# **Interaction Design**

[**Interaction Design 1**](#_pzqsadi96mhi)

[**26/09/23 Introduction 5**](#_lkazf4dq9g7)

[Understanding What to Design: 5](#_o98662wguc13)

[Involvement in Interaction Design: 5](#_6kiop5hrihm7)

[Core Characteristics of Interaction Design: 5](#_lzyymuf5vlo3)

[Understanding/Conceptualizing Interaction: 5](#_jl0ydqn65szo)

[Understanding/ analysing the Problem Space: 6](#_bfly77o85868)

[From Problem Space to Design Space: 6](#_ymyxponf8ada)

[Why Do We Need a Conceptual Model: 6](#_ie8rljfcb6w)

[Main Components of a Conceptual Model: 6](#_5wdsj3lk2fki)

[Interface Metaphors: 6](#_qbkmzgpl0nl0)

[- Pros: 6](#_cusl7stsg234)

[- Cons: 6](#_6au3cb5pl10g)

[Interaction Types: 7](#_h65ju310mjbx)

[Interaction Types: 7](#_ija6e3p1nsw0)

[Instructing: 7](#_m2b086tr2mu7)

[Conversing: 7](#_l2rw2pms1d02)

[Manipulating: 7](#_wh261mtcmvr2)

[Exploring: 7](#_isd5cu1im6dp)

[Key Points: 7](#_3w0z1rt4sr1k)

[**03/10/23 Design 8**](#_6qx4wjqhssjp)

[Conceptual Design: 8](#_k4otxzzf6z1a)

[Brainstorming: 8](#_up1ww62t414h)

[Card Sort: 8](#_u0mx4ubl0pei)

[Semantic Networks: 8](#_ozqcjcb7lr1y)

[Technology Myopia: 8](#_knwe4j46w9c7)

[Personas: 9](#_x18ut5inc3zy)

[Using Personas: 9](#_97jsbm8vri82)

[Advantages and Disadvantages of Personas: 9](#_fysbwr1ghium)

[- Pros: 9](#_699lofp4z8mb)

[- Cons: 9](#_nunekvi5wv86)

[Scenarios: 9](#_i7flv4gxqize)

[Flowcharts: 9](#_j34nwnp3v8rb)

[Cognitive Walkthroughs: 9](#_gitp4t9ufg0s)

[Physical Design: 10](#_w8v83i1q4eu8)

[Prototype Types: 10](#_spxs8532zzuv)

[Prototype Fidelity - Low-Fidelity: 10](#_eg56qayp9g4t)

[- Pros: 10](#_nipz3supaf35)

[-Cons: 10](#_ao2hj4la3rrh)

[Evaluation: 10](#_f6vcqk669mna)

[Wireframes: 10](#_yamuwqet10uy)

[Functional Prototypes: 11](#_4af459e7kigr)

[Key Points: 11](#_li76cakgg09f)

[**10/10/23 Design Patterns 12**](#_e98mdchpedz1)

[Basics of Information Architecture 12](#_zbq04n6xwbuc)

[Basics of Physical Structure 12](#_9qpuc8d3v63)

[Multiple Windows 12](#_qjvpihveqbb3)

[One-Window Paging 12](#_7mnvl61f2ktk)

[Tiled Panes 12](#_k1pe8dqj2h3u)

[Application Structure Patterns 12](#_u7p5c2kj0425)

[Two-Panel Selector Pattern 13](#_2jat96mf8gji)

[Canvas + Palette Pattern 13](#_a4e4nxihd7qb)

[One-Window Drilldown Pattern 13](#_fex57tp8y9lo)

[Wizard Pattern 13](#_4fzbak9sq00c)

[Navigation 13](#_syxqiqne9hpd)

[Navigation/Orientation Patterns 13](#_7ewr13pc5qic)

[Clear Entry Points Pattern 14](#_tbinoqoof0mz)

[Global Navigation Pattern 14](#_d3mxzjij0ol2)

[Hub and Spoke Pattern 14](#_aiyio4iw94bb)

[Pyramid Pattern 14](#_81c4o3fjgwkw)

[Sequence Map Pattern: 14](#_6b735mxo4mur)

[Breadcrumbs Pattern: 15](#_9lqj0ofgktbt)

[Page Layout: 15](#_igcm913mluuj)

