Class 8 - Pointing, Critique, jQuery

8.1 Pointing[1]

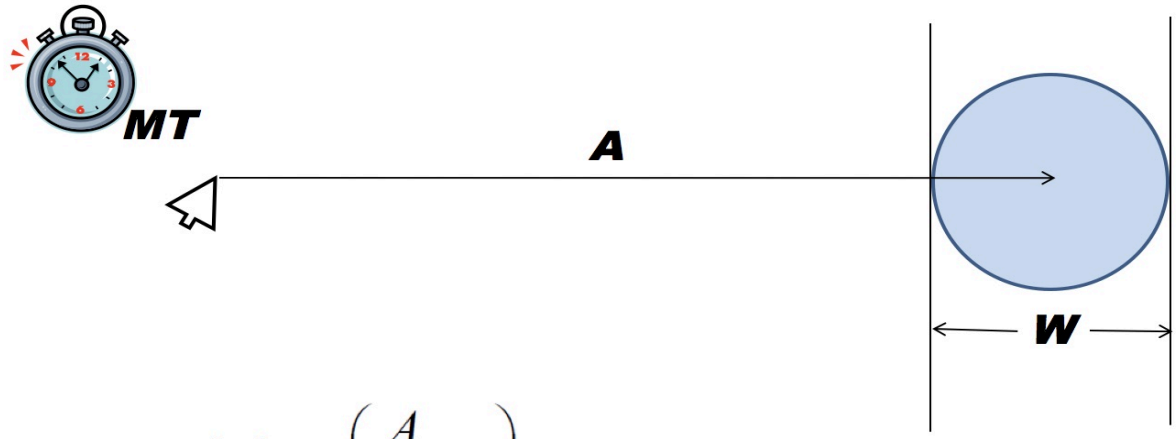
We start discussing specific concepts of interaction and the theory that underlies them. The first topic is *pointing*. We use fingers to point, in order to indicate the identity of something (that table, this picture, etc.) Pointing is used at a surprisingly frequent rate each day, which explains why it has been such a powerful paradigm in user interfaces (that button, this link, etc.)

8.1.1 Fitts's law

In the early 1950s, Paul M. Fitts was very interested in modeling human performance of pointing. He created the Aviation Psychology Research Laboratory in the Ohio State University, and worked on human factors engineering. He developed predictive models about pointing in order to help design dashboards, cockpits, and other aviation related components. His focus was on "aimed" movements, in which a person has a target and must move their pointing device (a hand) to indicate that target. This is an example of a "closed loop" motion:

- A system (human)
- Reacts to its evolving state (human finger is in space relative to its target)
- A person's reaction is the continuous correction of the trajectory as they move toward a target

Fitts started measuring this closed loop, searching for a pattern, and eventually found what we call *Fitts' Law*:



$$MT = a + b \cdot \log_2\left(\frac{A}{W} + 1\right)$$

Fitts' law (1954)

[1] Amy J. Ko et al. User Interface Software and Technology, Chapter 8; https://faculty.washington.edu/ajko/books/user-interface-software-and-technology/pointing

Here:

- MT (motion time) is the amount of time it would take for a person to move their pointer to precisely indicate a target.
- A is distance to the target
- W is the size ("width") of the target. The units of measure for A and W must be the same.
- a is a user- and device-specific constant. It represents the minimum time to move. This varies by person and device, accounting for things like reflexes and latency in devices.
- b is a measure of how efficiently movement occurs. For example, it may be harder to move a very sensitive mouse

Based on Fitts' Law what will increase chances of scoring on a soccer penalty kick? (Choose one correct statement)

1. Quicker foot release (larger b)
2. Quicker foot release (smaller a)
3. Heavier ball (larger b)
4. Heavier ball (smaller a)

Answer: quicker foot release results in smaller a, and faster time to target. Heavier ball results in less efficient movement, larger b, and longer time to target.

