Team Project: A11y Bootcamp

Team Project: A11y Bootcamp

Usability, Accessibility, and more

Volker Sorge

v.sorge@cs.bham.ac.uk

University of Birmingham, UK cs.bham.ac.uk/~vxs

Exercise 1

- Login
- Open the Chrome browser
- Navigate to https://zorkow.github.io/team-project
- Tab to the 2022 link
- Press Return

Overview

- What is Usability
- What is A11y: Accessibility?
- Persona
- Desktop and Web Apps
- Testing
- Assessment Remarks
- General Remarks

Usability

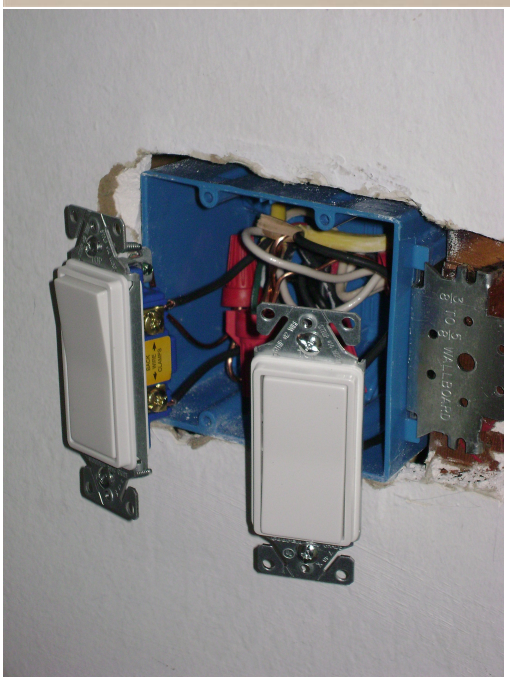
- Usability according to Wikipedia is the capacity of a system to allow users to perform tasks
  - safely,
  - effectively, and
  - efficiently
  - while enjoying the experience.

Unfortunately,

- Functionality is often easy to achieve
- Making it usable is much harder

General Usability

- Example: Light switch
- Find it and switch the lights on (or off)



General Usability (2)

- Example: Light switch
  - safe: If there are no life wires!
  - effective: Does it switch the light on/off?
  - efficient: Can I find it in the dark?
  - enjoyable: maybe if you have a dimmer...
- Just inserting a switch in a circuit is easy
- But it is important to put it somewhere reachable, safe, operable by anyone regardless of size, strength, age, etc.

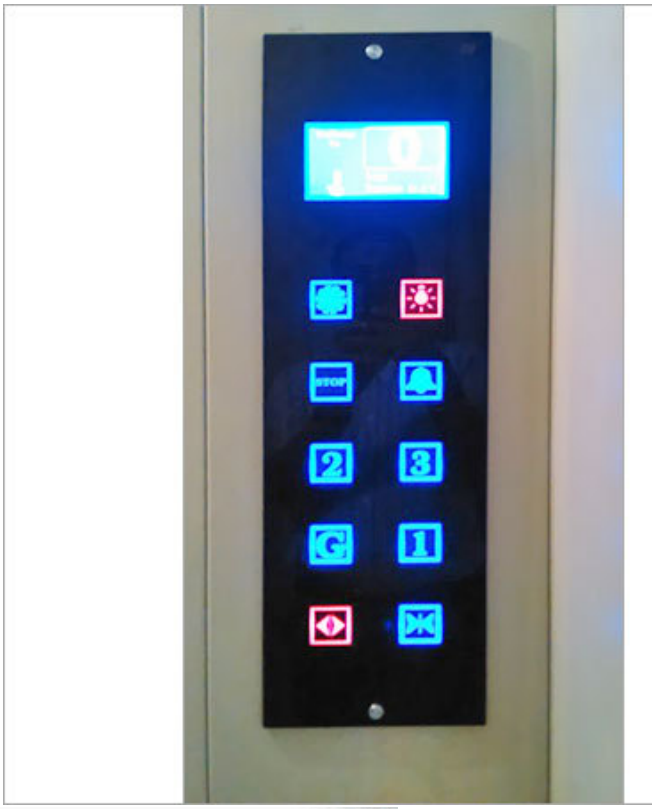
General Unusability

- Example: The Microwave Panel
- What are all these buttons for?

