \left(\frac{2.066 \frac{W}{A} \left(2^{\frac{MT_e - a}{b}} - 1 \right)}{\sqrt{2}} \right)$$

Other laws of performance

- Power law of practice
- Guillard's model for bimanual tasks
- Schmidt's law for open-loop movement

Beating Fitts' Law

Can we beat Fitts?

- What would it mean to beat Fitts Law?



Can we beat Fitts's Law?

- How does Fitts's Law teach us how to improve performance?
 - Reduce (effective) distance to target
 - Increase (effective) size of target

Improving performance

- Pointing performance will be improved when targets are **nearby** and **large**
- So, design UIs such that important functions are nearby and large

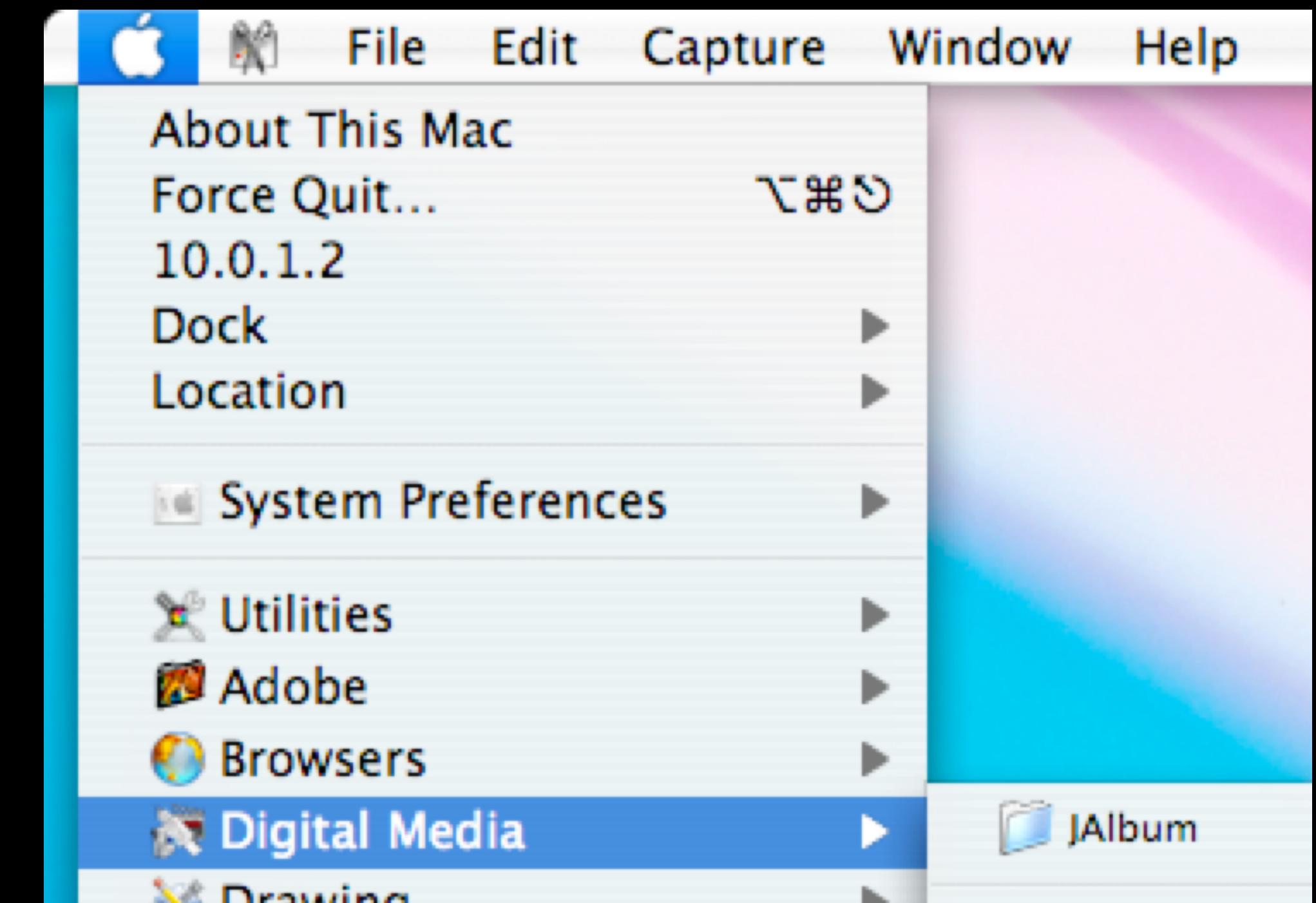
Radial menus

- Place menus around the cursor
- Often used in games and artistic software
- Limitations?



Edges and corners

- Edges and corners typically have infinite size
- No penalty for overshooting the target

