FIT3175 - Usability

Interaction Styles and Devices

Week 5 Lecture P2

MONASH University

Learning objectives

Interaction styles

- Interaction types and styles
- Implementation considerations

Interaction devices

- Input and output
- Input devices
- Direct and indirect pointing

Modelling user interaction performance

- Fitts' Law
- GOMS - KLM and variants

Student Evaluation of Teaching and Units

Interaction Styles

How does interaction work?

Different aspects of interaction that we are familiar with:

Perceptual

- The aspects of interaction that are detectable using human senses.

Conceptual

- Understanding a system using perceptible information.
- Understanding a system using mental models and problem-solving.

Physical

- Ability to perform required and understood actions

The implementation of the interface mediates the types of interactions performed.

Shneiderman's interaction types

Interaction **types** define the aim of an interaction

• **Instructing**	Issuing individual specific commands.
• **Conversing**	Having a dialogue - an exchange of information.
• **Manipulating**	Modifying representations of objects.
• **Exploring**	Finding information my moving through a space.

These types refer to the human approach to the interaction, not the specific of the interface design.

Shneiderman's interaction styles

The interaction **styles** are a mechanism that allows interactions to occur:

- **Command language** Instructions manually specified with a required syntax.
- **Natural language** Imprecise instructions that are interpreted by the system.
- **Menu selection** User-selection from a range of system-defined options.
- **Form fill-in** User provides parameters to fit a supplied context.
- **Direct manipulation** Modification of objects with tightly-mapped feedback.

Complex systems often combine different styles to optimise task performance.

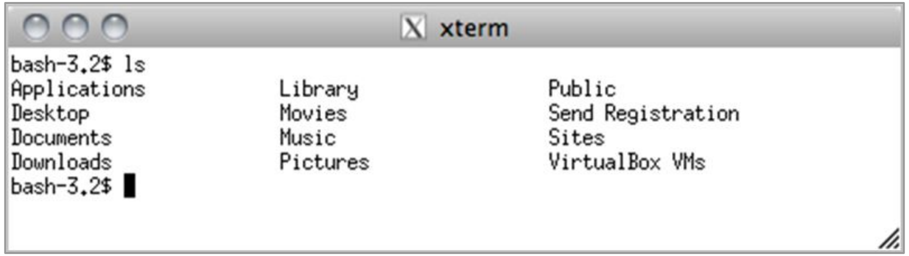
Interaction types and style compared

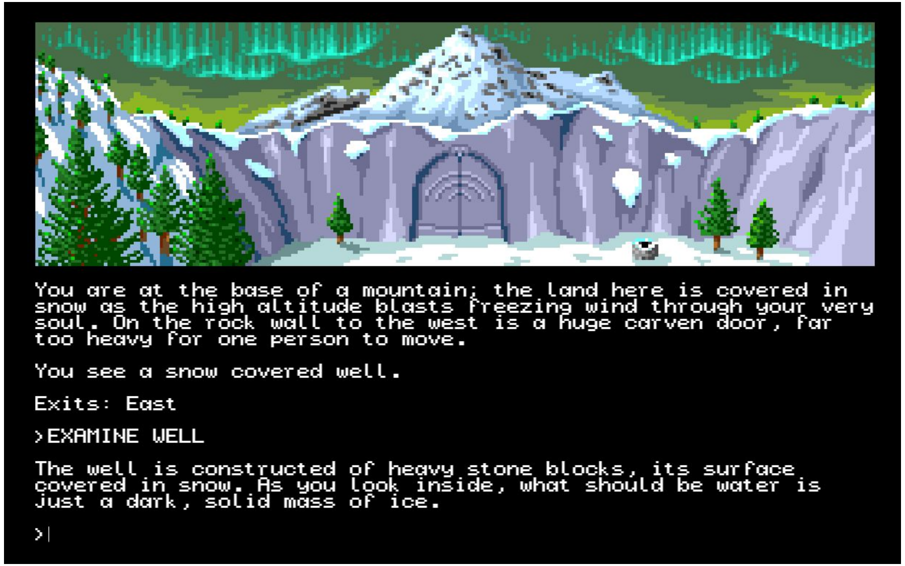
The interaction types may ***imply*** the use of a certain interaction style, but these are not necessarily exclusive relationships.

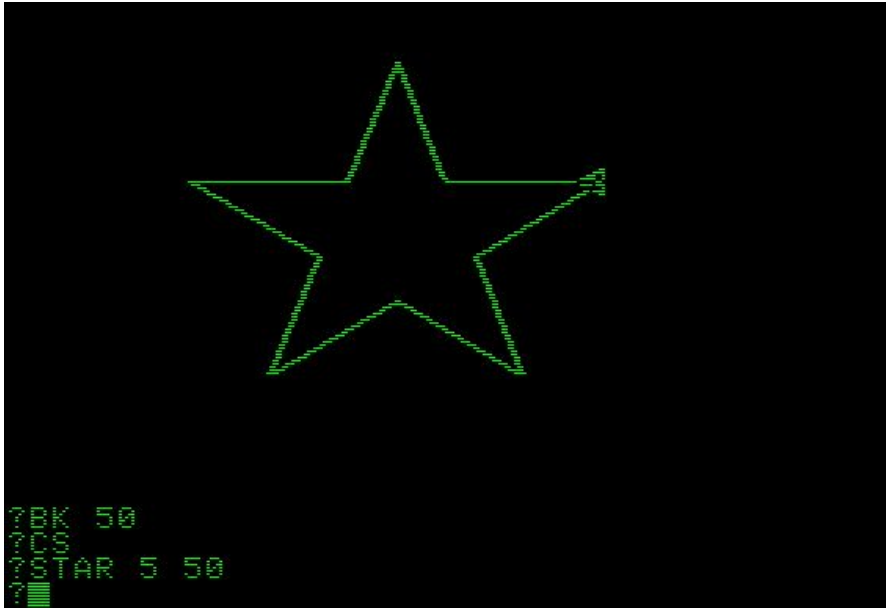
Interaction Types	Interaction Styles
Instructing	Command language
Conversing	Natural language
Manipulating	Menu selection
Exploring	Form fill-in
	Direct manipulation

Can you describe how a certain type could be implemented in multiple styles?

Command language







Most commonly associated with technical tasks in a terminal. However command language style can be used for less serious tasks.

Command language pros and cons

Advantages:

- Flexibility and speed for experts
  - **Ctrl+S** versus **File > Save**
  - **mv *.doc *.txt** versus manually renaming 100 files
- Provides a sense of control for expert users.
  - Issuing a complex command that with high precision.
  - Creating macros that automate a sequence of tasks.
- Low resource usage.

Disadvantages:

- Training is necessary.
  - Beginners might be overwhelmed!
- Requires recall of exact syntax.
  - Wrong syntax – **errors!**
- Low visibility/feedback of status.
  - What commands are available?
  - **mv file1 dir2** - how do we know if this command is right?
- Limited error prevention handling.