Particular Unusability

- Example: Elevator Panel
- Find your floor and get there
- ... if you can


WE WANT
YOU SAFE
1
5
M
4
G
3
B1
2


2
3
G
1


45
46
47
48
LP
UP
PH
38
39
40
41
42
43
44
31
32
33
34
35
36
37
24
25
26
27
28
29
30
17
18
19
20
21
22
23
9
10
11
12
14
15
16
SL
L
4
6
7
ALARM
CAPACITY 3500 LBS
Thank You For

Software Usability

Traditionally Usability has been equated with Accessibility. In software it is an important concept for all!

- effective: Is it serviceable for what I want to do?
- efficient: Does it help me to do my task or does it get in the way?
- enjoyable: The first two often define this!
- safe: How severe are mistakes?

Additional aspects

- learnability and memorability
  - Do I need a certificate to understand the software?
  - Do I have to relearn after not using a system for a while?

General Usability and Accessibility

A product should be usable by everyone regardless of age, disability or special needs

- Particular aspect of Usability is Accessibility
- Usability for one is unusablity for another
- Try to find a common ground
- General Principle of Universal Design

Importance and misconceptions

- Usability has to look good
  - Design is not equal to aesthetics
- Usability cannot be measured
  - Not true. E.g. Fitts's law for average time to complete a task with point-and-click

$$ T=a+b\log _{2}{\Bigg (}1+{\frac {D}{W}}{\Bigg)} $$

- Accessibility is expensive
  - Maintaining a poorly designed system is more expensive
  - Losing users is losing customers
  - Law-suits cost money!

Designing for everyone is important!

Fitts's Law Variables Explanations

- T is the average time taken to complete the movement.
- {\displaystyle a}represents the start/stop time of the device and
- {\displaystyle b} stands for the inherent speed of the device.
- These constants can be determined experimentally by fitting a straight line to measured data.
- {\displaystyle D} is the distance from the starting point to the center of the target.
- {\displaystyle W} is the width of the target measured along the axis of motion.

- {\displaystyle W} can also be thought of as the allowed error tolerance in the final position, since the final point of the motion must fall within
- {\displaystyle \pm {\frac {W}{2}}} of the target's centre.

Universal Design

A product and a process

- Design that is usable by all people
  - Not always possible!
- Design that works for as many people as possible
  - Design to extent the reach of your product to a wide audience
- Design that has no need for adaptation
  - Do not design special cases, add-ons, extra layers, etc.

Accessibility

Usability for users with special needs

Visual: blindness, low vision, impaired vision, distracted vision

Aural: deaf, hard of hearing, distracted hearing

Movement: limited use of extremities, slow reaction time, limited fine motor skills

Cognition: Dyslexia, Dyscalculia, Distraction, Memory deficits

Is this really important?

- Is this not something for specialists only?
  - Every software engineer needs to understand basic accessibilty considerations
  - Mistakes at back end design, data structures, development stack selection can destroy the ability to make software accessible
- Is designing for everyone important?
  - Every user facing software should be accessible
  - Maintaining a poorly designed system is more expensive
  - Losing users is losing customers
  - Law-suits cost money!