There's one critical detail missing in Fitts' law: failures to hit the target, i.e. errors. For example, you are trying to answer the phone on a touch screen and instead hang up. Jacob Wobbrock studied this gap and determined that Fitts' law implies a speed/accuracy tradeoff:

- The faster one moves during pointing, the less likely one is to successfully reach the target
- Also, error rates have different sensitivity to factors:
    - a and b strongly influence error rates
    - W matters more than A

8.1.2 Pointing in user interfaces

There are several implications of Fitts' formula for user interfaces:

- What if $W = \infty$?
    - $MT = a + b \times log_2(A/\infty + 1) = a + b \times log_2 1 = a$
        - That is, the movement time becomes minimal

    - Example:
        - The Apple menu is always placed at the top of the screen. The operating system always constraints the mouse position to be within the screen's boundaries, no matter how past the top of the screen you are trying to move the mouse.
- It turns out Fitts' law applies to any kind of pointing: using a mouse, a touch screen, a trackball, a trackpad, etc.
    - *Direct pointing* devices: input and output occur in the same physical space; e.g., a touch screen
        - Limitation: occlusion, where a person's hand obscures output
    - *Indirect pointing* devices: provide input in a different physical space from output; e.g., a mouse
        - Limitation: requires a person to pay attention to two different places
- Absolute vs relative pointing
    - *Absolute* pointing includes input devices where the physical coordinate space of input is mapped directly onto the coordinate space in the interface
        - Example: bottom left of the touch screen is the bottom left of the interface
    - *Relative* pointing maps changes in a person's pointing to changes in the interface coordinate space.
        - Example: moving a mouse to the left by an inch results in moving a cursor on the screen some number of pixels to the left. Note that this allows for a variable *gain* (i.e. number of pixels) based on user's preference

Consider two screens placed next to each other. What makes moving from one screen to another faster and less prone to errors?

1. Making a mouse cursor larger
2. Making a text cursor larger
3. Making a text cursor smaller
4. Making a mouse cursor smaller

Answer: making a mouse cursor larger decreases A, and effectively makes the target larger (easier to hit).

8.1.3 Making pointing better

One way to make pointing better is to focus on coefficients a and b. There have been several inventions in this area.

- Multi-touch mouse allowing users to provide input using multiple touches; see https://dl.acm.org/doi/10.1145/1622176.1622184
- Preserving physical measurements of input rather than mapping them to integer spaces
    - This enables sub-pixel precision
- LightRing is a new type of pointing device involving sensing infrared proximity between a finger and a surface point; see https://youtu.be/3_GbVIo6iq0 [2 min]

Other innovations focus on software, and aim to increase target size or reduce travel distance:

- Target-agnostic
    - Making a mouse pointer accelerate when it determines the user is trying to travel a large distance.
        - Note: this technique is not based on knowing the target, rather just knowing that the pointer is moving fast
    - Angle Mouse analyzes the angles of movement trajectory, reducing gain when a user is trying to "turn"
- Target-aware
    - The Bubble Cursor is an area cursor that dynamically resizes a cursor's activation area based on proximity to the target. Such cursors make it easier for people with motor impairments to click on targets.
        - One improvement to the Bubble Cursor is the Bubble Lens; see https://youtu.be/9b8QqLungzc
    - Enhanced area cursors use magnification and goal crossing to reduce the need for correction-phase pointing. This is particularly beneficial in case of small target size and to help users with motor impairments; see https://dl.acm.org/doi/10.1145/1866029.1866055 [2 min]
    - *Crossing* involves moving across a "goal line" instead of pointing and clicking

All this research uses Fitts' law fundamentals to achieve faster pointing. Since pointing is such an important part of how we interact with computers, making it fast and smooth is key to allowing a person to focus on their task and not on low-level details. This is particularly true of people with motor impairments.

8.2 jQuery[2]

jQuery is a natural progression in learning web user interface development as it assumes basic knowledge of:

- HTML
- CSS
- JavaScript

[2] jQuery Tutorial; https://www.w3schools.com/jquery/default.asp

jQuery is a lightweight (i.e., "write less, do more") JavaScriptLibrary. Its purpose is to make it much easier to use JavaScript. It takes a lot of common tasks that require multiple lines of JavaScript code to accomplish and streamlines them into methods that can be called with a single line of code. While there are a lot of other JavaScript libraries available, jQuery is arguably the most popular and extendable.

jScript key components/syntax are:

- Include jQuery from a CDN (Content Delivery Network) such as Google:
  - `<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>">`
- General syntax to select HTML elements and perform some action:
  - `$(selector).action()`
- Methods to manipulate DOM (HTML elements and attributes):
  - `$(selector).text(), $(selector).html(), $(selector).val()`
- Communication with the server via AJAX
  - AJAX is a protocol of exchanging data with the server without releasing the whole page
  - `$(selector).load(URL,data,callback), $.get(URL, callback), $.post(URL,data,callback)`

Example 8.1. We implement example 6.6 (prototyping navigator and login) using jQuery.

```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>INVESPAI - Menu Prototype</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"><
/script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <script>
        let DB = {
            "user1": ["User 1 Portfolio"],
            "user2": ["User 2 Portfolio"],
            "user3": ["User 3 Portfolio-1", "User 3 Portfolio-2"]
        };

        function processLogin() {
            // Empty the list first
            $('#portfolioList').html("");

            // If found, set up all available portfolios per user
            $.each(DB[$('#username').val()], function(index, value) {
                $('#portfolioList').html($('#portfolioList').html() + '<li>' + value
+ '</li>')
            });
```

```
        }

        $(document).ready(function() {
            $("#submitLogin").click(function() {
                processLogin();
            });
        });
    </script>
</head>

<body>
    <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
        <div class="container-fluid">
            <a class="navbar-brand">INVESPO</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#collapsibleNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>
            <div class="collapse navbar-collapse" id="collapsibleNavbar">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="#">Customer Service</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Open Account</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" data-bs-toggle="modal"
data-bs-target="#modalLogin" href="#">Login</a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" role="button"
data-bs-toggle="dropdown">Portfolios</a>
                        <ul class="dropdown-menu" id="portfolioList">
                        </ul>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="modal fade" id="modalLogin">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-body">
                    <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
                    <form>
                        <div class="my-3">
                            <input type="text" id="username" class="form-control" />
                            <label class="form-label"
for="username">Username</label>
                        </div>

                        <div class="my-3">
                            <input type="password" id="psw" class="form-control" />
                            <label class="form-label" for="psw">Password</label>
                        </div>
                        <button type="button" class="btn btn-primary my-3"
id="submitLogin">Submit</button>
                    </form>
                </div>
```

```
            </div>
        </div>
    </div>


</body>

</html>
```

Comments:

- JQuery is loaded from Google CDN
- All user events such clicking the submit button, are set after the web page is loaded
- Note how the DOM is updated using an html() function
- Overall, JavaScript code has become more compact

Which enhancement below would have the most impact on the user's pointing experience?

1. Hyperlink from INVESPAI's trademark to the home page
2. Bubble Cursor when entering/updating login credentials
3. Accelerating mouse pointer when selecting a portfolio from the navigation bar
4. Moving the navigation bar to the middle of the screen

Answer: bubble cursor has the benefit of better hitting a small target; particularly for users with motor impairment. Hyperlink has minimal impact and is not related to pointing. Placing the navigation bar in the middle of the screen increases motion time. Accelerating the mouse may increase errors when selecting a portfolio.

Example 8.2. Make changes to improve pointing experience in example 8.1. Store JavaScript code in a separate file and develop comprehensive unit tests using Jest.

8.3 How to Be Critical[3]

Once you have a design, you would like to evaluate it to determine if you are on the right path. In this course we will review three ways to know if the design is good:

1. Critique
2. Empiricism

[3] Amy J. Ko. Design Methods, Chapter 8; https://faculty.washington.edu/ajko/books/design-methods/critique

3. Analysis

We start with *critique*, which is a way to provide useful, constructive feedback. Critiques are not just evaluation of designs, but collaborative deconstructions of what makes a design successful and what makes it fail. Design critiques have a number of unique features to ensure the feedback is useful:

- Critiques are *two-way.*
    - A designer articulates the rationale for their decisions, and
    - The critic is responding to those judgements
- The critic must *engage deeply* in the substance of the problem
    - The more expertise they have about the problem, the better. Otherwise, they need to obtain full background from the designer to provide meaningful feedback.
- Critiques are both divergent and convergent
    - They can generate multiple ideas or point in different directions (divergent)
    - But the primary focus is to determine what doesn't work, and how to finalize the design (convergent)
- Follow the "*hamburger*" rule
    - Harsh negative feedback can cause the designer to become overly defensive, upset, and not receptive to the substance of the feedback
    - The hamburger rule: if you are going to say something sharply negative, say something genuinely positive first, and conclude with something genuinely positive as well.
    - This technique is often employed by coaches in youth sports such as soccer

Practice. Consider design of jQuery implementation in examples 8.1 / 8.2, and have a critique along the following lines:

- Why did you decide to use jQuery?
- What was the advantage of jQuery over plain JavaScript?
- What worked well?
- What didn't work and how to improve it?