[Page Layout Patterns: 15](#_1qp3zx5krnvo)

[Centre Stage Pattern: 15](#_bdit8wt6qd1f)

[Tiled Sections Pattern: 15](#_3hyjci8fyz49)

[Card Stack Pattern: 15](#_n4e9j4rdqzb4)

[Action and Command Patterns: 16](#_43bad88whj4k)

[Button Groups Pattern: 16](#_igo44ewtykrn)

[Action Panel Pattern: 16](#_i4ki3t1pogpc)

[Prominent 'Done' Button Pattern: 16](#_61xcededvuu)

[Key Points: 16](#_9xceyscht62e)

[**17/10/23 Design Principles 17**](#_khtu2j3c0r3n)

[Framework for Design Principles: 17](#_ys0st9x81uv9)

[Comprehensibility: 17](#_5fgy4esurp1)

[Learnability: 17](#_dh7znpdmgldf)

[Principles of ID: 18](#_yo1ytt8l9wjr)

[Effectiveness/Usefulness: 18](#_z4kaahmlj2w6)

[Effectiveness/Usefulness: Utility: 18](#_bsz77jt0f0qa)

[Effectiveness/Usefulness: Safety: 18](#_4bhj3u15t4nz)

[Effectiveness/Usefulness: Flexibility: 18](#_50k3cki7lrou)

[Effectiveness/Usefulness: Stability: 18](#_d7hl70r0bakw)

[Efficiency/Usability: 19](#_xtqi48h8t4le)

[Efficiency/Usability: Simplicity: 19](#_yyzvkrjt6nog)

[Efficiency/Usability: Memorability: 19](#_aynm3x3uu0nq)

[Efficiency/Usability: Predictability: 19](#_ioresyxj8atk)

[Efficiency/Usability: Visibility: 19](#_kaloq0c3t909)

[Grouping 20](#_g1raxe8jbnxy)

[Gestalt Principles of Perception 20](#_y3i3zcgjg8ah)

[White Space 20](#_bslay9aguo1e)

[Stimulus Intensity 21](#_gpyywvauqerg)

[Proportion 21](#_ys3ib45ggp72)

[Balancing Screen Complexity and Usability 21](#_rnazuqgx283z)

[Resolution (closure) 21](#_h7kxw0kl0yrv)

[Usability Goals 21](#_t37g5dha65h4)

[Simplicity 21](#_h955bgoa4sxu)

[**24/10/23 Colour 22**](#_rdmm4um9f4oi)

[The Human Perceptual System 22](#_w6ozla7oj499)

[Colour Perception: Limitations 22](#_a0rk3cas1vqt)

[Colour Deficiencies 22](#_l731x5mjkaio)

[Individual & Cultural Issues 22](#_akmogq1b2wnb)

[Using Colour in ID 22](#_yzoar4ycbmmu)

[Colour Concerns for ID 22](#_1rgp812j450y)

[Technical Issues Concerning Colour 23](#_4kqeaq7n2f0d)

[Key Points 23](#_qf8itnusn17b)

[**24/10/23 Icons 24**](#_iw3tb3vziv3m)

[Introduction: 24](#_vpf88dugwks4)

[Icons: Human Issues: 24](#_ogftyttnxlto)

[Icons: Human Issues - Articulatory Distance: 24](#_ybsmk3ze8ev0)

[Icons: Human Issues - Icon Analysis: 24](#_n3rrkk31q4tv)

[Using Icons in Interaction Design 25](#_skwb59ixudbe)

[Using Icons in Interaction Design - Search: 25](#_jugh5fki3e0r)

[Using Icons in Interaction Design - Real Estate: 25](#_bzyj2n6vosf4)

[Using Icons in Interaction Design - Conventions: 25](#_t366ms2rkvyj)

[Using Icons in Interaction Design - Context: 25](#_jpn1tkltyb1u)

[Using Icons in Interaction Design - Globalization: 25](#_s4cc0rjk877o)

[Technical Issues - Terminology: 26](#_8z14pcl2oqw9)

[Technical Issues - Icon Grammar: 26](#_63c7f7tz5v5x)

[Technical Issues - Deconstruction: 26](#_t14mmsntmoyg)

[Technical Issues - Icon Size: 26](#_k67kptig5xse)