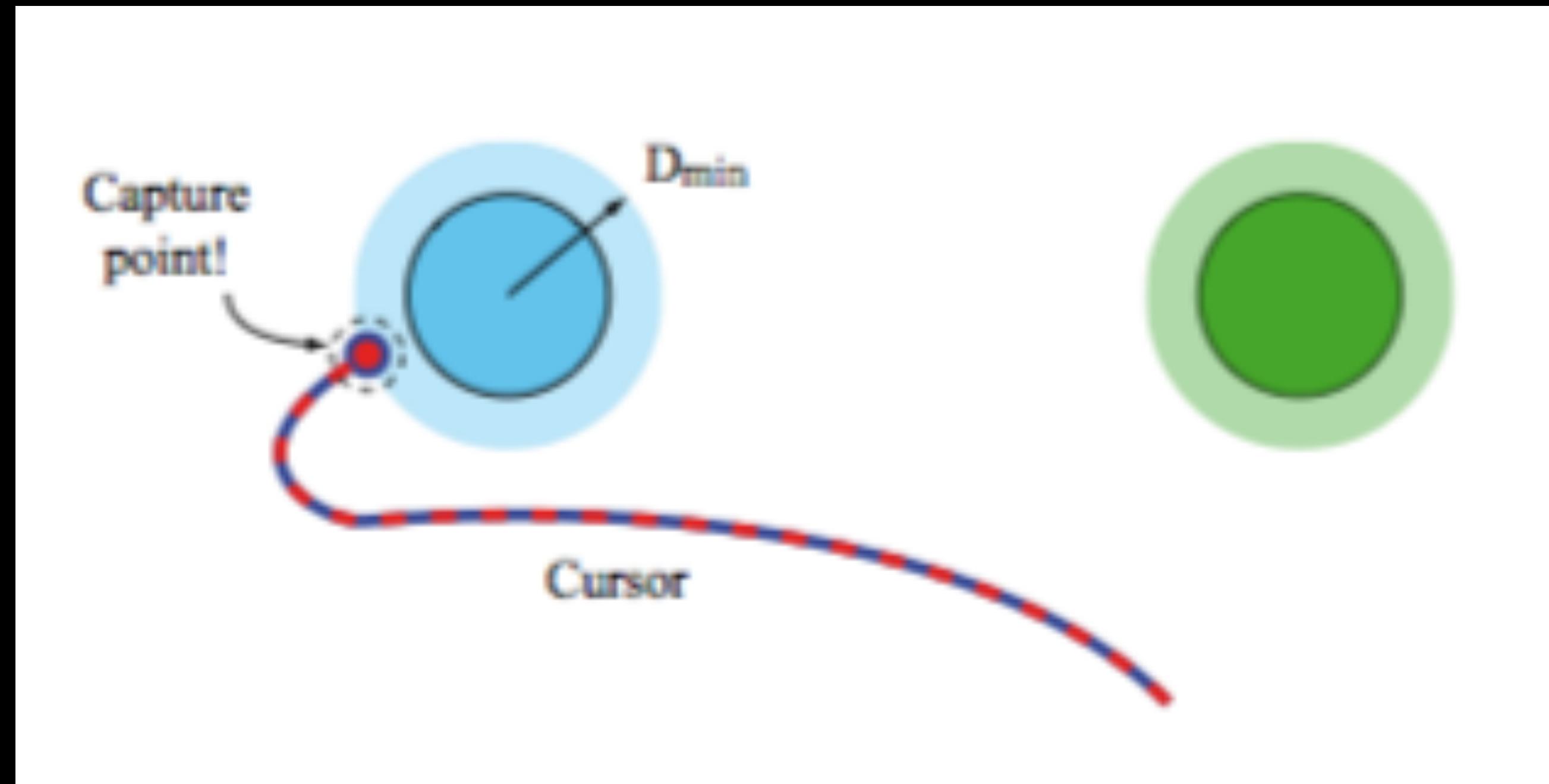


Pointing facilitation

- Two fundamental approaches
- **Target aware** - the system knows where targets are on the screen
- **Target agnostic** - the system does not know where targets are on the screen