Natural language

The advent of always-on internet computing has brought natural language processing to a range of different device and applications.

Natural language processing

Commands given in "natural" phrases rather than set computer commands. These interactions can range from...

Simple Voice controlled menu systems (e.g. telephone voice menus)

- Closer to command language and menu selection systems

to... Virtual agents/avatars with basic language processing

- Website chat support systems

complex Handling complex natural language dialogues

- IBM's Watson playing Jeopardy
- Smart assistants handling continued conversations

Natural language is familiar and comfortable

Experts tell us:

- The user does not need to be trained in how to use the interface
- They allow more flexibility in executing steps of a task.
- They are faster than menu systems or composing queries.

https://www.youtube.com/watch?v=YVbMciuhA8E

Natural language pros and cons

Many natural language advantages have limitations that result in disadvantages.

Advantages

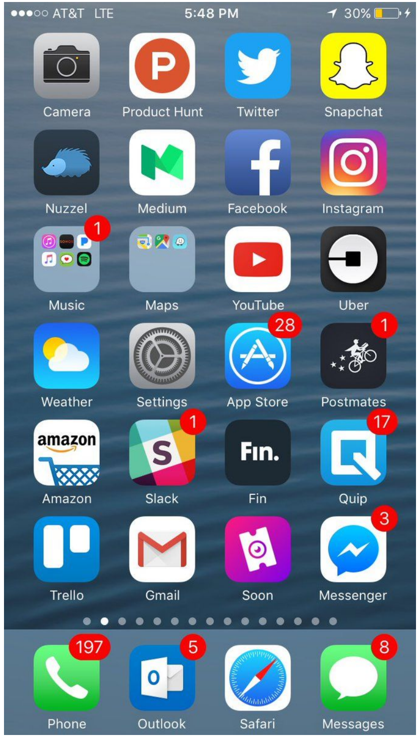
- No need to remember specific commands.
- Flexible syntax for novice users.
- Reduced learning curve for interface.
- Natural output of feedback/errors.

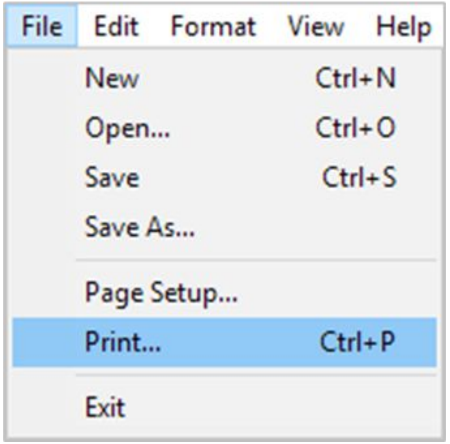
Disadvantages

- Low visibility of the available commands.
- Probability of syntax parsing errors.
- Low discoverability of interaction method.
- Generic errors are not very helpful.

Based on your experience - do current natural language systems work well?

Menu selection





Any method of selection from predefined actions that does not rely on users recalling commands and syntax is a menu selection.

This is true regardless of the input method used.

Can you identify command language style that is also implemented in any of these examples?

Menu selection pros and cons

Compared to other interactions styles:

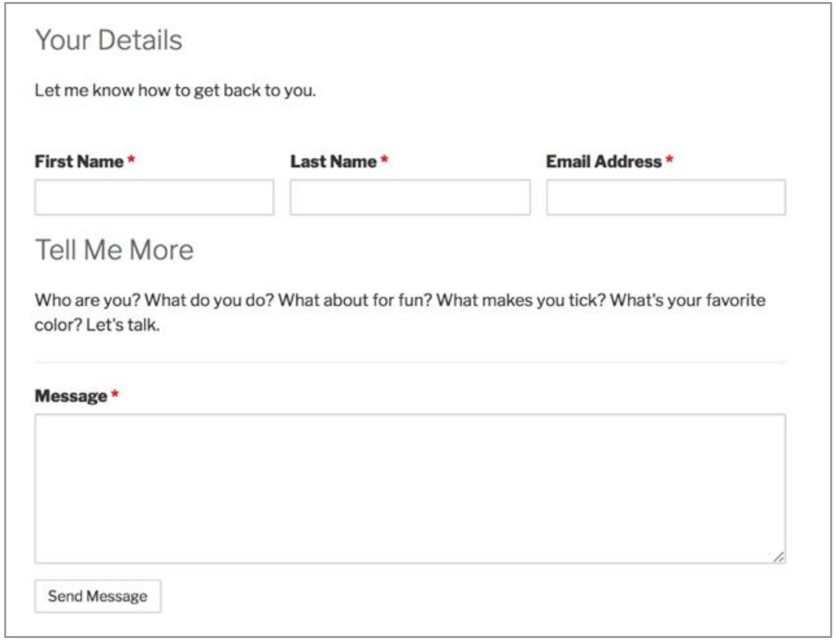
Advantages

- Easy to use and learn as it offers memory cues.
  - High visibility, recognition over recall.
- Assists users to structure decision making.
  - Fixed list of options implements constraint to reduce possible errors.

Disadvantages

- Organisation design may be difficult and time-consuming
  - Labelling design can be difficult for some actions/concepts
- May require very large numbers of menus/items to cover all choices
  - Menus may be slow, especially for frequent actions
  - Requires more screen space

Form fill-in



Form-fill-in interactions don't have to be boring and tedious. Well-defined structure, labels and appropriately conversational tone provide a more pleasant experience.

Form fill-in

This interaction style was originally developed for non-expert users. The system simply guides a user through providing parameters.

Advantages

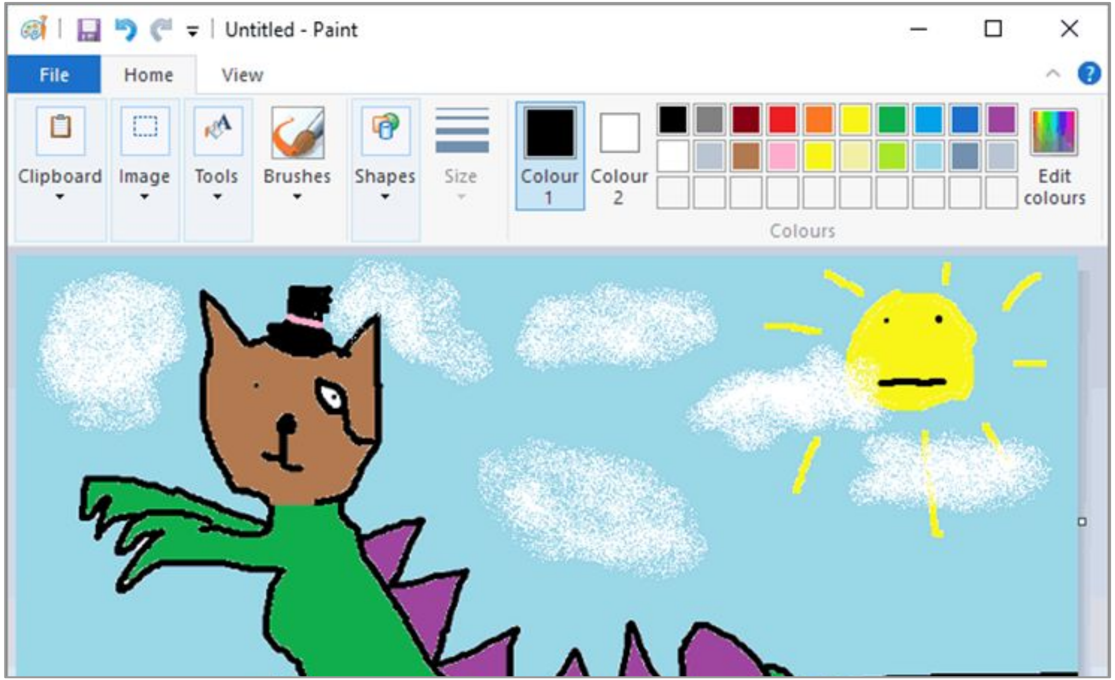
- Simplifies data entry (via hierarchy, structure)
- Easy to learn - guides users via predefined rules