[Key Points: 26](#_s0yffe714rij)

[**31/10/23 Text 27**](#_82p9k1pl0hqp)

[Text: Human Issues: 27](#_91jyv4x3nchw)

[Text: Using Text in ID: 27](#_tfuhq9qvmrj2)

[Text: Using Text in ID - Legibility: 28](#_f6xjgglfv68)

[Text: Using Text in ID - Readability: 28](#_r68hwelhav0o)

[Text: Using Text in ID - Technical Issues: 28](#_gw8z4g306mfl)

[Key Points: 28](#_707nuogvvz2)

[**31/10/23 UI Components 30**](#_vglqbms9rrxb)

[Introduction: 30](#_nihytx8l4ypn)

[WIMP Interfaces: Windows: 30](#_nx2gw93bbkqq)

[1. MDI (Multiple Document Interface): 30](#_tn0zwal7llye)

[2. SDI (Single Document Interface): 30](#_ikzhn9mv060r)

[3. TDI (Tabbed Document Interface): 30](#_r7helnu06k29)

[WIMP Interfaces: Icons: 30](#_16mmv4hh5z5x)

[WIMP Interfaces: Menus: 31](#_fm37qnj4zk4)

[Other Components: 31](#_ph61mnqjkp6p)

[Key Points: 31](#_8hzssq5of3id)

[**14/11/23 Speech & Hearing 32**](#_iuwt3f6uj1la)

[Human Perceptual System: 32](#_swy4w4qpho7t)

[Human Perceptual System: Hearing: 32](#_zfsg93qz9ijq)

[Human Perceptual System: Speech: 32](#_98d0vde9ja4o)

[Human Perceptual System: Non-Speech: 32](#_m5to2jwtxz1e)

[Using Sound in Interaction Design: 32](#_6psemfdldi7f)

[Speaking Sounds: Speech Applications: 33](#_fvynpm5ajwn)

[Other Kinds of Sounds: Non-speech Applications: 33](#_ayv8y2ttijcq)

[Talking Pictures: Auditory Icons: 33](#_yd42d7pgdgj2)

[Musical Sounds: Earcons: 33](#_6qk069mdg1i2)

[Sounds in Menus: Hierarchical Earcons: 33](#_o9ykd8rhnrv)

[Many Sounds Together: Parallel Earcons: 33](#_93448ax2fw2t)

[Rules for Making Sounds: Guidelines for Earcons: 33](#_b93i90y8a0ts)

[Musical vs. Talking Pictures: Earcons vs. Auditory Icons: 34](#_injzm6o9v4wa)

[Key Points: 34](#_t9qs2hd2b6v7)

# **26/09/23 Introduction**

Aim of interaction design: Improve usability and bring user-centred design into the process.

Goal: Design interactive systems that are easy, effective, and enjoyable from the users’ perspective.

- User-centred design leads to better design.

## Understanding What to Design:

- Consider who the users are, what activities they’d use the system for, and how they’d interact with the system.

- Optimise user interactions to match their needs and activities.

- Achieved by understanding users, considering what helps them, and using user-centred methods.

## Involvement in Interaction Design:

- Identifying user needs and requirements.

- Developing alternative designs to meet these requirements, select the best design.

- Building interactive prototypes and testing with users for feedback.

- Continuous evaluation.

## Core Characteristics of Interaction Design:

- User involvement throughout development.

- Clearly identify usability and user experience goals.

- Iteration throughout core activities.

- ^^ Aids in understanding user needs, their capabilities, and cultural differences.

## Understanding/Conceptualizing Interaction:

- Conceptualise the current user experience and how we can improve it.

- Need to think about: what we want to create, what are our assumptions, what are our claims, what do we want to achieve.

## Understanding/ analysing the Problem Space:

- Articulate the problem space before focusing on design.

- Consider:

* problems with existing product
* how the proposed design can overcome these problems
* how will your new design affect existing ways of doing things

## From Problem Space to Design Space:

- Develop a conceptual model:

* Outline of what users can do with the product.
* Structure concepts and relationships.

## Why Do We Need a Conceptual Model:

- Provides a working strategy and framework.

- Helps avoid misunderstandings and confusion.

- Helps designers understand user perceptions.

## Main Components of a Conceptual Model:

- Metaphors and analogies to understand what a product is for/ how to use it.