Another form of critique is Socratic questioning. In this approach, the critic wants to deeply probe the designer's way of thinking and dig beneath the surface of their design. Some types of questions in Socratic approach are:

- *Clarification*
    - Encourage the designer to clarify their thought process
- Challenge the designer's *assumptions*
- Encourage the designer to consider *alternative perspectives*
- Encourage the designer to spell out *implications* and *consequences* of their design

In class 6 we studied an MVC architecture and worked out example 6.4 to design a portfolio view moving towards that architecture. As we critique this design, the following question is asked:

In your design where is the controller? Isn't it a key part of the MVC design pattern?

What type of question is this in the Socratic approach? (Choose the best answer)

1. Clarification
2. Determining implication
3. Challenging assumptions
4. Considering alternatives

Answer: clarification. While the question has a sharp undertone, it is still trying to understand the designer's thinking. Following questions may challenge if this is indeed an MVC design, if other design patterns should be considered, and what the implications are of sticking with MVC if there is no controller.

Another aspect of critiques is to judge what makes a design "good". We first list some common, but vague principles that are not very useful:

- *Intuitive*
    - This depends on the person's prior knowledge and may vary greatly from person to person
- *User-friendly*
    - This is not well defined as it may mean a lot of different things: nice, helpful, easy, etc.

There are many design principles in broad use that are a bit more precise:

- *Simple*
    - A design aesthetic that prizes minimalism and learnability
    - Drawback: may not reflect inherent complexity of a problem
    - Example: chat window may not be able to prevent hate speech
- *Novel*
    - In certain design cultures (e.g., fashion design), the best design is the new design that pushes boundaries
    - Drawback: trades off against simplicity as it may not be easy to learn
    - Example: Artificial Intelligence to generate code may be very complex to use

- *Powerful*
    - Values ability of designs to augment human ability
    - Drawback: may lead to complexity
    - Example: financial calculator
- *Invisible*
    - Values designs that "get out of the way"
    - Drawback: trades off against power and control
    - Example: Alexa intelligent assistant
- *Universal*
    - Something that all humanity should be able to access, prizing equality
    - Drawback: trades off against power and novelty
    - Example: website that is screen readable for vision impaired users
- *Just*
    - Purpose not to amplify inequalities and injustices in the world, but to dismantle them
    - Example: small business loan networking

What design principle is best represented by a Bubble cursor?

1. Invisible
2. Universal
3. Novel
4. Just

Answer: novel as compared to an existing mouse pointer. It is not invisible as the design is quite expressive. It does not try to address inequalities and injustices. It is also not universal as it mostly targets motor impaired users.

Example 8.3. UX Pilot is an AI tool that can be used to review a user interface design; see https://adamfard.com/blog/design-feedback-ai. Otter is an AI meeting assistant that claims to remove the need for taking meeting notes, and other functions; see https://otter.ai/. UX Pilot was used to review Otter's homepage as follows.

- Select UX Design Review from the UX Pilot dashboard
- Drag and drop the Otter's homepage image
- Enter the following text prompt to guide the AI:
    - Review the design based on these criteria:
        - *Clarity*: confirm that the text is clear and easy to understand
        - *Conciseness*: ensure brevity, avoiding unnecessary words or jargon
        - *Tone and Voice*: check if the page aligns with the brand's tone and voice

        - *Engagement*: assess whether the page is engaging and likely to keep user's interest
    - If you find any mistakes, please suggest improvements
- UX Pilot generated the following review:
    - Clarity
        - "Introducing Meeting GenAI" is clear but consider adding a brief explanation to ensure understanding.
        - ...
    - Conciseness
        - "Never take meeting notes again..." can be shortened to: "Stop taking notes manually. Get transcripts and summaries, and chat with Otter for meeting insights."
        - ...
    - Tone and Voice
        - The overall tone is informative and professional, aligning well with the brand's voice.
        - ...
    - Engagement
        - The phrase "Start for Free" is engaging, but consider a more action-oriented phrase like "Try It Free."
        - ...
    - UX and UI Improvements
        - Implement tooltips or info icons to explain features like "Otter AI Chat" and "AI Channels" for first-time users.
        - ...

