Lecture 2: Visual representation
Overview of the course

• Theory driven approaches to HCI
• **Design of visual displays**
• Goal-oriented interaction
• Designing smart systems
• Designing efficient systems
• Designing meaningful systems (guest lecturer)
• Evaluating interactive system designs
• Designing complex systems
Why talk about visual design?

Visual design questions are often where discussions about HCI start

“What colour should this be?”
“Should it be 5 pixels or 7?”
“Will this work for colour blind people?”

These conversations are a lot more complicated than they seem...
The range of visual representations

TYPOGRAPHY AND TEXT
\[ i\hbar \frac{\partial}{\partial t} \Psi = -\frac{\hbar^2}{2m} \nabla^2 \Psi + V\Psi \]
### Database

#### Data

<table>
<thead>
<tr>
<th>Conversation Tags</th>
<th>Tag ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>messageTags</td>
<td>tag-12be1f863</td>
</tr>
<tr>
<td>nook_conversations</td>
<td>tag-14a8e13</td>
</tr>
<tr>
<td>sms_raw_message</td>
<td>tag-acc035760</td>
</tr>
<tr>
<td>suggestedReplies</td>
<td></td>
</tr>
<tr>
<td>systemMessages</td>
<td></td>
</tr>
<tr>
<td>tables</td>
<td></td>
</tr>
<tr>
<td>translation_cac...</td>
<td></td>
</tr>
</tbody>
</table>

Cloud Firestore location: eu3 (Europe-West)
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.

```
> lukes-mbp:~ lukechurch$ cd ~/GitRepos/Lark/KK-Project-2020-IOM/nook_sms_channel/
> lukes-mbp:nook_sms_channel lukechurch$ ls
  Pipfile
  Pipfile.lock
  firebase_sync_cli.py
  firebase_sync_cli_mock.py
  rapidpro_sms.py
  rapidpro_to_firestore_msg_map.py
  rapidpro_to_firestore_msg_subcol.py
  pagerduty_trigger.py
  rapidpro_to_firestore_msg_subcol.py
```

```
> lukes-mbp:nook_sms_channel lukechurch$ pipenv sync

Installing dependencies from Pipfile.lock (b5dd415)
An error occurred while installing -e git+https://www.github.com/AfricasVoices/RapidProTools@9a656080c4e0f5f32cc6a2e8f1b39a8104aa0#egg=rapidprotools! Will try again.

Installing initially failed dependencies...

To activate this project's virtualenv, run pipenv shell.
Alternatively, run a command inside the virtualenv with pipenv run.
All dependencies are now up-to-date!
```
MAPS AND GRAPHS
The EDSAC UI (1950s)
Exports and Imports to and from DENMARK & NORWAY from 1700 to 1780

BALANCE in FAVOUR of ENGLAND.

BALANCE AGAINST

The bottom line is divided into Years, the right hand line into L.10,000 each.

SAGE air defense (1960s)
Mappa Mundi (1300s)
## Bertin’s *Semiologie Graphique* (1969)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Position</th>
<th>Size</th>
<th>Value</th>
<th>Texture</th>
<th>Hue</th>
<th>Orientation</th>
<th>Shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point</td>
<td>+ + + +</td>
<td><img src="image1" alt="Size Diagram" /></td>
<td><img src="image2" alt="Value Diagram" /></td>
<td><img src="image3" alt="Texture Diagram" /></td>
<td><img src="image4" alt="Hue Diagram" /></td>
<td><img src="image5" alt="Orientation Diagram" /></td>
<td><img src="image6" alt="Shape Diagram" /></td>
</tr>
<tr>
<td>Line</td>
<td><img src="image7" alt="Line Diagram" /></td>
<td><img src="image1" alt="Size Diagram" /></td>
<td><img src="image2" alt="Value Diagram" /></td>
<td><img src="image3" alt="Texture Diagram" /></td>
<td><img src="image4" alt="Hue Diagram" /></td>
<td><img src="image5" alt="Orientation Diagram" /></td>
<td><img src="image6" alt="Shape Diagram" /></td>
</tr>
<tr>
<td>Area</td>
<td><img src="image8" alt="Area Diagram" /></td>
<td><img src="image1" alt="Size Diagram" /></td>
<td><img src="image2" alt="Value Diagram" /></td>
<td><img src="image3" alt="Texture Diagram" /></td>
<td><img src="image4" alt="Hue Diagram" /></td>
<td><img src="image5" alt="Orientation Diagram" /></td>
<td><img src="image6" alt="Shape Diagram" /></td>
</tr>
</tbody>
</table>