- Concepts exposed to users.

- Relationships between concepts.

- Mappings between concepts and user experience.

## Interface Metaphors:

- Leverage familiar knowledge to make learning new systems easier.

### - Pros:

* Understand underlying conceptual model
* Makes computer applications more accessible to more users

### - Cons:

* Conflict with design principles
* Places constraints on designers in the way the conceptualise a problem

## Interaction Types:

= way of thinking about how to best support user activity

## Interaction Types:

### *Instructing:*

- Users issue commands, select options.

- Common for quick and efficient actions.

- Effective for repetitive tasks.

### *Conversing:*

- Two-way communication with the system.

- Ranges from voice recognition to natural language dialogues.

- Familiar interaction for novices.

### *Manipulating:*

- Exploits users' physical world knowledge.

- Virtual objects can be manipulated.

- Direct manipulation framework = user feels they are directly controlling the digital objects.

### *Exploring:*

- Users move through virtual or physical environments.

- Examples include 3D virtual worlds and smart homes.

## Key Points:

- Interaction design focuses on creating quality user experiences.

- Understanding the problem space is essential.

- Conceptual models help inform design.

- Decisions about conceptual design should precede physical design.

- Interaction types help determine how to support user activities.

# **03/10/23 Design**

## Conceptual Design:

- Use collected information from the discovery phase to create design ideas.

- Structure information space and generate solutions then move onto the physical design phase.

- Tools used: brainstorming, card sort, semantic networks, personas, scenarios, flowcharts, cognitive walkthroughs, use cases. Using the tools iteratively for best results.

## Brainstorming:

- Team activity for generating ideas.

- Involves sketches and diagrams.

- Storyboarding helps visualise user progress through a task.

- Goal: Generate many creative ideas.

## Card Sort:

Users group items into categories based on their perspective.

The results help designers create user-friendly website structures and navigation menus that align with users' mental models and preferences.

## Semantic Networks:

- Semantic networks focus on the connections and relationships between things

- Less user-centred but structured.

- *Advantages*: easy exploration, graphical view, related elements clustering.

- *Disadvantages*: requires knowledge of the problem space, may lead beyond scope, no formal semantics.

## Technology Myopia:

- Interaction design should consider socio-technical aspects.

- Avoid focusing only on technology.

- Be sensitive to human communication, implicit knowledge, and non-technological aspects.

- Integrate technology and human activities carefully.

## Personas:

- Create characters that represent potential users from our knowledge elicitation.

- Give them faces, skill set, and backgrounds.

- These personas help designers make better decisions.

### Using Personas:

- Use personas to discuss and understand users.

- Don't create too many personas; they should reflect real users from the discovery phase (unique stakeholders).

### Advantages and Disadvantages of Personas:

#### - Pros:

* easy to create
* provide consistency
* make users feel real

#### - Cons:

* can be hard for international audiences
* too many personas make things difficult
* not a substitute to using real users

## Scenarios:

- Scenario = typical task a persona may undertake, including the:

* conditions that exist
* activities the persona will engage in
* outcomes of those activities

- Focus on what users do, not on the system itself.

## Flowcharts:

- Use symbols to represent how screens or pages connect.

- Helps visualise how users navigate through a design.

## Cognitive Walkthroughs:

- Test scenarios using flowcharts.

- Pretend to be the user and look for problems.

- Ask questions to ensure users can complete tasks.

## Physical Design:

- Think about how things will look and what components are needed.

- Create prototypes/ wireframes to see how designs work in action.

## Prototype Types:

- *Full prototypes* have complete functionality but not full performance

- *Global prototypes:* high level of functionality included but not complete (how things work together)

- *Local prototype:* focusses on a single detail (perfecting one part)

- *Horizontal prototypes* have high-level functionality but low level detail (car parts but no detail of car colour etc) (like an overview of different ideas).

- *Vertical prototypes* include high and low-level functions for a part of the system (focus on one part eg the wheel but is very detailed, e.g. tyres, rims, material and finish).

## Prototype Fidelity - Low-Fidelity:

- Use materials like paper for early prototypes.

### - Pros:

* they're cheap
* easy to change
* don’t need special software
* help gather feedback

### -Cons:

* not interactive
* cant calculate response times

## Evaluation:

- Regularly test and assess designs.