We are all not fully able at some point in our life

Types of Disabilities

- Permanent
- Temporary
- Situational

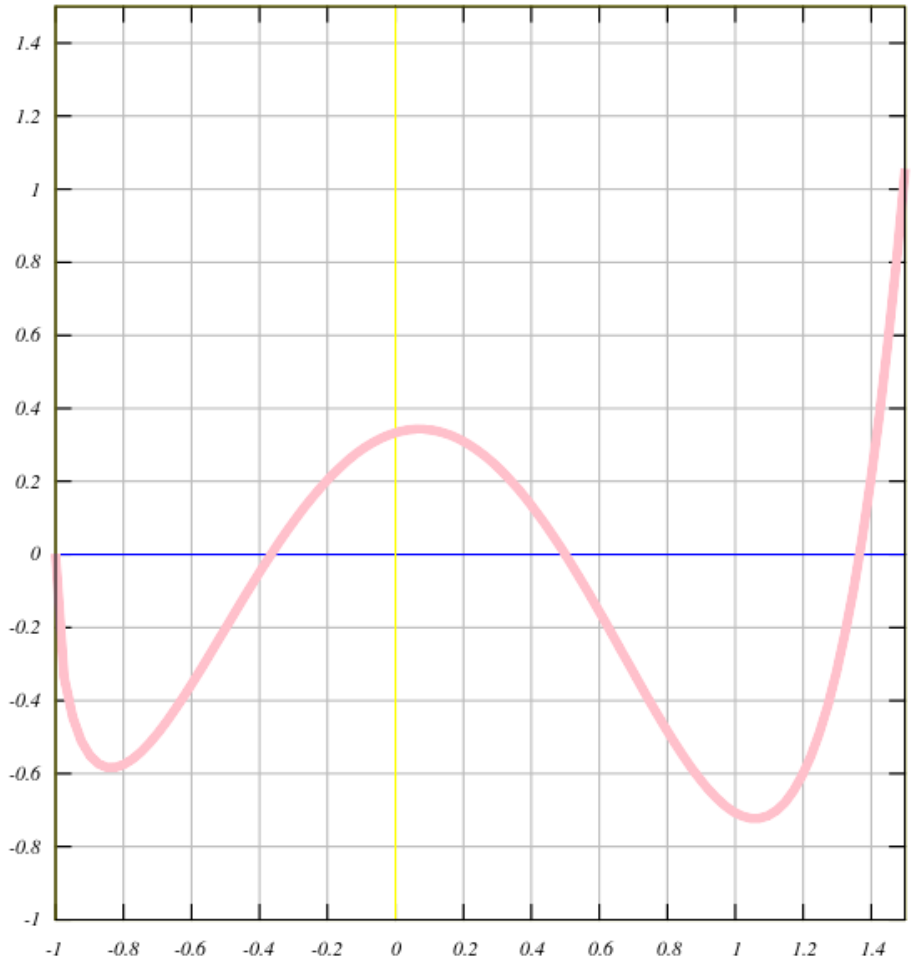
Examples:

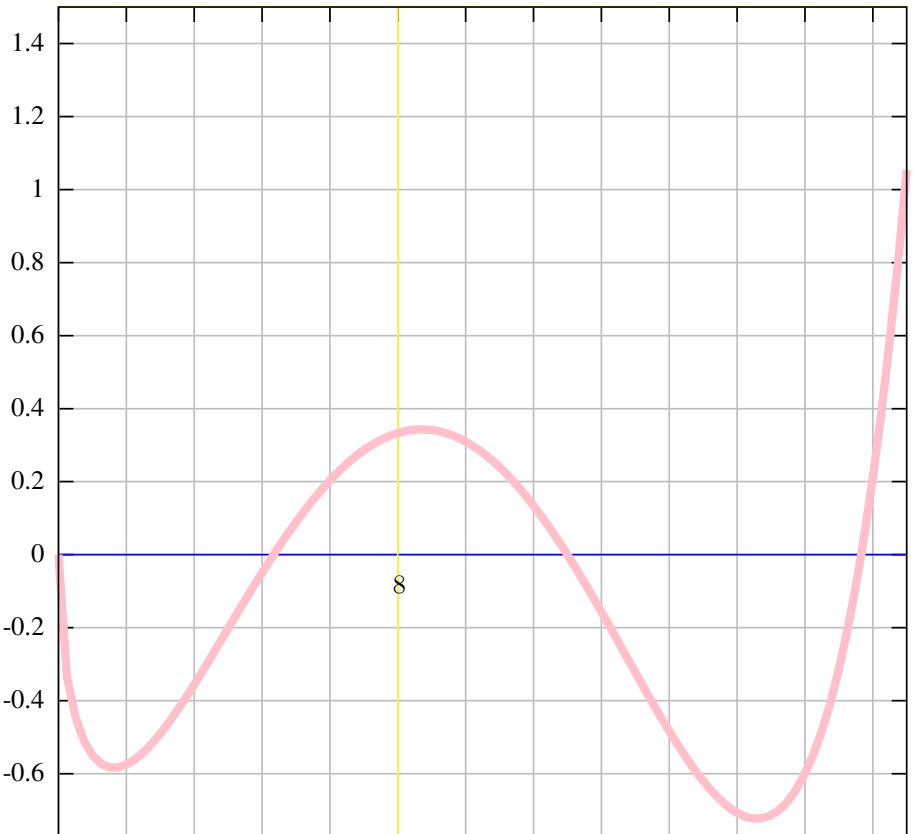
- Vision: Blind — after eye operation — sitting in the sun
- Sound: deaf — ear infection — noisy environment
- Motor: loss of limb — broken arm — carrying shopping bag

From Specialist Support to Mainstream

- Voice output: designed BVI community
  - Everyone listens to audiobooks, home assistants, etc.
- Voice input: designed for motor impaired people
  - We all talk to our assistants, SatNavs, etc.
- Subtitles: Originally designed for DHH community
  - Many watch with CC by default
- Dark modes: aimed at low vision and dyslexia support
  - Now more important than ever...
- High contrast, Large fonts, Screen magnification, ...

