UCI University of California, Irvine

Informatics 134

Software User Interfaces
Winter 2022

Mark S. Baldwin
baldwinm@ics.uci.edu

2/10/2021

Agenda

1. Upcoming

2. Graphical Toolkits and Accessibility

3. References

Upcoming

Upcoming

- Accessibility and Toolkits lecture
- Next week: Team evals on Tue, and lecture on Thur
- Keep working on A3 (Part 1 DUE 2/15)
- Keep working on T3 (DUE EOQ, Final Presentations Start 3/08)

Graphical Toolkits and Accessibility

What is accessibility?

Historical Perspective

Transition from the terminal to a GUI

- Ushered in personal computing era (good)
- Ushered in the accessibility era (not so good)

[Wikipedia, 2021, theverge.com, 2021]

Historical Perspective

Although millions of new people were now **able** to understand and make use of computational systems, millions of people were simultaneously **unable** to use and operate new graphical based systems.

[theverge.com, 2021]

Accessibility

The Graphical User Interface Crisis: Danger and Opportunity

> ‘‘Our intuition tells that the more an interface is optimized for a person who can see, the less useful that computer will be to people who cannot see.’’
>
> ——[Boyd et al., 1990]

Accessibility

Historical Perspective

According to [Boyd et al., 1990], the problem manifested in two major categories:

1. Perceptual: Screen rendering via pixels requires deciphering of graphical information
2. Control: Interaction with visual representations of information and manipulation and control of the flow of information

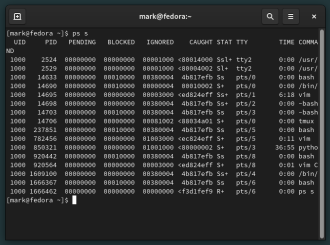
[theverge.com, 2021]

Accessibility

Historical Perspective

WIMP (?):

- Transition from a concise command language
- To visual metaphors – graphical representations of everyday objects



[theverge.com, 2021]

Historical Perspective

Gaining Access: Enter the Screen Reader

- IBM Screen Reader/DOS (1984)
- IBM Screen Reader/2 (1986-1994)

...

IBM adapted early work on speech synthesis to create SAID, the Synthetic Audio Interface Driver [Thatcher, 1994].

Historical Perspective

Gaining Access: Enter the Screen Reader

> Early version of SR/2 relied on a separate custom keyboard to control speech synthesis to avoid system conflicts!
>
> The key principle from Thatchers work was that text-based DOS and GUI are different interfaces for doing the same thing. So, the solution was to map GUI to textual equivalents [Thatcher, 1994].
>
> ‘‘Abstract away what is *graphical* about the graphical user interface.”

Historical Perspective

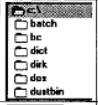
From *access* to *degrees of access*

- By mid-1990's focus changed to efficiency, coherence, exploration, and cost
- Refinement of the abstraction – developing a consistent, reusable, lexical understanding of graphical widgets
- Locate widgets without a mouse, support exploration, interact without clicking
- Consistent mental model (WHY?)

Historical Perspective

From *access* to *degrees of access*

- The Mercator Project (Georgia Tech, Center for Rehab Technology)
- GUIB: Textual and Graphical User Interfaces for Blind People (European Commission, Technology Initiative for the Disabled and Elderly Persons or TIDE)

interaction object	example	braille-based presentation	example	speech-based presentation	example	nonspeech-based presentation	example
caret, mouse pointer	text	one braille character	t■xt ■	text around caret is spoken	"e"	audio "click" at caret	t e click x t
text, text attributes	sample	braille, attributes through dots 7 and 8 or on request	sample	text is spoken, attributes are verbalized (??)	"sample"	pitch of speech is modified for attributes	
window		window frame in braille name is spoken pop-up, move, size by sound	+[-] Notepad-[Unt [↓] [↑] |F|File E|Edit S|Search +----------	name is spoken	"notepad"	auditory icon for window object, modified for popup, iconify, or focus	tapping on glass sound
icon		name in braille	[Icon] Dustbin		"dustbin"	auditory icons	sound of dropping something in a trashcan
menu		all items in braille vertical or horizontal layout	F|File O|Options W|Wi *A■Auto Arrang M|Minimize on *S|Save Settin +----------	new selection is spoken	"auto arrange"	auditory icon for menu-button, pitch is modified relative to location in menu	flipping sound at a high pitch
scroll bar		in braille	■-□----------■	status is verbalized	"slider at zero percent"	auditory icon for scollbar, location conveyed with pitch	slide whistle sound, low pitch
edit field	winword.exe	in braille, selection with dots 7 and 8	|winword.exe	text is spoken	"winword dot exe"	auditary icon for editable text field	sound of an old-fashioned typewriter
list box		all items in braille	+----------+ c:\ batch bc dict dirk dos dustbin +----------	selection is spoken	"c colon backslash"	auditory icon for list, pitch is modified relative to location in list	line-printer sound at a high pitch
button		in braille	[Y|Yes]		"yes"	auditory icon for push button	sound of pushing an old elevator button