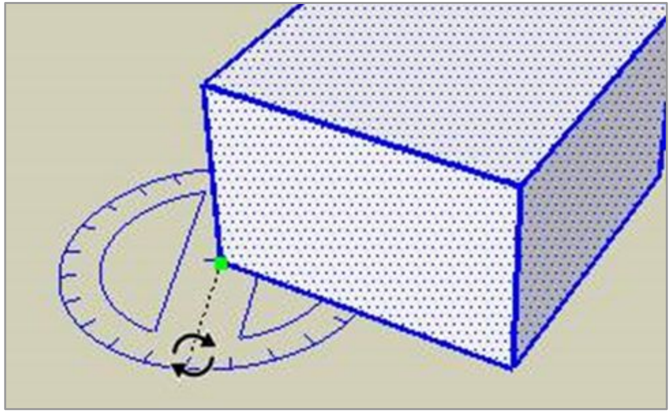
Disadvantages

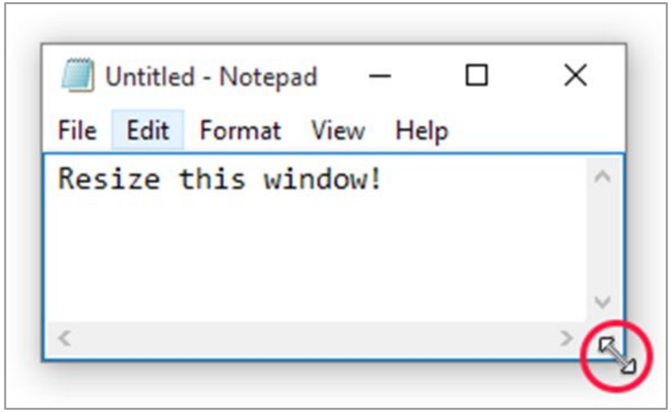
- Consume screen space
- Rigid process (is it enjoyable to fill in a form?)
- Single-task focus

The interaction in a form fill-in style **is** the entry of data. Typically there isn't a choice of tasks to execute. However, useful for single tasks that require many parameters.

Direct manipulation







Real-time manipulation with explicit feedback that incorporating strong mappings makes cause and effect easy to understand and creates engaging user experience.

Direct manipulation

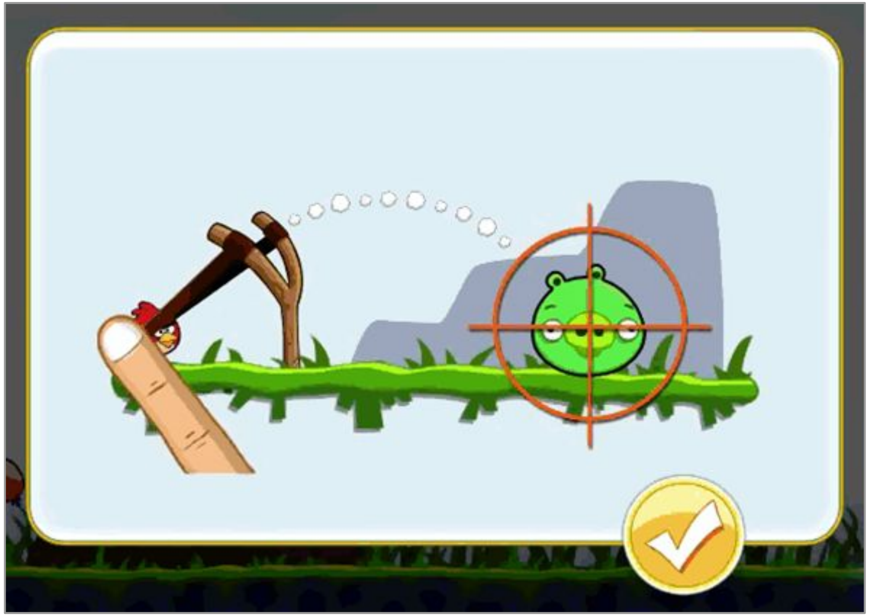
This interaction style provides visible and continuous representation of task objects and the actions used to modify them.

Task objects manipulated by physical actions require:

- Accurate visual representation of objects being interacted with.
- Incremental and reversible interactions.
- Close **spatial mapping** of task domain and interface domain helps user to focus on tasks rather than interface.
- Close **temporal mapping** of interaction and visual feedback.

Metaphors are often used to link manipulation actions to similar real-world actions.

Direct manipulation in touch interfaces



Touch interfaces allow visual representation of and feedback of an action to be perfectly mapped to a user's physical movements. As a result, objects and our effect on them feel more real.

Direct manipulation pros and cons

Advantages

- Intuitive, easy to learn and remember.
- Reduces errors as minimal syntax required.
- Reduces errors by allowing immediate reversals
- Enjoyable and encourages exploration by immediate feedback and evaluation.

Disadvantages

- More difficult to program.
- High system resource requirements.
- May require more screen space.
- Complex actions require more user dexterity to execute.
- Manipulations may be slower than typed input for the same accuracy.

Interaction Devices

Input and Output

We cannot separate the design of interfaces from the devices that allow them to be controlled and perceived. **The hardware and its design is part of the interface.**

The majority of current interfaces are designed to be operated with:

- **Switch and key device** - e.g. keyboard
- **Pointing input device** - e.g. mouse
- **Video output device** - e.g. monitor
- **Audio output device** - e.g. speakers

Many other pieces of hardware exist to support a wider range of interactions:

- **Input:** Cameras, microphones, sensors (motion, position, orientation, etc.)
- **Output:** Lights (status LEDs), haptics (vibrations), printers

Early computer I/O

Design goals and interaction devices

Whether designing completely new methods of interaction, or choosing which existing devices to support in your apps - **overall interface design goals remain the same...**

1. Reduce **learning** time
2. Increase **performance** speed
3. Reduce **error** rates of users
4. Support **retention** over time
5. Provide subjective **satisfaction**

Can you think of a 2 mostly interchangeable input devices that differ in usability?