Dark Mode Example





Defining Accessibility: POUR Principles

WCAG's core principles: POUR

For **all** users, **all** content must be

- Perceivable
  - Alt text, sub-titles
- Operable
  - Keyboard, touch
- Understandable
  - Language, Icons
- Robust
  - Not just for OS X, Browser Y or screen reader Z

Some Accessibility Concepts

- Keyboard accessibility
  - Every task should be a achievable with keyboard only
- Visual adaptability
  - Fonts can be enlarged or even changed
  - Colour palette supports high contrast
  - Magnification and zoom available
- Provision of Alternative media
  - Visual aids are supplied with alternative descriptions
  - Sounds or voice output is subtitled or replaced by visual clues

Accessibility Personas: Guide

Create Personas that cover some of the major a11y concepts

Some helpful resources:

- Four Example Personas
- UK Government
- UK Government sources

Also think about how it can improve the User Experience of every user

- Different ages,
- Environment of use of software
- ...

A11y in Design Workflow

- Not just a front end consideration
- Consider all the information you need
- Make sure that your data structures a general enough for all use cases
- Ensure your back-end exhibits everything you need at the front-end
- Avoid premature optimisation
  - Do not throw information away that you might need at a later point!

Designing with A11y in Mind

Do not try to retro-fit as this is often impossible!

Good approaches:

1. Think about all the users that can benefit from your product
2. Consider as many corner case as possible
3. What are the requirements for your software to satisfy these?
4. Where do you need to make allowances for different needs?

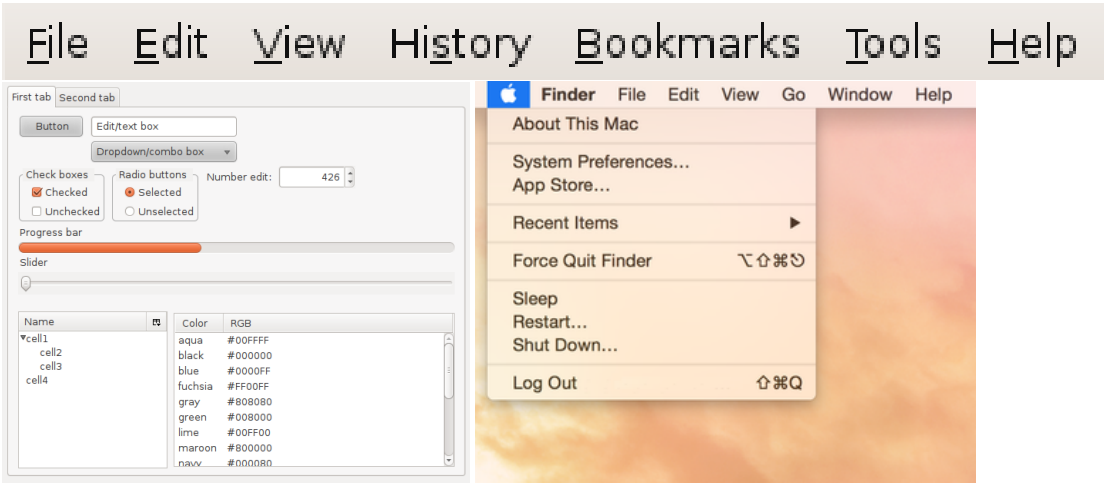
Non-Accessibility is expensive: Lawsuits cost money!

Accessibility in Practice

- Desktop programs
- Web applications
- Mobile apps

Accessibility of Desktop Applications

- Standard widgets get accessibility support from OS
- Careful with transitions: Content vs "Chrome"



- Avoid overly handcrafted UIs
- Traps and pitfalls: **dead-ends**
  - Classic dead-end in keyboard accessibilty: A field you tab into but never get out without using the mouse.