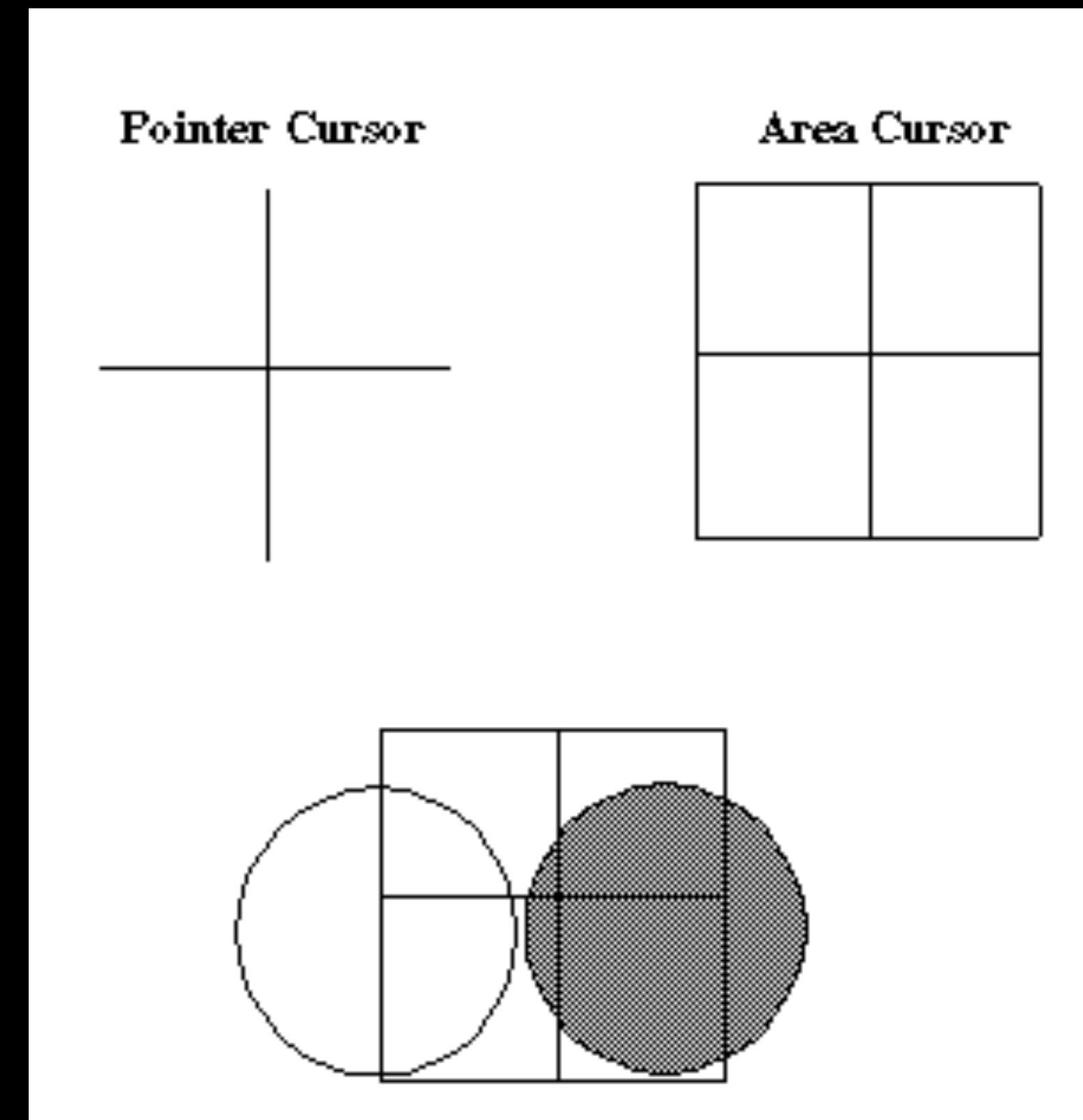
Target aware: Facilitating movement

- Jump between icons or make “sticky” icons
- “Gravity fields”



Area Cursors

- One way to improve pointing is to increase the size of the cursor: easier to hit targets with a bigger cursor
- Downside: this method can break down when targets are close to each other



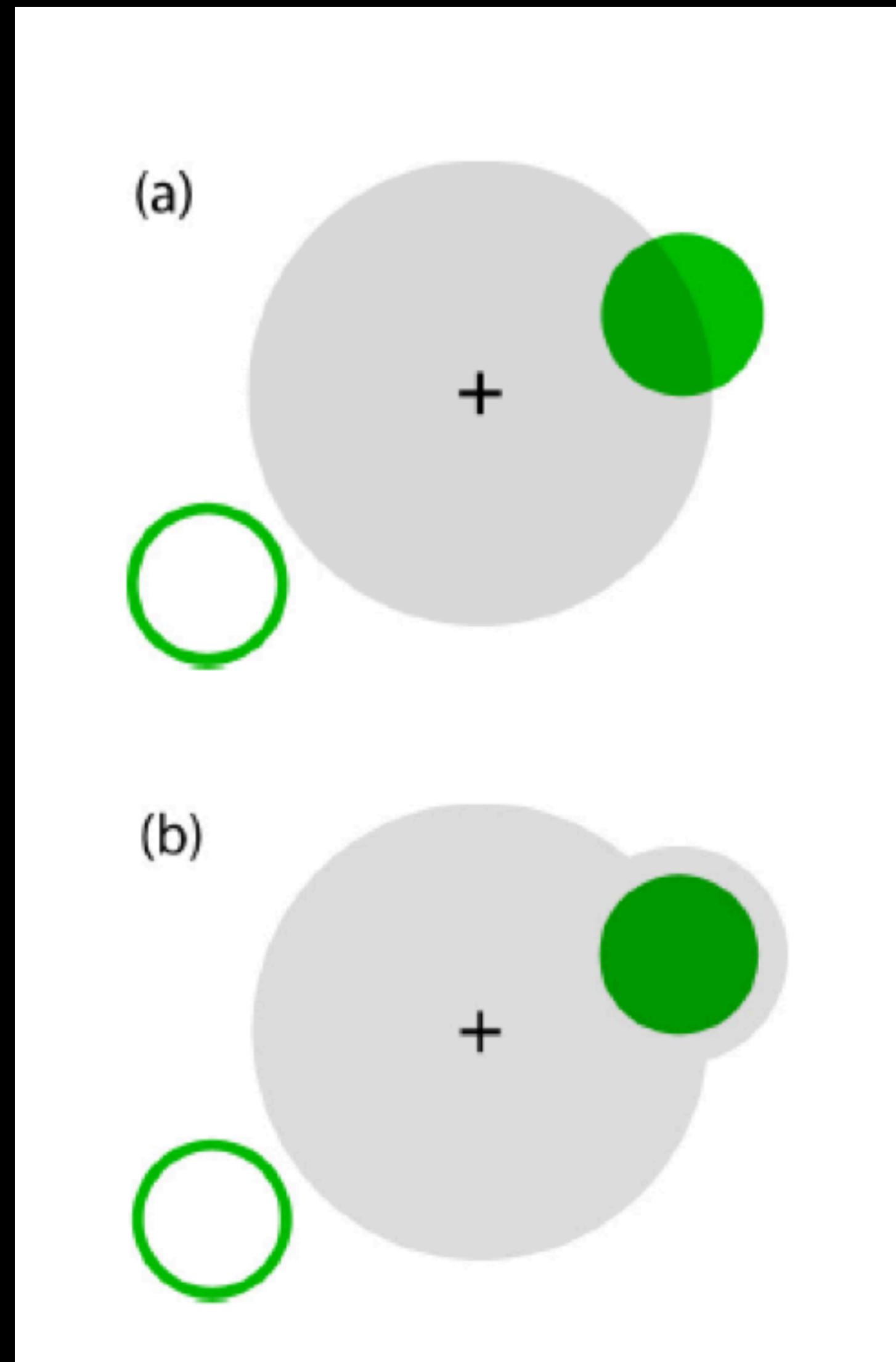
Limitation of area cursor

- What to do with densely packed targets?