Doug Engelbart and "the mother of all demos"

On December 9, 1968, Doug Engelbart and the Stanford Research Institute publicly demonstrated several new computing concepts that would shape the future of human-computer interaction.

The demonstration would later be known as the **"mother of all demos"**.

https://www.youtube.com/watch?v=B6rKUf9DWRI

Keyboards and keypads

Consider and Describe the usability impact of:

Microsoft Ergonomic Keyboard

- To provide support and comfort
- To prevent or ease repetitive strain injury (RSI)

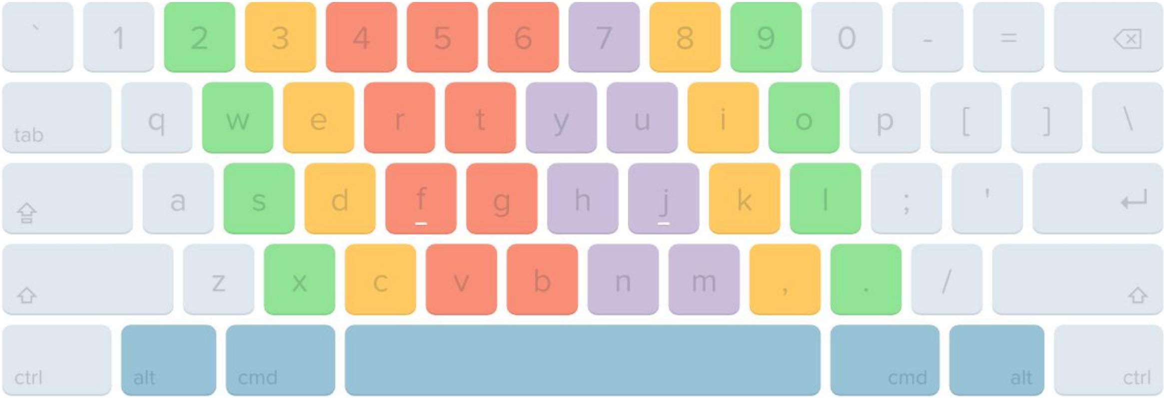
A2Z Alphabetic Keyboard

- Also supports traditional QWERTY layout.

HORI Tactical Assault Commander

- Keyboard input device for Playstation 4.
- Used with a mouse input controller.

Keyboard layouts



QWERTY

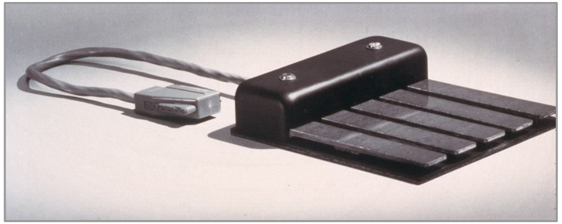
- Originally designed for typewriters
- Intended to reduce clashing typewriter letter arms? (unverified)

DVORAK

- Designed to reduce finger movement.
- Claimed to reduce error, increase speed and improve typing comfort.

https://www.typingpal.com/

Faster typing with chorded input



Doug Engelbart's Keyset (1967)

- Designed for one handed use
- Different letters could be inputted by pressing different multi-finger chords

Stenographer keyboards

- Fast shorthand typing for courtroom stenographers.
- Chorded input combined with intelligent software.

CharaChorder

- *"Type at the speed of thought™"*
- *Each finger also has multi-directional input.*

Virtual keyboards

Also known as **software keyboards**, these are common on touchscreen devices, but also appear as an option on computers.

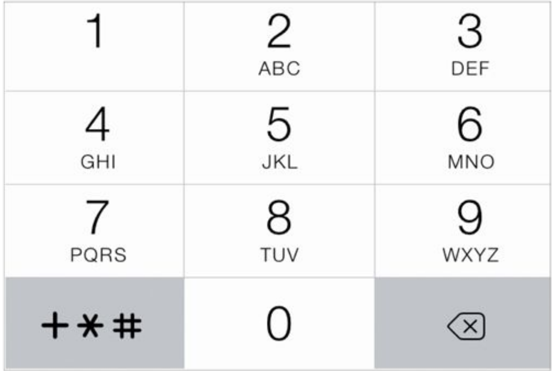
What are some of the benefits of a software keyboard?

Can you think of any disadvantages?

Default

Email

URL



Phone

https://developer.apple.com/library/archive/documentation/StringsTextFonts/Conceptual/TextAndWebiPhoneOS/KeyboardManagement/KeyboardManagement.html

Direct and indirect pointing devices

Pointing devices control the movement of a cursor or interactive object on the screen to allow spatial movement and selection.

Direct Pointing (absolute movement)

- Touchscreen with finger
- Touchscreen with stylus
- Graphics tablet with screen

Indirect Pointing (relative movement)

- Mouse and trackball
- Trackpad / touchpad
- Graphics tablet without screen





Indirect pointing devices require *translation* of user movement to screen space.

Mice

Invented by Doug Engelbart in 1964, used for **indirect pointing**, clicking and dragging. Can have a variety of features.

- Additional buttons
- Scroll wheels
- Optical and laser sensors
- Touch buttons and surfaces
- Wireless connectivity
- RGB (**?**)

Discuss usability impact of these features.

Trackpads / touchpads

A common alternative to a mouse for **indirect pointing**, especially when used in self-contained devices.

- May support multi-touch gestures.
  - May support tap-to-click.
  - May have external buttons.
- Lower accuracy for movements
  - Large movement performance limited by touchpad edge.
  - Accuracy improves for smaller movements.



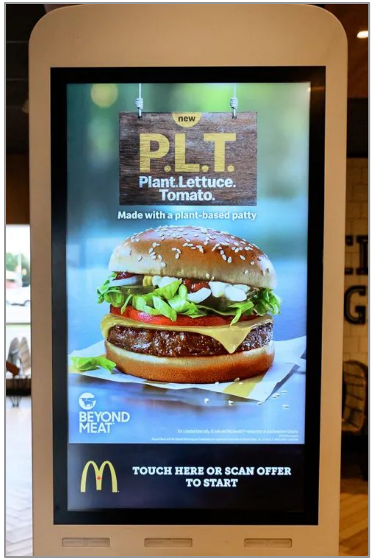


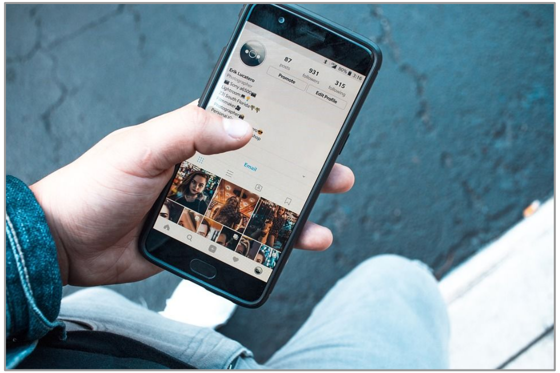

Touchscreens

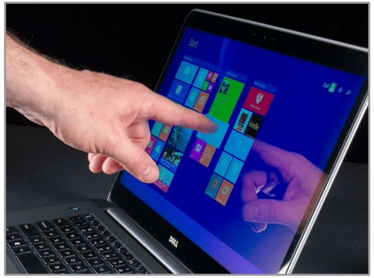
Used in device large and small, most often when external input devices are not convenient.