## Wireframes:

- Define basic layouts and screen components.

- Use information from the discovery phase.

## Functional Prototypes:

- High-fidelity prototypes with interactive features.

- Created using software tools.

## Key Points:

- Design involves both conceptual and physical phases.

- Avoid getting too focused on technology.

- Use personas, scenarios, and prototypes to create user-friendly designs.

# 10/10/23 Design Patterns

## Basics of Information Architecture

- Structure content and functionality

- Organise, label, and guide users through the UI

- Base decisions on nature and domain of the application, users' domain knowledge, comfort levels with computers, and matching users' mental models of the domain

## Basics of Physical Structure

- Translate design into windows, pages, and controls

- Choose between multiple windows, single window with tiled panes, or one window with swapping-out content based on user tasks and technology constraints

- Analyse tasks users will perform, considering the need to work in multiple areas simultaneously

## Multiple Windows

- Best for users who want to customise their screen

- Multiple windows can be confusing for users, especially infrequent ones

- Use when users need to see multiple windows in parallel

## One-Window Paging

- Suitable for simple web applications

- Shows one page at a time

- Conserves space and is ideal for small devices

## Tiled Panes

- Great for users who want to see a lot at once

- Can take up a lot of screen space

- Users responsible for focusing their attention

- Not suitable if you need to lead users along a specified path

## Application Structure Patterns

- Two-panel selector

- Canvas + palette

- One-window drilldown

- Wizard (linearizing a task)

### Two-Panel Selector Pattern

- Two side-by-side panels e.g. emails

- Use for presenting lists of objects, categories, or actions

- Allows users to navigate items at their own pace

- Requires a large enough display to show separate panels

### Canvas + Palette Pattern

- Iconic palette next to a blank canvas e.g. adobe photoshop or paint

- Used in graphical editors

- Palette buttons create objects on the canvas

- Utilises visual recognition and iconic norms

### One-Window Drilldown Pattern

- Show each page in a single window e.g calendar

- Contents are replaced when drilling down e.g. opens event when date selected

- Suitable for applications with many pages of content

- Keeps options simple and clear

### Wizard Pattern

- Leads users step by step e.g. checkout forms

- Suitable for long or complicated tasks

- Useful for tasks with many user-made decisions

- Requires users to surrender some control

- Inflexible for some users

## Navigation

- Minimise commute within the application

- Use signposts and wayfinding (tabs/titles etc)

- Provide good signage, environmental cues, and maps

- Minimise cognitive cost of context switching

## Navigation/Orientation Patterns

- Clear entry points

- Global navigation

- Hub and spoke

- Pyramid

- Sequence map

- Breadcrumbs

### Clear Entry Points Pattern

- Present task-oriented entry points

- Use for task-based applications or first-time users

- Provides a clear starting point and delineates the application's purpose eg e-commerce websites (home, garden categories etc)

### Global Navigation Pattern

- Consistent links/buttons on each page e.g. social media (home, notifications & DMs etc)

- Useful for applications with many sections or tools

- Provides an overview of the UI and facilitates exploration

### Hub and Spoke Pattern

- Isolates sections into mini-applications (spokes) e.g. online banking

- Suitable for discrete tasks or sub-applications (pay & transfer , deposit cheque)

- Restricts workflow and prevents clutter by only showing pertinent actions (exit, help, back etc)

- Forces users to focus on one thing at a time

### Sequence Map Pattern:

- Use for linear applications.

- Saves user time and shows progression.

- Place a small map along the edge of the page.

- Keep it in one line or column.

- Highlight the current location.

- Use labels or step numbers e.g. installation wizard

### Breadcrumbs Pattern:

- Show parent pages in the hierarchy e.g. home > downloads > cv.pdf

- Useful for hierarchical apps.

- Helps users understand their position

- Context, not history (indicates current location relative to structure not the path taken to get there)

## Page Layout:

- Manipulate attention for meaning and interaction.

- Visual hierarchy, flow, grouping, alignment, and dynamics.

## Page Layout Patterns:

### Visual Framework Pattern:

- Keep consistent layout, colours, style e.g. MS Office applications

- Applies to multiple pages, adds familiarity and cohesion.

- Helps users know where they are.