Bubble Cursor

(Grossman and Balakrishnan 2005)

- Dynamically resize area cursor to contain the closest target, but not the second closest
- Considered by many to be the best pointer facilitation method



Beating the bubble cursor

The Bubble Lens

Martez E. Mott, University of Washington [[contact](#)]
Jacob O. Wobbrock, University of Washington

Download

Current Version 1.0.0

Windows executable: [BubbleLensExe.zip](#)

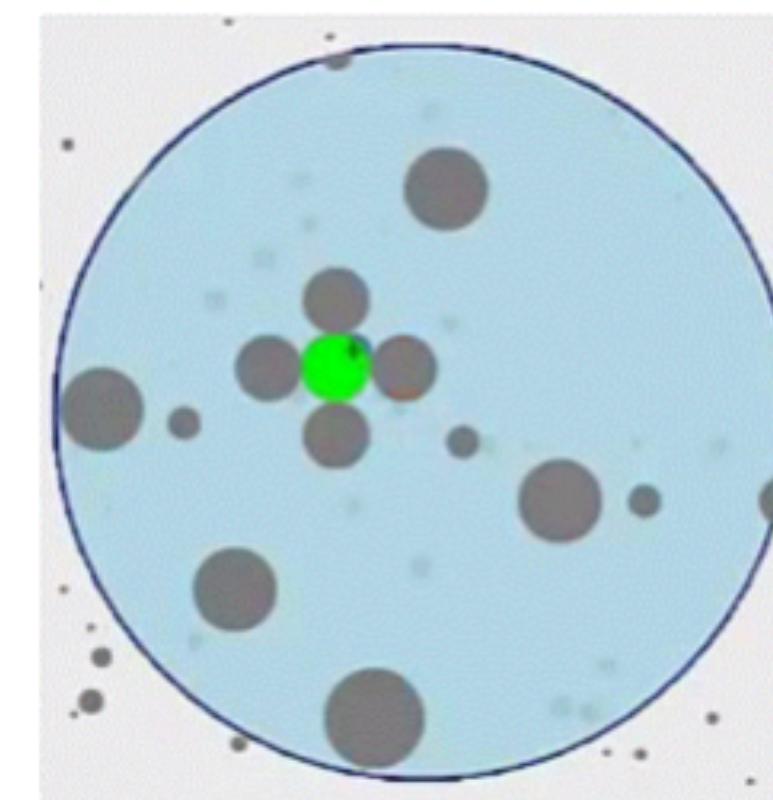
Source code: [BubbleLens.zip](#)

Pseudocode: [Pseudocode \(PDF\)](#)

The Bubble Lens requires the Microsoft .NET 2.0 Framework or later. [Download it here.](#)
This software is distributed under the [New BSD License](#) agreement.

About

The Bubble Lens is a pointing facilitation technique that builds upon the Bubble Cursor by remedying the Bubble Cursor's poor performance around small, dense targets. Specifically, the Bubble Lens detects when the user is approaching a small, densely packed target and, as the user attempts to acquire this target, the cursor automatically magnifies into a circular lens, speeding the user's final stage of target acquisition. The automatic magnification is based on a technique called *kinematic triggering* that continually monitors a movement's velocity profile for the presence of certain events or conditions. In the case of the Bubble Lens, kinematic triggering fires the lens on the downslope of the first corrective movement in the velocity profile. In laboratory studies of random target-field pointing, the Bubble Lens was 10.2% faster and 37.9% more accurate than the Bubble Cursor.



Why target agnostic?

- Sometimes the programmer does not have the ability to programmatically identify UI layout (e.g. old UI toolkits)
- Sometimes the number of targets is too large

Where are the targets?

- Buttons, tabs, etc.
 - In between every character

The screenshot shows a Microsoft Word document titled "kane-cv-july2018 — Saved to my Mac". The ribbon menu is visible at the top, with "Home" selected. The toolbar includes standard options like Paste, Insert, Draw, Design, Layout, References, Mailings, Review, View, and Zotero. The font and style ribbon shows "Helvetica N..." and "11.5". The main content area contains the following information:

Updated November 2018

Shaun K. Kane

+ Department of Computer Science
University of Colorado Boulder
430 UCB, Boulder CO 80309

phone: 720-446-9995
email: shaun.kane@colorado.edu
web: shaunkane.com

Academic Employment

- Associate Professor, Department of Computer Science (2018–)
University of Colorado Boulder, Boulder CO
- Assistant Professor, Department of Computer Science (2014–2018)
University of Colorado Boulder, Boulder CO
- Assistant Professor, Department of Information Systems (2011–2014)
University of Maryland Baltimore County, Baltimore MD

Education

- Ph.D. in Information Science, University of Washington, Seattle WA (2005–2011)
Chairs: Jacob O. Wobbrock and Richard E. Ladner
- M.S. in Information Science, University of Washington, Seattle WA (2011)
Advisor: Jacob O. Wobbrock
- M.S. in Computer Science, University of Massachusetts, Amherst MA (2003–2005)
Advisor: Robert Moll
- B.S. in Computer Science, University of Massachusetts, Amherst MA (1999–2003)
Honors Advisor: Beverly Park Woolf

Publications

Journal Articles

[J.7] Profita, H.P., Stangl, A., Matuszewska, L., Sky, S., Kushalnagar, R., and Kane, S.K. (2018). "Wear it loud": How and why hearing aid and cochlear implant users customize their devices

Target-agnostic methods

- Angle Mouse (Wobbrock et al.)
 - Detect when corrective movements start (based on speed and angular deviation)
 - Slow the cursor during correction phase