Touchscreens provide **direct pointing**.

- Lower precision than a mouse
- Multi-touch and gesture inputs
- No buttons for different actions
- No visible cursor or hover
- Minimal-to-no learning time
- Screen may be obscured on input







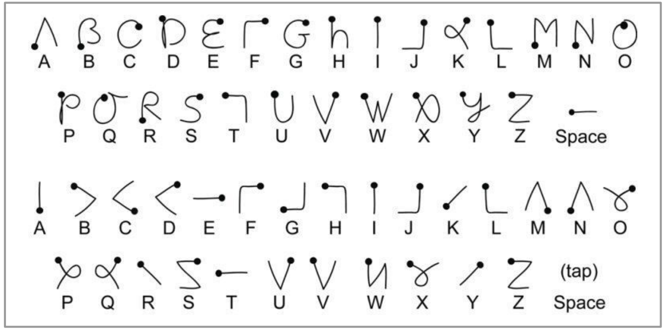
Handwriting input

Automatic recognition of handwritten input is still not optimal, but has improved greatly in recent years.

- Uses pattern recognition to match sections of writing to different character-shape models.
- Quality of recognition depends on
  - **Vocabulary** Pre-trained character shapes.
  - **Writing style** Recognition of specific styles.
  - **Trainability** Learn user's specific style.

Right-top: *Handwriting detection errors were often the butt of jokes.*

Right-bottom: *PalmPilot devices required users to learn "Graffiti".*



Audio and video input devices

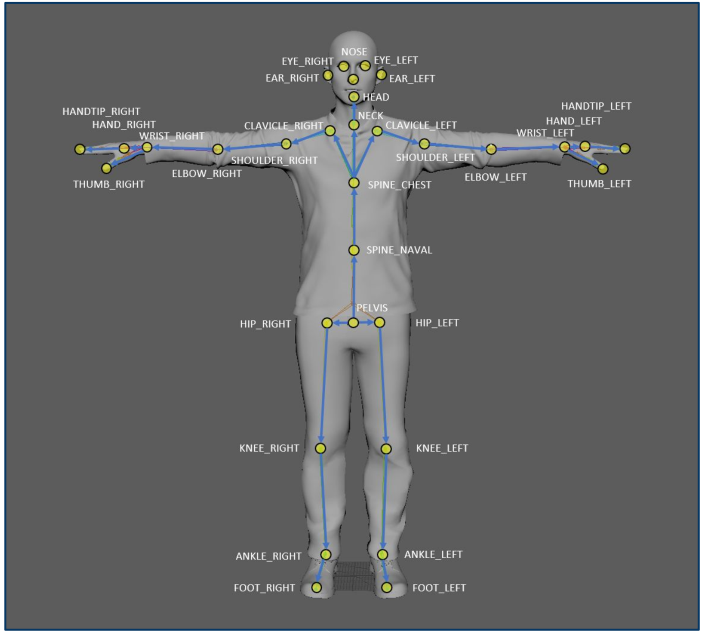
Camera-based tracking

- A camera recognises eye, head, hand or body movement.
- Gesture/pose recognition to for spatial pointing and command execution.

Microphone audio input

- Natural speech with reduced physical effort.
- Can be used remotely.

However, quality of inputs and results are highly dependent on data signal and processing.



Composite devices

Composite devices combine a variety of input types into a single device.

Consider the PlayStation Dualshock 4:

- Digital buttons
- Analog triggers
- Analog joysticks
- Digital gamepad
- Touchpad
- Headset/microphone input

Yes or no: a mouse is a composite device…

Modelling User Interaction Performance

Why model interactions?

During the design process, it can be difficult to know how well an interface performs.

Low-fidelity prototypes

- Tasks cannot be tested in a realistic way using actual interaction devices.

High-fidelity prototypes

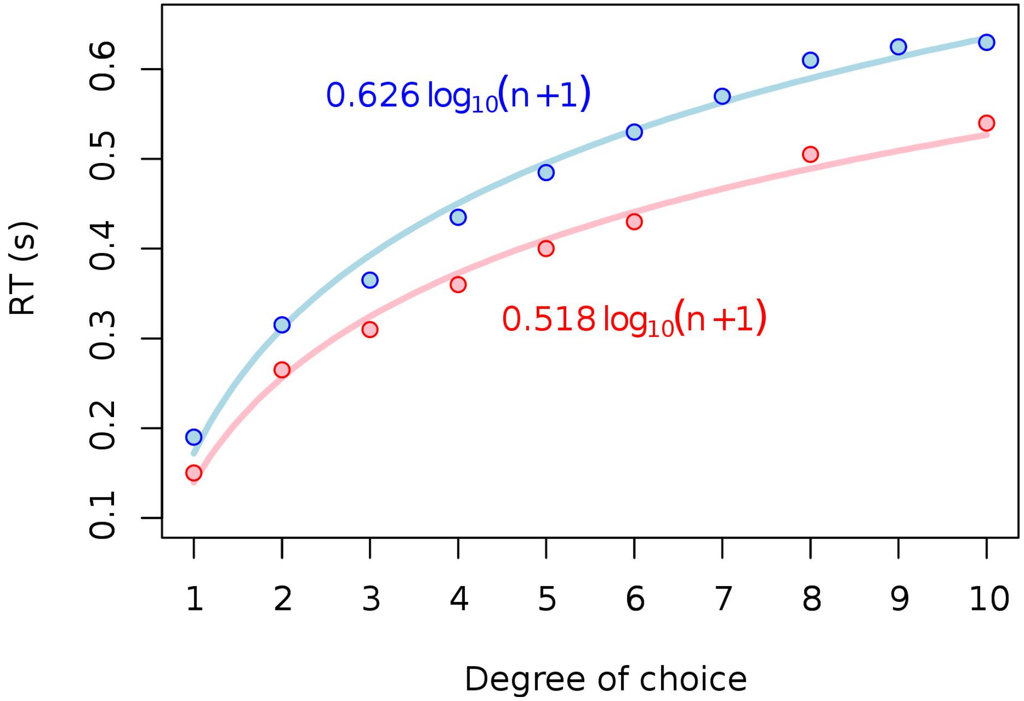
- Formal usability testing may be isolated to specific scenarios
- Some parts of the prototype may be incomplete.

Being able to create detailed models of task interactions allow performance of tasks to be estimated, predicted and equally compared.

Recap: Hick-Hyman Law (Hick's Law)

Research by psychologists William Edmund Hick and Ray Hyman, describes logarithmic increase in reaction time as the number of possible choices increases.