Web Applications

- No standard layout for your web components
- Widgets can be build in a myriad of ways
- Example: drop down menu
  - A bad example
  - A good example

Use the web trinity:

- HTML/CSS: Basic elements with styling or Syntax
- JavaScript: Functionality
- ARIA: Describe meaning or Semantics

Making Web Components Accessible

- Use a reasonable page layout
- Use logical structures like `header`, `main`, `footer`, etc.
- Roles to describe
  - standard widget for menu, interaction elements and states, ...
  - content regions, headings, tables, ...
- Navigation enhancements via landmarks
  - Complement the HTML regions
- Keyboard navigation via `tabindex`
  - Finetune using roving tabindices
- Live regions to alert to changing content

Mobile Devices

- Very similar to web accessibility
- Equally important
- But even harder

Exercise 2

- Get your phone out
- Find accessibility settings
- Switch on VoiceOver (iOS) or TalkBack (Android)
- ... good luck

Example: Using the back button consists of two actions.

- Visual task
  1. Perception: See the button with backarrow
  2. Operation: Touch the button to trigger action
- Non-visual task
  1. Perception: Hear the button's function with initial tap
  2. Operation: Tap the button again to trigger action

A11y Testing

Do some simple tests that consume very little time

- Zoom
  - How does content react when zooming in or out?
  - Just a few key strokes
- High Contrast
  - What if you change the colour palette?
  - One Chrome extension, one key stroke
- Keyboard interaction

  - Can you reach everything without a mouse?
  - Multiple keystrokes

See in desktop settings of your OS how to use them.

A11y Testing (2)

- Form Factors (desktop, laptop, tablet, mobile ...)
  - Simulate in browser
- Web accessibility with the Lighthouse tool in your browser
- Screen reading
  - Try a screen reader yourself
    * Windows: Narrator (Windows + Enter), NVDA, Jaws,
    * Apple Mac: Voiceover (Command + F5)
    * Linux: Orca (Super + Alt + S)
    * Chrome/ChromeOS: ChromeVox
    * iOS: Voiceover (in Settings)
    * Android: TalkBack (in Settings)
- ...

Assessment

We will test your product for accessibility. In particular

- Keyboard accessibility
- Visual adaptability
  - Magnification and zoom
  - Fonts
  - Contrasts and colour
- Screen reading compatibility
- Provision of alternatives
  - text or sound for graphics
  - visual cues for sounds Other features depending on the particular application you implement

Assessment Tips

Document all your accessibility efforts

- highlight what works
- describe what is challenging
- document limitations
- do not try to hide them

There can be parts you can not make accessible

- Discuss what are the problems are
- What could be a possible solution?

General Remarks

- Organise your team
  - Assign responsibilities and roles

  - Choose an Accessibility Evangelist
- Organise your communication
  - Chat platform with Video everywhere (Discord or Signal over WhatsApp)
- Organise your coding
  - Choose a coding and commenting style
  - Use Merge requests and code reviews
- Organise your continuous integration
  - Think about semantic versioning
  - Meaningful commit messages
  - Automate release notes

Myself as a Resource

- Make use of my experience on some of the above
- Email me, or drop me a chat on deparmental zoom or teams
  - I am often slow to reply!
  - Ping me again after a day or two.
- I will announce weekly drop in sessions for A11Y issues (on zoom, teams, or similar)

But note:

- I can give you general advice
- I will **not** solve your specific problems, or debug your software

If you have more interest in Accessibility, talk to me

Wednesday's Drop-Inn Session

Wednesday 11-14

- Q&A only
- Mop-up session
- I will not present prepared content
- One person per team
- Send you Accessibility Evangelist

Exercise 3

- Switch off the screen reader (if you've switched it on)
- Log off

A11Y Resources: Testing per OS

- For desktop
  - Windows
  - MacOS
  - Linux... not many resources
- For web applications (mostly for webkit based browsers)
  - WCAG Access Audit UI
- Accessibility Insights for the Web

- WebAIM WAVE Evaluation Tool
- For mobile
  - Android Accessibility
  - iOS Accessibility

A11Y Resources

- Microsoft
- Google
- Apple
- Amazon
- IBM

A11Y Guidelines

- Web Content Accessibility Guidelines (WCAG)
  - ~60 "Success Criteria" across 3 Levels: A, AA, AAA
  - A&AA is the legal basis in most countries, AAA optional/ideal
  - many criteria cover forms & applications, not "just" content
  - Accompanying specs:
    - Understanding WCAG
    - WCAG Quick Reference
- Shorter: W3C Accessibility Principles WebAIM WCAG Checklist
- More: WebAIM article