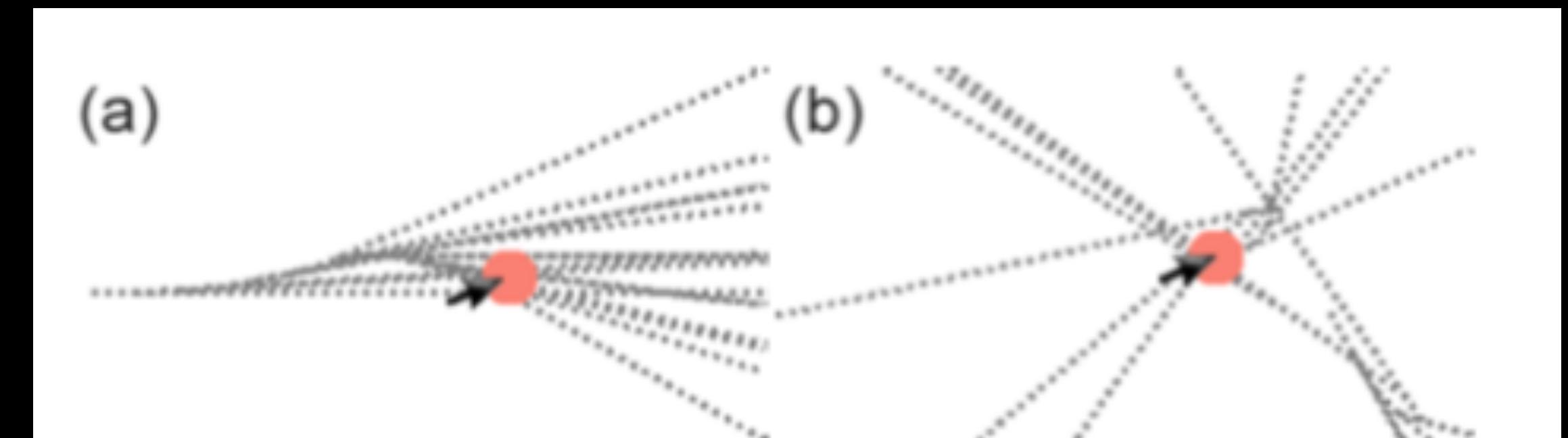


Figure 2. (a) Coherent movement produces low angular deviation (σ_θ), 13.7° in this example, and retains most of the maximum gain (89%). (b) Divergent movement produces high angular deviation (94.8°) and keeps less of the maximum gain (21%).

More target-agnostic methods

- Steady Clicks (Trewin et al.)
 - For pen-based interfaces
 - Some users, especially older users, accidentally “slip” the cursor while clicking
 - Detect this movement (based on characteristic shape) and find the target from a few seconds before

Implementing target aware techniques

- Incredibly difficult unless you control the OS
- Some research has used computer vision to reverse engineer the UI (see Morgan Dixon's [Prefab](#) system)
- Depends on there being an appropriate number of targets

User-specific models

- Some individuals may have additional, atypical challenges related to pointing
- Examples
 - Parkinson's Disease may cause tremor when using a pointing device
 - Muscular Dystrophy may impair large movements, but doesn't affect precision
- Can we build user-specific models of pointing?

Supple++ (Gajos et al.)

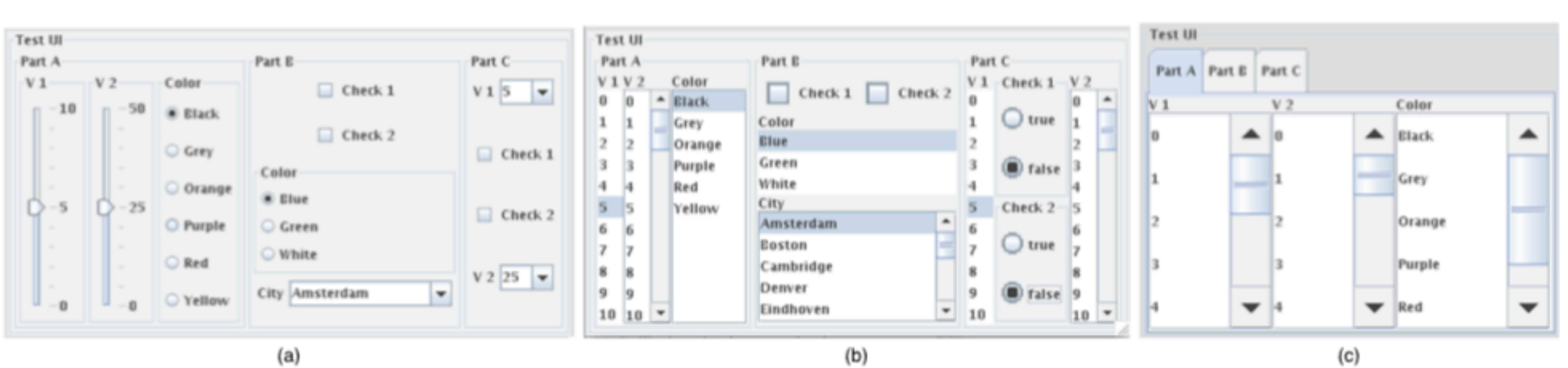


Figure 5: Three renderings of a synthetic interface used in the preliminary study and automatically generated under the same size constraint: (a) base line preference-optimized GUI; (b) personalized for the mouse user with muscular dystrophy (M03); (c) personalized for the eye tracker user (ET01). GUIs (b) and (c) were generated by SUPPLE++.