**Figure 1.** The visual variables and their effectiveness in signifying the three levels of measurement of data (after Bertin [1983]).
SCHEMATIC DRAWINGS
Sutherland’s Sketchpad (1963)
The London Underground Map (1920)
Henry Beck’s London Underground Diagram (1931)
ICONS AND SYMBOLS
VISUAL METAPHOR
XEROX 8010 Star Information System

Star provides integrated text and graphics. A variety of type styles and sizes may be used.

<table>
<thead>
<tr>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>$0.20</td>
</tr>
<tr>
<td>Table</td>
<td>$0.50</td>
</tr>
</tbody>
</table>

This is some text in a text frame.
- Form field
- Button
PICTURES
Ways of Seeing

John Berger

Seeing comes before words. The child looks and recognizes before it can speak.

But there is also another sense in which seeing comes before words. It is seeing which establishes our place in the surrounding world; we explain that world with words, but words can never undo the fact that we are surrounded by it. The relation between what we see and what we know is never settled.

The Surrealist painter Magritte commented on this always-present gap between words and seeing in a painting called The Key of Dreams.

The way we see things is affected by what we
Theories of visual design
<table>
<thead>
<tr>
<th>Graphic Resources</th>
<th>Correspondence</th>
<th>Design Uses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shape</td>
<td>Literal (visual imitation of physical features)</td>
<td>Mark position, identify category (shape, texture colour)</td>
</tr>
<tr>
<td>Orientation</td>
<td>Mapping (quantity, relative scale)</td>
<td>Indicate direction (orientation, line)</td>
</tr>
<tr>
<td>Size</td>
<td>Conventional (arbitrary)</td>
<td>Express magnitude (saturation, size, length)</td>
</tr>
<tr>
<td>Texture</td>
<td>Simple symbols and colour codes</td>
<td></td>
</tr>
<tr>
<td>Saturation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Colour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Line</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Topological (linking)</td>
<td>Texts and symbolic calculi</td>
</tr>
<tr>
<td></td>
<td>Depictive (pictorial conventions)</td>
<td>Diagram elements</td>
</tr>
<tr>
<td></td>
<td>Figurative (metonym, visual puns)</td>
<td>Branding</td>
</tr>
<tr>
<td></td>
<td>Connotative (professional and cultural association)</td>
<td>Visual rhetoric</td>
</tr>
<tr>
<td></td>
<td>Acquired (specialist literacies)</td>
<td>Definition of regions</td>
</tr>
<tr>
<td>Symbols</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Geometric elements</td>
<td>Containment</td>
<td></td>
</tr>
<tr>
<td>Letter forms</td>
<td>Separation</td>
<td></td>
</tr>
<tr>
<td>Logos and icons</td>
<td>Framing (composition, photography)</td>
<td>Identifying shared membership</td>
</tr>
<tr>
<td>Picture elements</td>
<td>Layering</td>
<td>Segregating or nesting multiple surface conventions in panels</td>
</tr>
<tr>
<td>Connective elements</td>
<td></td>
<td>Accommodating labels, captions or legends</td>
</tr>
<tr>
<td>Regions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alignment grids</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Borders and frames</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area fills</td>
<td></td>
<td></td>
</tr>
<tr>
<td>White space</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gestalt integration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Surfaces</td>
<td></td>
<td></td>
</tr>
<tr>
<td>The plane</td>
<td>Literal (map)</td>
<td>Typographic layouts</td>
</tr>
<tr>
<td>Material object on which the marks are imposed (paper, stone)</td>
<td>Euclidean (scale and angle)</td>
<td>Graphs and charts</td>
</tr>
<tr>
<td>Mounting, orientation and display context</td>
<td>Metrical (quantitative axes)</td>
<td>Relational diagrams</td>
</tr>
<tr>
<td>Display medium</td>
<td>Juxtaposed or ordered (regions, catalogues)</td>
<td>Visual interfaces</td>
</tr>
<tr>
<td></td>
<td>Image-schematic</td>
<td>Secondary notations</td>
</tr>
<tr>
<td></td>
<td>Embodied/situated</td>
<td>Signs and displays</td>
</tr>
</tbody>
</table>
Use 1: Usability analysis

- Analyse a design using this taxonomy
  (e.g. Uses connotative correspondence)

- Determine if the uses are appropriate
  (e.g. do people think that blue is ‘off’ or ‘cold’?)