Applying Hick's Law we can make better predictions about the cognitive aspect of interaction times in a proposed menu design.



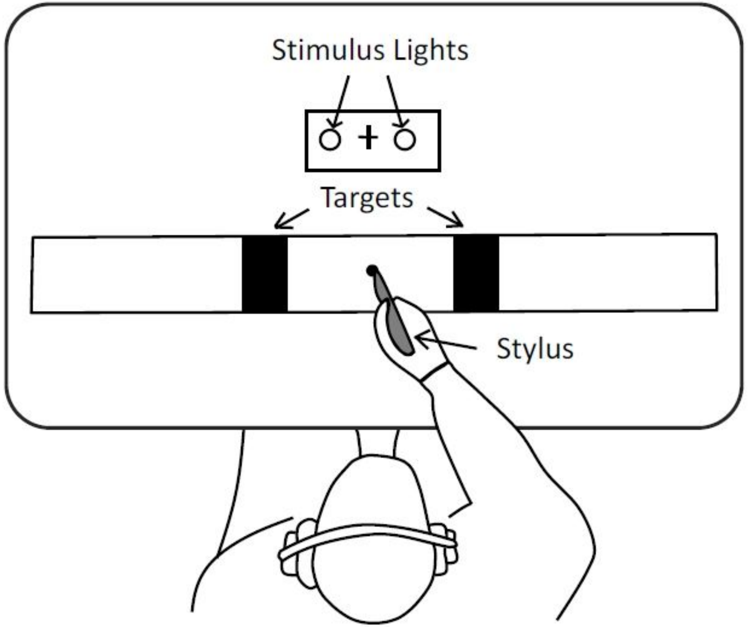
What Hick's Law doesn't model is the physical performance of interaction methods.

Fitts' Law

Fitts' Law provides a model of human movement that **predicts movement time** associated with pointing devices.

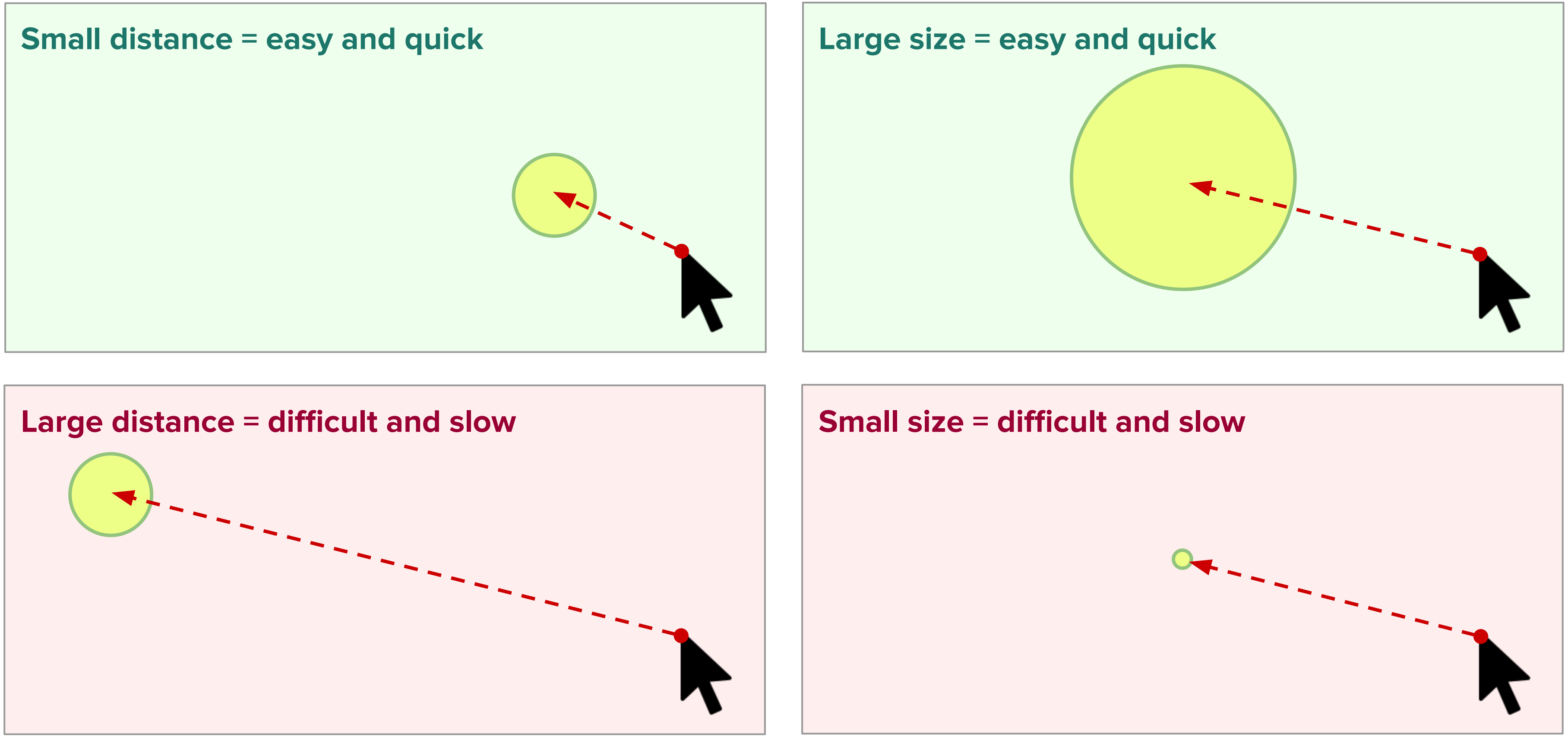
$$\mathrm{MT} = \boxed{a + b} \cdot \boxed{\mathrm{ID}} = \boxed{a + b} \cdot \boxed{\log_2\left(\frac{2D}{W}\right)}$$

- **MT** = movement time
- **a + b** = device delay + acceleration
- **ID** = index of difficulty
- **D** = distance to target center
- **W** = size (width) of target