Roots of SUPPLE

- Originally developed to adapt remote controls to different devices

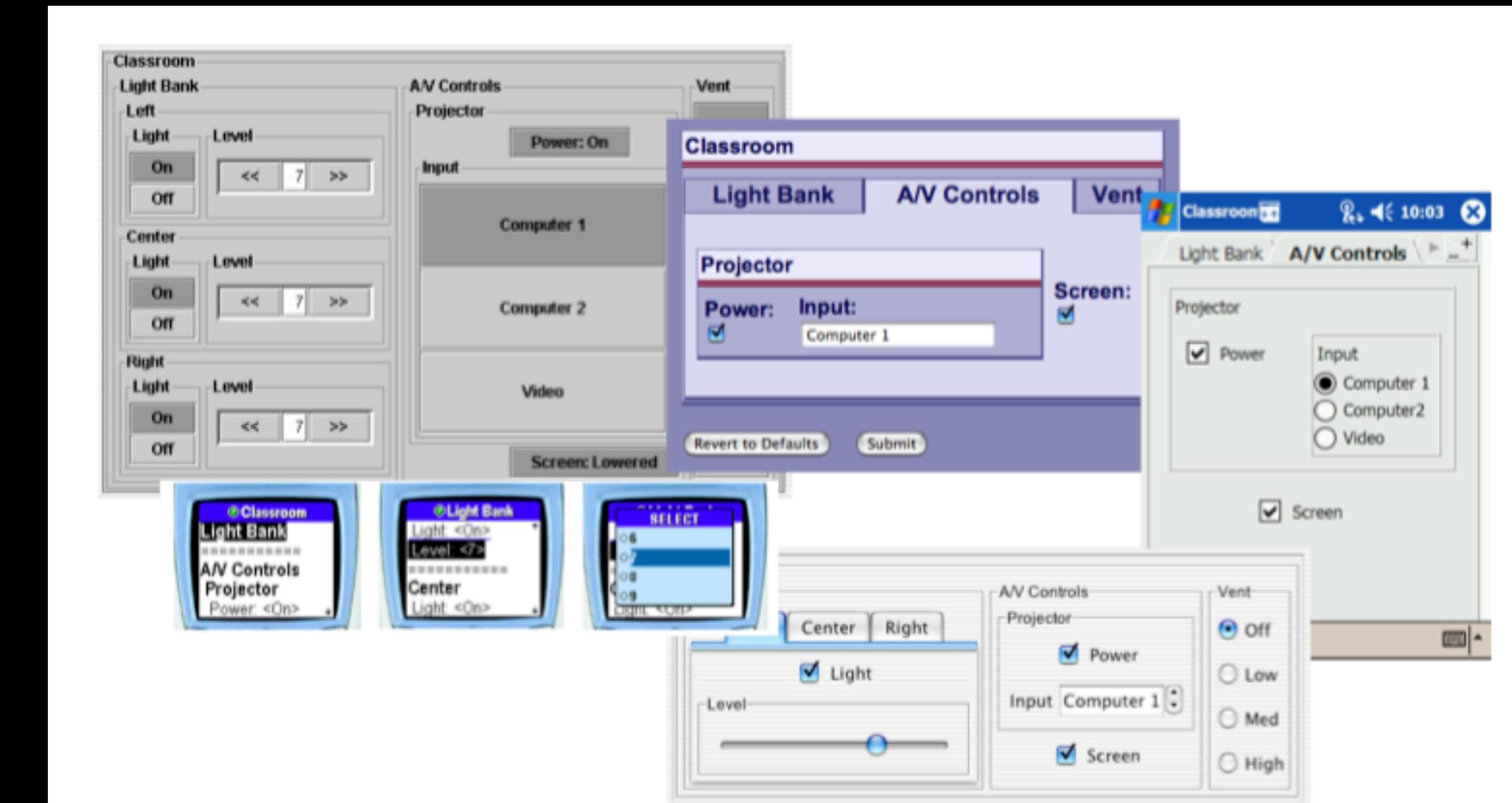


Figure 1: An interface for a simple application rendered automatically by SUPPLE for a touch panel, an HTML browser, a PDA, a desktop computer and a WAP phone.

Heuristic vs. data-driven

- Most of these approaches rely on some heuristic rules, identified and specified by a human programmer
- More modern techniques are often based on modeling a user (e.g. SUPPLE++)
- However, we still need to have an understanding of the problem space to understand possible solutions

Mini-design activity

- Consider our phone dialer user interface
- Sketch out a variant for
 - User with tremor
 - User with hand fatigue
 - User with low vision (but fine motor ability)