Cases where the visual analysis implies one connection but the operation implies another will create substantial usability problems
Use 1: Usability analysis
Use 1: Usability analysis

Mapping, Categorical: Off, cold, warm, hot

Mapping, Magnitude, Colder -> Warmer
Use 1: Usability analysis

Mapping, Categorical: Off, cold, warm, hot

Mapping, Magnitude, Colder → Warmer
Mapping, Direction(!!!), Colder -> Warmer
Use 2: Generating divergent designs

- Topological linking
  =>

- Arbitrary conventional

Cone.ByPointsRadii

| startRadius | 2.5 |
| endRadius | 7 |

- startRadius (🐔) = 2.5
- endRadius (🦔) = 2.5
Use 3: Design systems

Color usage
Color helps express hierarchy, establish brand presence, give meaning, and indicate element states.

(https://material.io)
Analysing visual representation in a UI

Look more carefully at familiar things …
Analysing visual representation in a UI

Shaded regions define separate functional contexts

White foreground region for user’s “WYSIWYG” content

Shaded region groups all controls
Analysing visual representation in a UI

Coordinate space for quantitative calculation of grid layout
Analysing visual representation in a UI

Blue coloured border:
1. Corporate identity (cf IBM)
2. Context within Office suite - Excel docs have green borders, PPT are orange.
Analysing visual representation in a UI

How do we read these?

Difficult standards:
When the designer’s mental model differs from the user’s mental model

= ?
The programming analogy challenge 2024:
Example #1: ChatGPT (actually Google Bard)

what is the html code to make an image no more than the width of the browser page?

I can't create images yet so I'm not able to help you with that.
Example #1: ChatGPT (actually Google Bard)

Here are two ways to set the width of an image to the full width of the page in HTML:

1. Using the `width` attribute:

```html
<img src="your-image.jpg" width="100%" alt="Image description">
```

Use code with caution. Learn more.
Example #1: ChatGPT (actually Google Bard)

2. Using CSS:

```
<html>
  <head>
    <style>
      .full-width { width: 100%; }
    </style>
  </head>
  <body>
    <img src="your-image.jpg" class="full-width" alt="Image description">
  </body>
</html>
```
Example #1: ChatGPT (actually Google Bard)

- Use responsive image techniques to serve different image sizes for different screen sizes.

  - **Responsiveness:** For a truly responsive solution that adapts to different screen sizes, consider using CSS media queries to adjust the image's width based on the viewport width.

**Additional tips:**

- **Image optimization:** Optimize your images for web use to reduce file size and improve loading times.

- **Alternative text:** Always provide meaningful alternative text for images to improve accessibility and SEO.

**Best practices recommend using CSS for styling elements, including image width.** The `width` attribute can still be used as a fallback for older browsers or when inline styling is necessary.
Example #1: ChatGPT (actually Google Bard)

How can I tell my Alexa to close the curtains automatically when it gets dark?

There are a few ways to tell your Alexa to close the curtains automatically when it gets dark, depending on the specific smart home setup you have and the capabilities of your curtains. Here are a few options:

Using Routines:

1. Alexa app: Open the Alexa app on your smartphone or tablet. Tap on "Routines" and then "Create Routine." Choose "When this happens" and select "Smart Home" followed by "Schedule" or "Sunset." Set the desired time or choose "Sunset" as the trigger. Then, choose "Add action" and select "Smart Home" again. Choose your curtains and select "Close." Save the routine and name it something like "Close Curtains at Sunset."

2. Voice command: Once you have created the routine, you can simply say "Alexa, activate Close Curtains at Sunset" to trigger it.
Summary: ChatGPT (or Gen AI) as programming language

If we were designing a language for instructing computers what to do, should it look like English?

The AI alignment problem - how do we know whether a computer will do what we really want, rather than follow its own ideas?

If you’re a computer scientist rather than a philosopher, these problems look very familiar!

   a) specification & coding
   b) testing & debugging