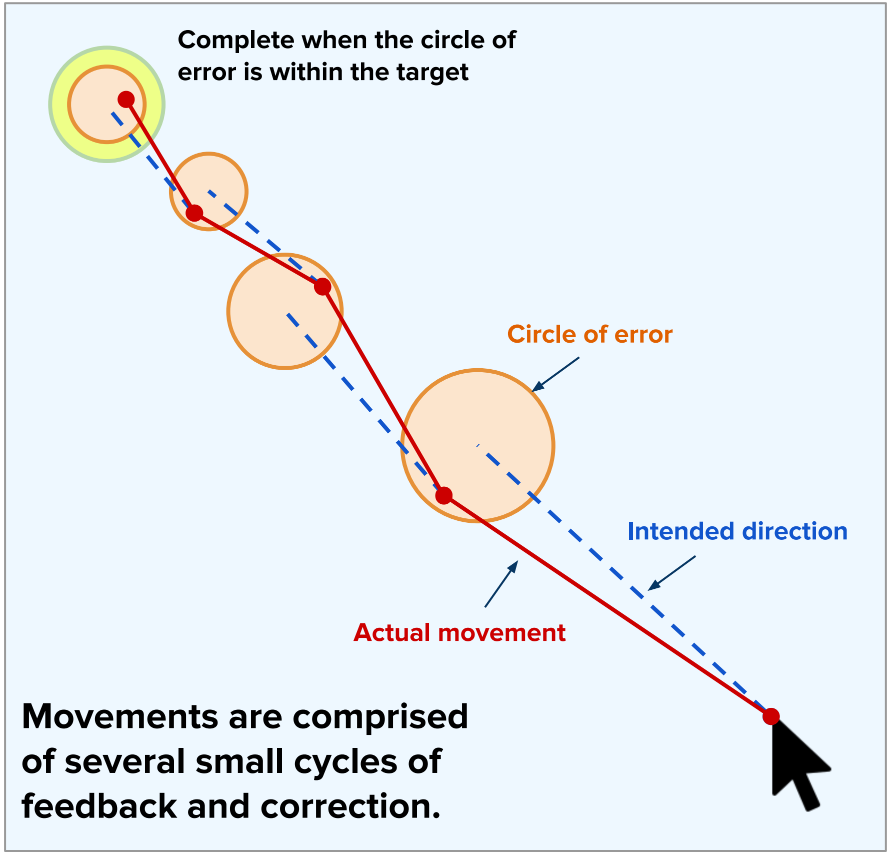


Above: *A Fitts' Law experiment. The user moves a pointer between targets that vary by width and distance.*

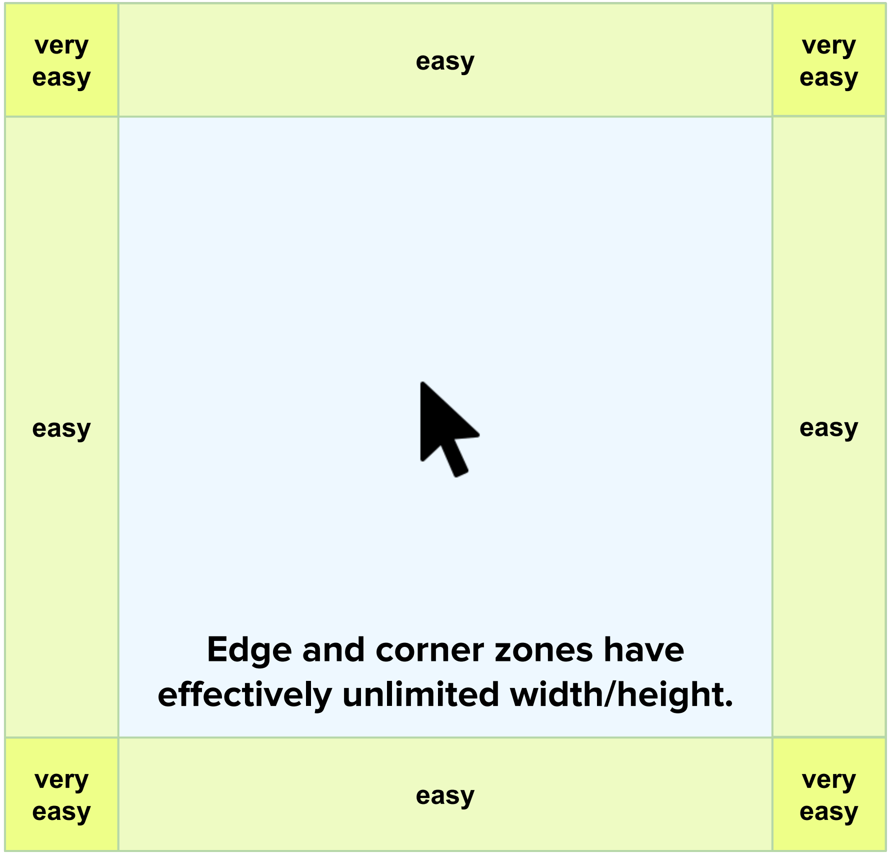
Fitts' Law "index of difficulty" factors



Further considerations for Fitts' Law



Movements are comprised of several small cycles of feedback and correction.

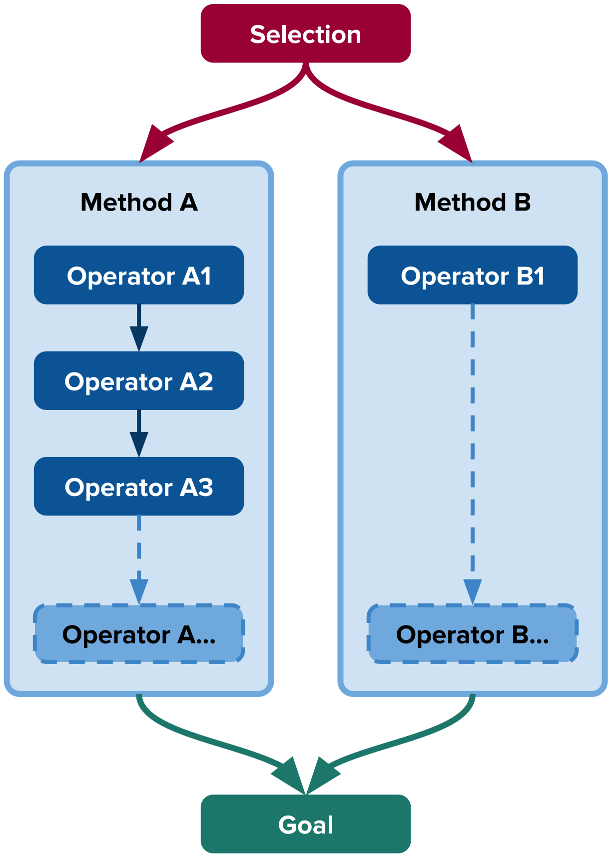


Edge and corner zones have effectively unlimited width/height.

GOMS modelling

To model a complex goal involving many diffent interactions, we can decompose it into basic actions:

- **Goals** The outcome to be achieved.
- **Operators** The different possible perceptual, physical, cognitive actions.
- **Methods** Variations of different actions that can be used to achieve the goal.
- **Selection** Rules for deciding which method to use in a given circumstance.



GOMS Keystroke-Level Model (KLM)

A discount usability application of GOMS is the **Keystroke-Level Model (KLM)**.

1. With a goal in mind, select a particular sequence of actions to measure.
2. Perform a cognitive walkthrough of an interface.
3. Count every minor operation required to perform each action.
4. Estimate the time for each type of action and multiply by the number of actions.