An inclusive design perspective

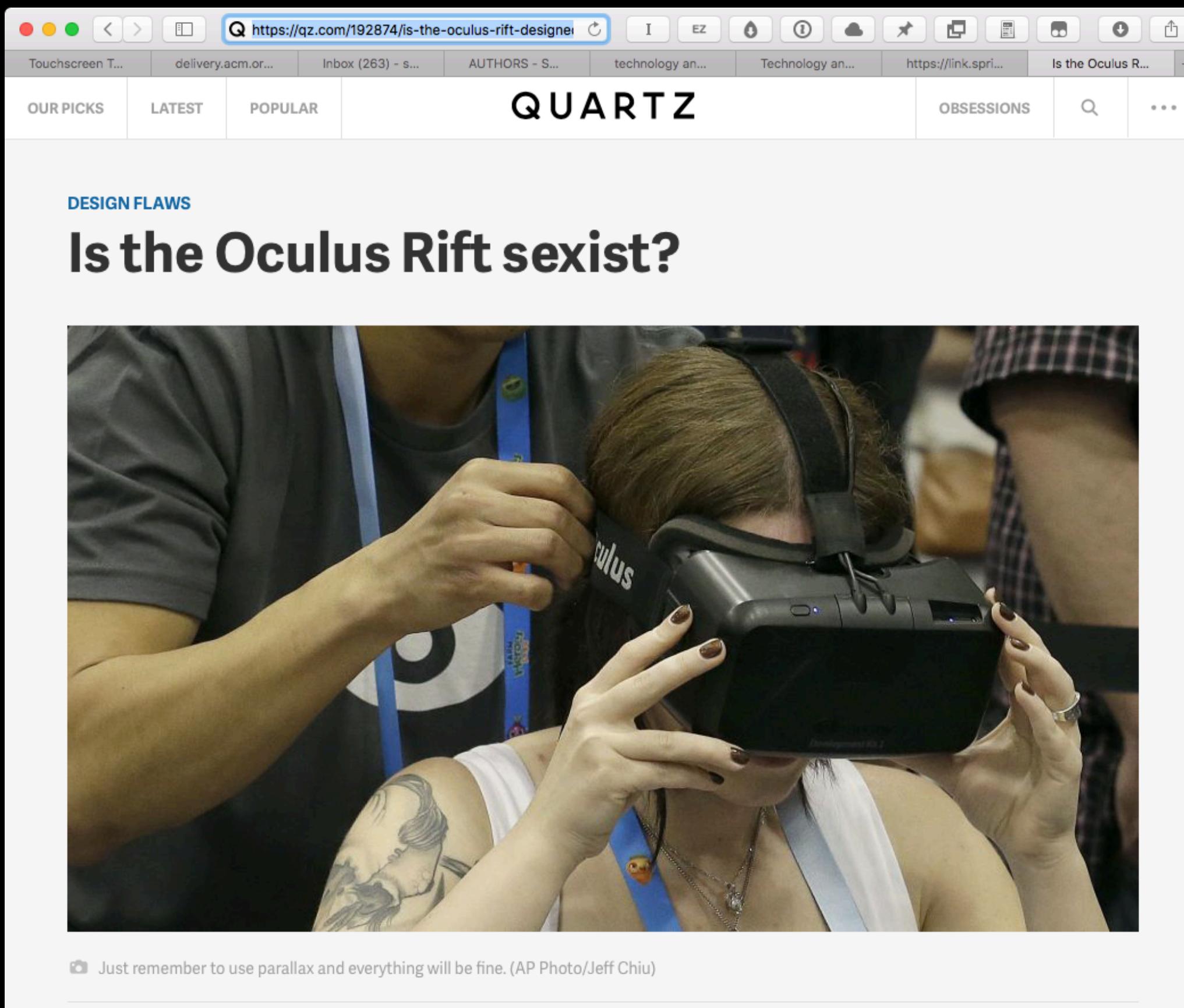
Considering diverse users

- How do we think about the individual?
- How do we think about the design process?

Disability & inclusion

- We'll spend much of our time talking about access for people with disabilities. Why?
 - There are many well-known but remaining accessibility problems
 - Addressing these problems can significantly impact people's lives

Inclusion beyond disability



Quartz website screenshot showing a woman wearing an Oculus Rift headset. The article title is "Is the Oculus Rift sexist?". A caption at the bottom left reads: "Just remember to use parallax and everything will be fine. (AP Photo/Jeff Chiu)".



PCWorld website screenshot showing a group of children holding Kinect accessories. The article title is "Is Microsoft's Kinect Racist?". A caption at the bottom left reads: "Will all these kids be able to use the facial recognition features of Microsoft's Xbox 360 Kinect accessory? Photo courtesy of Microsoft".

Accessibility can affect lots of groups

- Future technologies, but also issues of affordability, resources, etc.
- We should keep this in mind throughout what we do here
- For now, we'll talk more about disability

Some terminology re: disability

- PWD - people with disabilities
- Accessibility - the practice of designing artifacts that are usable (usually focusing on PWD)
- a11y - “cool” abbreviation for accessibility; [a+11 letters+y]. Inspired by internationalization
→ i18n

More on PWD terminology

- Class follows the **reasonable person principle**: assume everyone is reasonable, acting in good faith
- Disability isn't a bad word
- Some terms have gone out of fashion (e.g., wheelchair bound, mental retardation)
- Some terms raise questions: what does it mean to call something **normal**?

What, even, is disability?

- Who is included? Excluded?
- How to think and talk about it?

Models of disability

- **Medical model:** people with disabilities (PWD) are patients; our goal is to restore or replace function
- **Social model:** What counts as “disability” is defined socially; people are naturally different, but society sets the bounds for what is considered normal and what is abnormal. Maybe we don't need to fix people.

The social model in action

- [The Shared Sign Language of Martha's Vineyard](#)
- In the 18th century, deafness was common on the island of Martha's Vineyard
- Sign language was commonly used by everyone



By the early 18th century, it was not uncommon for people in Martha's Vineyard to be deaf from birth. This had a profound effect on the culture of Martha's Vineyard - and one that went on to influence Deaf culture in the United States as a whole.

Definitions change over time

- Being nearsighted was not a significant disability until literacy became common
- Changing norms around texting may benefit some (e.g., deaf people) and impair others (e.g., people with dyslexia)

WHO model of disability (2012)

- **Body functions** and **body structures** describes how a person's body functions
- **Impairments** refer to functional limitations of a person's body
- **Activity limitations** refer to difficulties performing specific activities
- **Participation restrictions** refer to difficulties participating in activities of daily life
- (and all of this is affected by environmental factors)

Is impairment situational?



Situational vs. lifelong impairments

- Must consider
 - Compensatory skills
 - Assistive technologies
 - Ability to change the impairing context
(e.g. by changing activity)

Perspectives and solutions

“What should we do about it?”

- **Rehabilitation technology** - enable people to restore natural function (usually temporary)
- **Assistive technology** - provide technology to solve an accessibility problem (usually permanent)
- **Special education** - focus on access to education and empowerment
- **Legal perspective** - focus on rights and societal supports
 - Whose responsibility is this problem?