8.4 jQuery Continued

In the next example we expand on the MVC architecture by storing user information on the server and processing it on the client.

Example 8.4. User credentials and portfolios are stored in a JSON format to be obtained from the server and processed using jQuery.

The following JSON is stored in 8_3.json:

```
{
  "user1": {
    "name": "John Doe",
    "password": "foo1",
    "portfolios": [
      "Portfolio1 (user1)"
    ]
  },
  "user2": {
    "name": "Jane Doe",
```

```
    "password": "foo2",
    "portfolios": [
      "Portfolio1 (user2)",
      "Portfolio2 (user2)"
    ]
  }
}
```

The following is HTML (should be run in a browser with the local http server).

```
<!DOCTYPE html>
<html lang="en">

<head>
    <title>INVESPO - Menu Prototype</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"><
/script>
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <script>
        var DB = {};

        function processLogin() {
            // Empty the list first
            $('#portfolioList').html("");
            $('#userFullName').html("");

            userData = DB[$('#username').val()];
            if (!userData) {
                alert("Incorrect user name!");
                return;
            }

            if (userData["password"] !== $('#psw').val()) {
                alert("Incorrect password!");
                return;
            }

            // If found, set up all available portfolios per user
            $.each(userData["portfolios"], function(index, value) {
                $('#portfolioList').html($('#portfolioList').html() + '<li>' + value
+ '</li>')
            });
            $('#userFullName').html(userData["name"]);
            $("#modalLogin").modal('hide');
        }

        $(document).ready(function() {
            $("#submitLogin").click(function() {
                processLogin();
            });

            $.getJSON("http://localhost:8080/8_3.json", function(data) {
                DB = data;
            });
        });
```

```
    </script>
</head>

<body>
    <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
        <div class="container-fluid">
            <a class="navbar-brand">INVESPO</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#collapsibleNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>
            <div class="collapse navbar-collapse" id="collapsibleNavbar">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="#">Customer Service</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Open Account</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" data-bs-toggle="modal"
data-bs-target="#modalLogin" href="#">Login</a>
                    </li>
                    <li class="nav-item dropdown">
                        <a class="nav-link dropdown-toggle" href="#" role="button"
data-bs-toggle="dropdown">Portfolios</a>
                        <ul class="dropdown-menu" id="portfolioList">
                        </ul>
                    </li>
                </ul>
                <span class="navbar-text" id="userFullName"></span>
            </div>
        </div>
    </nav>

    <div class="modal fade" id="modalLogin">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-body">
                    <button type="button" class="btn-close" data-bs-dismiss="modal"
aria-label="Close"></button>
                    <form>
                        <div class="my-3">
                            <input type="text" id="username" class="form-control" />
                            <label class="form-label"
for="username">Username</label>
                        </div>

                        <div class="my-3">
                            <input type="password" id="psw" class="form-control" />
                            <label class="form-label" for="psw">Password</label>
                        </div>
                        <button type="button" class="btn btn-primary my-3"
id="submitLogin">Submit</button>
                    </form>
                </div>
            </div>
        </div>
    </div>

</body>
```

```
</html>
```

Comments:

- Run a server:
  - `http-server ./`
- Open the above HTML is a browser window by specifying URL:
  - `http://localhost:8080/8_3.html`
- JSON containing all user data is retrieved via HTTP call
  - Note: it is retrieved after the page is loaded
- Login window is closed after a successful login
- User full name is updated after a successful login

Example 8.5. Critique the interface 8.3 (preferably in a group), and make at least one improvement.

8.5 HW2 Presentations

- Presentations:
  - Group 3:
    - Gitesh Ahirrao, Prathamesh Bachhav, Akhil Varier
    - Link: https://mason.gmu.edu/~avarier/
- Format
  - Time limit: 7 minutes
  - Order:
    - Design
      - Ensure to cover argument
    - Implementation
      - Demonstrate functional features
    - Interface metrics
      - Cover accessibility
    - Critique by class
  - All team members must take part in the presentation