An example of estimated action timings:

- **H** **Home** Move hand(s) to input device **~0.4 sec**
- **K** **Keystroke** Keypress or mouse click **~0.3 sec**
- **M** **Mental** Perceive or think about task **~1.2 sec**
- **P** **Pointer** Move pointer to UI element **~0.4 sec**

GOMS KLM: Replacing a word

Description	Operation	Time (seconds)
Reach for mouse	H	0.4
Move pointer to "replace" button	P	0.4
Click "replace" button	K	0.3
Return to keyboard	H	0.4
Think of word to replace	M	1.2
Enter word to be replaced	K × 6	1.8
Press TAB to focus text input field	K	0.3
Return to keyboard	H	0.4
Think of new replacement word	M	1.2
Type new replacement word	K × 5	1.5
Reach for mouse	H	0.4
Move pointer to "replace all" button	P	0.4
Click on "replace all" button	K	0.3
	Total	**9.0 seconds**

In this example a relatively simple process is broken down component actions.

- Method selection should be representative of user workflow.
- Every action must be considered.
- The times for each action use previously estimated values.
- Allows different methods of achieving the same goal to be directly compared.

Limitations of GOMS

GOMS KLM is an expert prediction - **not a replacement for user testing**.

- **Only applicable for goal-directed tasks**
  - Does this method work for tasks that involve casual browsing?
- **Assumes equal user skill levels and performance**
  - Estimates typically for expert users, but can be adjusted.
- **Does not address UI issues that may slow users down**
  - This also varies by user skill level.
- **Does not consider performance differences of different input methods**
  - Model can be extended with different timings for different interactions.
- **Does not consider variability of waiting times for system processing**
  - Often cited as "negligible" assuming that modern computers are "fast"

Perception of delay

Waiting times not only impact efficiency - they can negatively affect a user's perception of a system's behaviour.

- **0.1 second** The user feels that the system is reacting instantaneously.
- **1.0 second** Approximate limit for uninterrupted flow of thought.
  Between 0.1s and 1.0s, the user will notice the delay.
  The user loses the feeling of operating directly on the data.
  Over 1.0s, some additional feedback should be provided.
- **10 seconds** Approximate limit for keeping the attention focused dialogue.
  For longer delays, users want to perform other tasks while waiting.
  Provide additional feedback to provide expectation of wait time.

https://www.nngroup.com/articles/response-times-3-important-limits/

Waiting time is torture

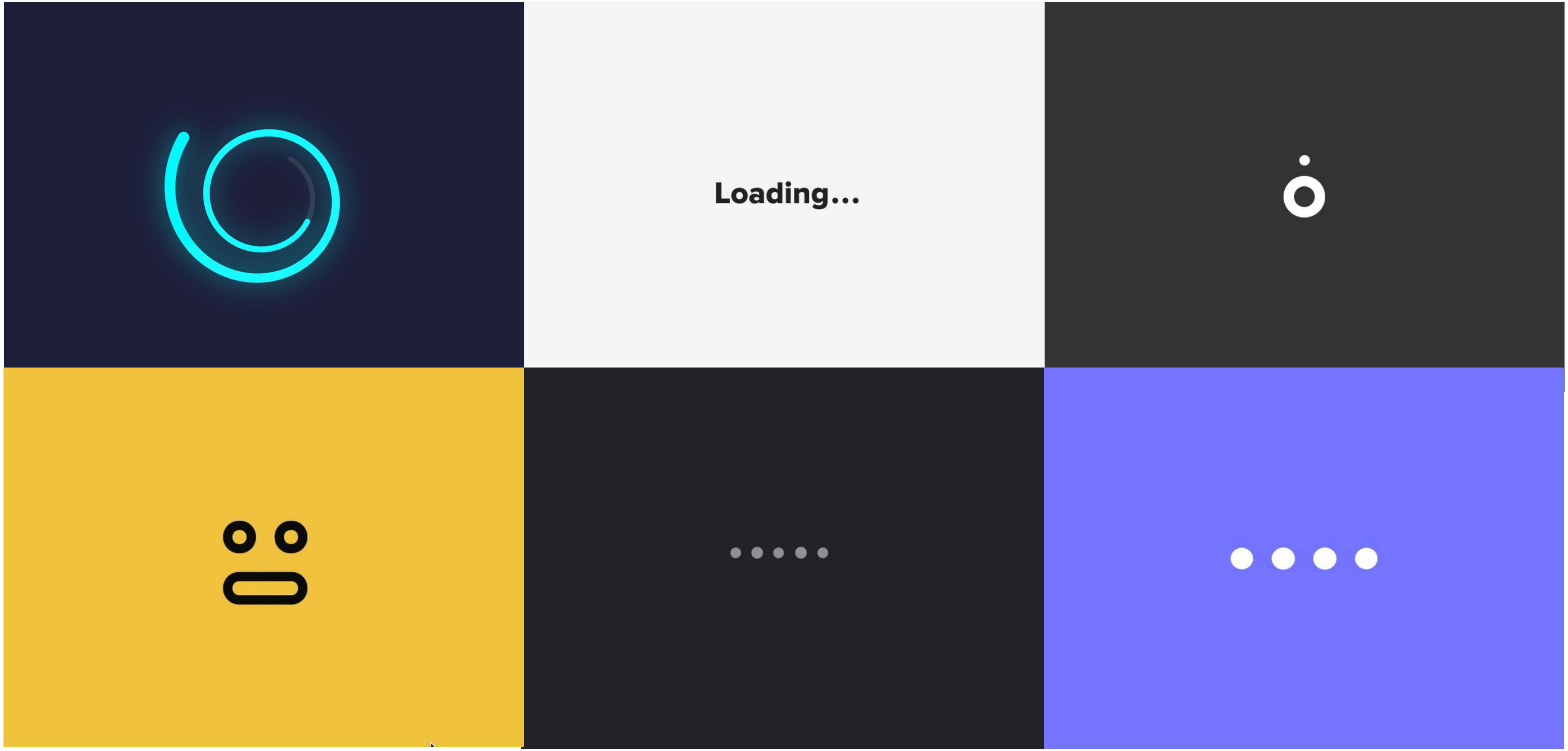
"... executives at a Houston airport faced a troubling customer-relations issue. Passengers were lodging an inordinate number of complaints about the long waits at baggage claim.

In response, the executives increased the number of baggage handlers working that shift.

The plan worked: the average wait fell to eight minutes, well within industry benchmarks. **But the complaints persisted.**"

- The New York Times, August 2012

Designing around waiting times



Student Evaluation Of Teaching and Units

Complete a SETU for this unit

At Monash University, we are always seeking ways to improve your learning experience. One way we do this is by asking for your feedback on the content, structure, assessment tasks and learning technologies used in this unit. Please take the time to complete this survey and help us to improve your learning experience.

Look for the SETU link on this week's Moodle page.

Student Evaluation of Teaching and Units (SETU)

Incomplete SETUs can also be saved for later completion.

Next session

- Voice and Multimodal Interaction Systems
- Onboarding techniques and dark UX

Reminders

- **Peer Evaluation for Stage D is open**
  - Rate your group members and provide feedback to staff.
- **Stage E + F are submissions next week**
  - Stage E: Group high fidelity prototype (due Friday week 6, 11:55PM)
  - Stage F: Individual presentation (in-class, Week 6 Tutorial P2)