Design approaches

- **Assistive technology** - developing technology to “repair” a disability or address an activity limitation
- **Universal design** - build it from the ground up to accommodate the widest user group
- **Inclusive design** - often more pragmatic than universal design (include more, but not everyone), emphasizes participation by excluded users in the design process

Universal design in action



“Born accessible”
vs. bolt-on

Benefits for
everyone

More design approaches

- **Ability-based design** - develop systems that model and adapt to a user's ability
- **Design for empowerment** - emphasize empowering and increasing representation from marginalized groups
- **Value Sensitive Design** - consider other values in design other than usability (e.g., independence, self-determination)
 - Example: if we could, should we replace ASL interpreters with ASL-translating smart gloves? Why or why not?

But who is right?

- _ツ)
- All of these perspectives may be useful at certain times
- Not necessarily mutually exclusive
- Useful to be aware of any implicit assumptions we are carrying with us

Considering impact

- Often we are considering trade-offs between the size of the group we are including and the magnitude of the impact
- Consider:
 - Helping 1 million people do their jobs 1% faster
 - Helping 100k people gain access to education
 - Helping 500 people live independently
- In reality, we are often dealing with external constraints. But it's worth taking a moment to consider this in the abstract case.

Why make things accessible?

- Laws or regulations
- Because we want to do the right thing
- Because accessible design is good design

Laws and regulations in the US

- Section 508 of Rehabilitation Act - guidelines for access to electronic resources
- Americans with Disabilities Act - guidelines for “reasonable accommodations” to work, education, public life
- Individuals with Disabilities Education Act (IDEA) - guidelines for access to education

How to know what to do?

- In the old days, sometimes it was difficult or impossible to create accessible versions of our programs
- Fortunately, we now have better tools on our side:
 - Programming frameworks that support accessibility (if you do the right thing)
 - Guidelines for accessible design (and sometimes automated testing tools)

HTML is great for this

- Well defined standards for creating accessible content
- Web Content Accessibility Guidelines (WCAG)
- By design, the web separates content (HTML) from presentation (CSS)
- Everything is rendered as HTML, can be scraped or altered by user's own device

Remixing the Web

https://userstyles.org/styles/browse/twitter

technology | Technology... | Is the Oculus... | www.who.int... | americans wi... | Americans w... | Section508.... | WebAIM: Uni... | Twitter The... | +

stylish Search for a style Create New Style FORUM LOG IN or SIGN UP

Home Editor's Choice Most Popular Newest Styles Recently Updated

Twitter Themes & Skins

Ice Cold Coca Cola - twitte... by Stylish Updated: 3/29/17 0 38

Twitter Champions Leagu... by Stylish Updated: 3/8/17 0 23

Twitter Flat Dark & Clean by Martinius Ulberg Updated: 1/14/18 1 630

Twitter Dark Mode (2017) by dubstrike Updated: 10/10/17 0 473

New Twitter dark mode by... by Dear_VooDoo Updated: 12/16/17 0 318

Twitter - Dark n Simple by Gibbu Updated: 2/4/18 0 237

Go to "https://userstyles.org/styles/155039/a-better-twitter"

The screenshot shows the Stylish browser extension's user interface. At the top, there's a navigation bar with tabs for different websites like 'technology', 'Technology...', 'Is the Oculus...', etc., and some utility icons. Below that is a search bar and a 'Create New Style' button. The main content area is titled 'Twitter Themes & Skins' and displays a grid of six different Twitter theme styles. Each theme is shown with a preview image, the name, the creator, the last update date, and some engagement metrics (likes and saves). The themes include: 'Ice Cold Coca Cola - twitte...' by Stylish (updated 3/29/17), 'Twitter Champions Leagu...' by Stylish (updated 3/8/17), 'Twitter Flat Dark & Clean' by Martinius Ulberg (updated 1/14/18), 'Twitter Dark Mode (2017)' by dubstrike (updated 10/10/17), 'New Twitter dark mode by...' by Dear_VooDoo (updated 12/16/17), and 'Twitter - Dark n Simple' by Gibbu (updated 2/4/18). A large white Twitter logo is centered at the bottom of the page.

Making HTML accessible

- We'll get into the guts of how to do it in code on Wednesday
- But first, let's think about what it actually means
- So: what does it mean for a web application to be accessible?

Ways to think about accessibility

- Following standards - this helps our design work on a variety of current and future devices
- Presenting content in multiple ways - since people may consume media in different ways, have we included the appropriate content for doing so?
- Supporting various input methods - including mouse, touch, keyboard

More ways to think about accessibility

- Using appropriate accessibility features of HTML (e.g., alt tags for images, ARIA labels for dynamic user interfaces)
- Semantic UI code - should represent the logical structure of the interface, not just visual
- Support end-user customization - e.g. font size, color contrast

Where we go wrong

- Writing buggy UI code
- Neglecting to include alternative representations (image descriptions, video captions) - or including lousy versions of these
- Getting the visual presentation right but messing up the semantic representation (no **semantic UI**)

Semantic UI?

```
<h1>My favorite animals</h1>
<ul>
<li>Cat</li>
<li>Goat</li>
<li>Capybara</li>
<li>Axlotl</li>
</ul>
```

```
<p><strong>My favorite animals
</strong></p>
<p>• Cat</p>
<p>• Goat</p>
<p>• Capybara</p>
<p>• Axlotl</p>
```

My favorite animals

- Cat
- Goat
- Capybara
- Axlotl

Project 1

- Due in 3 weeks (Feb 25th)
- Build a small interactive app in HTML/CSS/JavaScript
- **Let's spend some time planning this out**