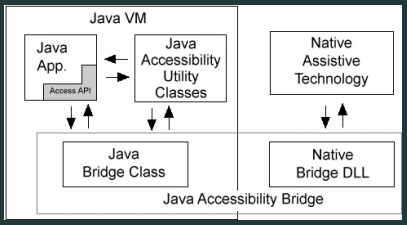
[Mynatt and Weber, 1994]

Accessibility

Historical Perspective

From *access* to *degrees of access*

Projects like Mercator and GUIB influenced the design and implementation of accessibility API's in programming languages like Java, .Net, Cocoa (OSX)



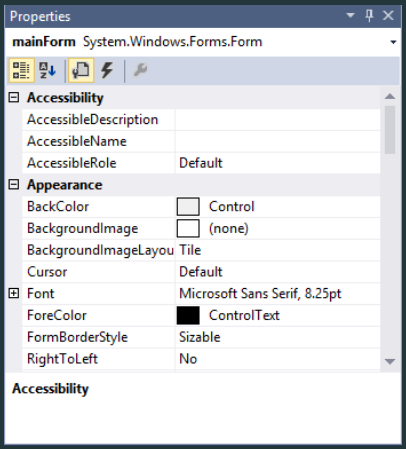
[Harper et al., 2005]

Accessibility

Historical Perspective

From *access* to *degrees of access*

An example from Microsoft Visual Studio



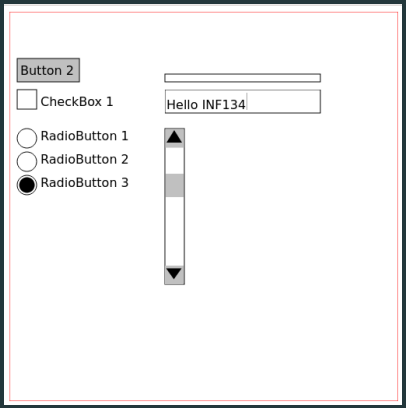
When converted to code:

```
1 //describes control
2 mainForm.AccessibleDescription = "the main form";
3 //name reported to accessibility aids
4 mainForm.AccessibleName = "My Program";
```

learn more

Historical Perspective

How would you make your widgets accessible?



From *degrees of access* to *supporting access*

Accessibility

Supporting Access

The WebAIM Million: 2019-2021

- Over 51 million a11y errors, avg of 51.4 errors per page
- 97.4% of home pages had detectable WCAG2 failures
- 65% missing alt text for images
- 25% empty buttons or button misuse

Today we *have* the ability to make software accessible, but we lack the *desire or awareness* to make it so.

Why is the second example *more* accessible?

```
1  <div>Next</div>
```

```
1  <button>Next</button>
```

Why is the second example *more* accessible? Semantics

```
1  <div>Next</div>
```

```
1  <button>Next</button>
```

Which example is *harder* to implement?

Semantic HTML

They require the same amount of effort :) Plus:

- Easier to reason about, less code (don't have to add accessibility)
- Therefore...lighter/faster/better optimization (SEO, etc.)
- Browser accessibility engine translates accessible functions automatically (*e.g.*, Tab and Enter keys, purpose)

see: HTML: A good basis for accessibility for a more detailed overview.

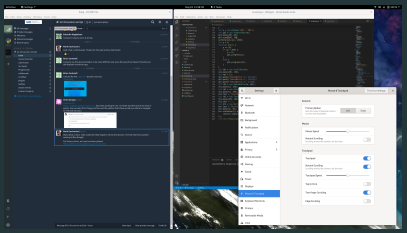
The Visuospatial Sketchpad

Making graphical interfaces accessible is more than just semantics. In his work on human working memory, Alan Baddeley [Baddeley, 1992] identifies the visuospatial sketchpad as:

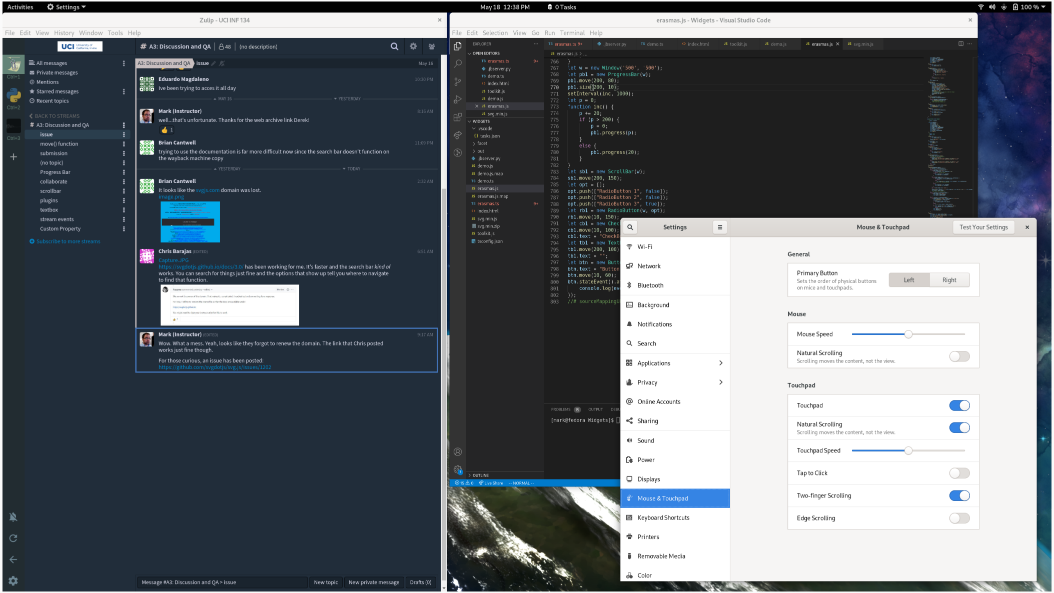
> ‘‘virtual environment for physical simulation, calculation, visualization, and optical memory recall.”
>
> Move optically retained information from short-term to long-term memory and back again
>
> Think about how humans recognize faces
>
> visual cache - holds information about form and color
>
> inner scribe - manages spatial and movement information

The Visuospatial Sketchpad

Take a minute to study the screen shot pictured here. Can you identify some ways that information is communicated visually?



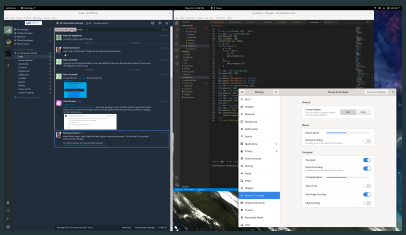
Remind Mark: Full screen image on next slide ;)


Activities
Settings
May 18 12:38 PM
0 Tasks
100 %
Zulip - UCI INF 134
File Edit View History Window Tools Help
A3: Discussion and QA
48
(no description)
All messages
Private messages
Mentions
Starred messages
Recent topics
BACK TO STREAMS
A3: Discussion and QA
issue
move() function
submission
(no topic)
Progress Bar
collaborate
scrollbar
plugins
textbox
stream events
Custom Property
Subscribe to more streams
Eduardo Magdaleno
Ive been trying to acces it all day
Mark (Instructor)
well...that's unfortunate. Thanks for the web archive link Derek!
Brian Cantwell
trying to use the documentation is far more difficult now since the search bar doesn't function on the wayback machine copy
Brian Cantwell
It looks like the svgjs.com domain was lost.
image.png
Chris Barajas
Capture.JPG
https://svgdotjs.github.io/docs/3.0/ has been working for me. It's faster and the search bar kind of works. You can search for things just fine and the options that show up tell you where to navigate to find that function.
Mark (Instructor)
Wow. What a mess. Yeah, looks like they forgot to renew the domain. The link that Chris posted works just fine though.
For those curious, an issue has been posted:
https://github.com/svgdotjs/svg.js/issues/1202
Message #A3: Discussion and QA > issue
New topic
New private message
Drafts (0)
erasmas.js - Widgets - Visual Studio Code
File Edit Selection View Go Run Terminal Help
let w = new Window('500', '500');
let pbl = new ProgressBar(w);
pbl.move(200, 80);
pbl.size(200, 10);
setInterval(inc, 1000);
let p = 0;
function inc() {
p += 20;
if (p > 200) {
p = 0;
pbl.progress(p);
}
else {
pbl.progress(20);
}
}
let sbl = new ScrollBar(w);
sbl.move(200, 150);
let opt = [];
opt.push(["RadioButton 1", false]);
opt.push(["RadioButton 2", false]);
opt.push(["RadioButton 3", true]);
let rbl = new RadioButton(w, opt);
rbl.move(10, 150);
[mark@fedora Widgets]$
Settings
Mouse & Touchpad
Test Your Settings
Wi-Fi
Network
Bluetooth
Background
Notifications
Search
Applications
Privacy
Online Accounts
Sharing
Sound
Power
Displays
Mouse & Touchpad
Keyboard Shortcuts
Printers
Removable Media
Color
General
Primary Button
Sets the order of physical buttons on mice and touchpads.
Left
Right
Mouse
Mouse Speed
Natural Scrolling
Scrolling moves the content, not the view.
Touchpad
Touchpad
Natural Scrolling
Scrolling moves the content, not the view.
Touchpad Speed
Tap to Click
Two-finger Scrolling
Edge Scrolling

Accessibility

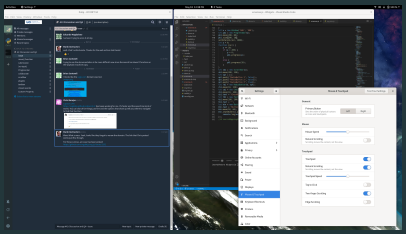
The Visuospatial Sketchpad

- Visual glance
- Spatial arrangement
- Color



Accessibility

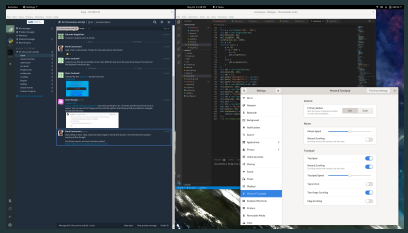
How do we make these visual enhancements more accessible?



Accessibility

How do we make these visual enhancements more accessible?

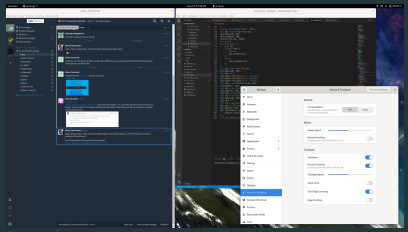
- Hierarchy!
- Shortcuts (skip links, etc.)
- Color alternatives, high contrast



Accessibility

How do we make these visual enhancements more accessible?

- Hierarchy!
- Shortcuts (skip links, etc.)
- Color alternatives, high contrast



Accessibility

How do we make these visual enhancements more accessible?

Some thoughts from my own work...

- Semantic versus lexical
- Semantic information as haptic information
- Visuospatial to tangible

Demo Video

Accessibility

Let’s return to the topic of supporting access...

Today we *have* the ability to make technology accessible, but we lack the *desire or awareness* to make it so.

Access to technology is democratizing. It is our responsibility, as creators of technology to accommodate all people, regardless of ability.

Accessibility

What are some ways that you can be an ally and advocate for accessible design?

References

References i

- Baddeley, A. (1992).
  Working memory.
  Science, 255(5044):556–559.
- Boyd, L. H., Boyd, W. L., and Vanderheiden, G. C. (1990).
  The graphical user interface: Crisis, danger, and opportunity.
  Journal of Visual Impairment & Blindness, 84(10):496–502.
- Harper, S., Khan, G., and Stevens, R. (2005).
  Design checks for java accessibility.
- Mynatt, E. D. and Weber, G. (1994).
  Nonvisual presentation of graphical user interfaces: Contrasting two approaches.
  In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, page 166–172, New York, NY, USA. Association for Computing Machinery.

References ii

- Thatcher, J. (1994).
  Screen reader/2: Access to os/2 and the graphical user interface.
  In *Proceedings of the First Annual ACM Conference on Assistive Technologies*, Assets '94, page 39–46, New York, NY, USA. Association for Computing Machinery.
- theverge.com (2021).
  40 years of icons: the evolution of the modern computer interface.
- Wikipedia (2021).
  Computer terminal.