### Centre Stage Pattern:

- Emphasise the most important part e.g. snapchat

- Smaller panels for secondary content e.g. snapchat the chat section or stories

- Focus on a central task or object e.g. taking photos/ videos

### Tiled Sections Pattern:

- Use strong titles to identify content sections e.g. pinterest boards

- Useful for organising lots of content

### Card Stack Pattern:

- Use panels or cards, show one at a time e.g. MS Outlook emails

- Tabs for access e.g. inbox, junk

- Ideal for managing extensive content.

## Action and Command Patterns:

- Patterns for presenting actions, non-instantaneous actions, and sequences.

### Button Groups Pattern:

- Cluster related actions as aligned buttons e.g cut, copy and paste in MS word

- For 2-5 related actions.

- Promotes self-explanation.

### Action Panel Pattern:

- Display a large group of actions as a panel e.g. MS Paint

- When there are too many actions for button groups or menus e.g. fill, select colour, draw shape

### Prominent 'Done' Button Pattern:

- Place a concluding button at the end e.g. UberEats

- Use when a clear conclusion is needed e.g. place order button

## Key Points:

- Importance of app structure, content, and navigation.

- Various patterns for UI design.

- Navigation's critical role.

- Consider design principles and layout patterns.

# **17/10/23 Design Principles**

- ID blends tech knowledge and human psychology.

- Logic and aesthetics matter.

- Aesthetically pleasing interfaces seem easier to use (aesthetic-usability effect).

- Goal: Craft elegant solutions for complex interactions.

- Principles are the guiding stars.

- Cover comprehensibility, learnability, effectiveness, efficiency, grouping, and more.

- 2 main categories: effectiveness and efficiency.

- Principles are flexible guides, not rigid rules.

- They help make informed decisions, even when faced with contradictions.

## Framework for Design Principles:

- Design principles are interconnected. For example, predictability and memorability go hand in hand.

- Two main usability goals are the Comprehensibility Barrier and the Learnability Barrier.

- Effective interaction design requires ensuring functionality is accessible, comprehensible, learnable, and ultimately useful.

- Understanding these principles is vital for creating a usable interface.

## Comprehensibility:

- Comprehensibility is the most crucial aspect of interaction design.

- If users cannot understand the user interface (UI), it becomes useless, even if it has valuable functionality.

- The design's success depends on how it effectively communicates its features and functions to the user.

## Learnability:

- Learnability is based on comprehensibility. If the design is not understandable, it cannot be learned.

- Even highly comprehensible user interfaces may still present a learning curve.

- Learnability and comprehensibility work together, creating a feedback loop that can be improved by applying design principles.

## Principles of ID:

- The goal of effectiveness and usefulness includes utility, safety, flexibility, and stability.

- The goal of efficiency and usability encompasses simplicity, memorability, predictability, and visibility.

## Effectiveness/Usefulness:

- Effectiveness refers to the design's usefulness in fulfilling users' needs.

- It ensures the design offers the required functionality to achieve users' goals, such as providing information on a website or tools in a drawing program.

### Effectiveness/Usefulness: Utility:

- Utility is about what users can do with the system.

- A design with good utility proves effective for specific tasks and applications.

### Effectiveness/Usefulness: Safety:

- Safety is vital for effective design, particularly in mission-critical environments.

- Safety involves physical parameters and ensuring the system does not put users at risk or compromise their work.

### Effectiveness/Usefulness: Flexibility:

- A flexible tool can adapt to diverse environments and meet various needs.

- Customization allows a tool to be tailored to individual users' preferences, enhancing its flexibility.

### Effectiveness/Usefulness: Stability:

- A stable system that consistently performs well is more useful than an unstable one that frequently crashes.

### Efficiency/Usability:

- Efficiency measures how easily and quickly users can accomplish tasks with the design.

- It ensures that the system does not waste users' time or require them to perform unnecessary and complex procedures.

#### Efficiency/Usability: Simplicity:

- Simplicity enhances a design's usability by making it easy to understand, learn, and remember.

- Simplicity aims to include only the essential elements required for clear communication.

- Concepts like *Ockham's Razor*, the *80/20 Rule*, and *Satisficing* highlight the importance of simplicity in design.

- Progressive Disclosure shows only necessary information at a given moment.

- Constraints, both physical and psychological, can be applied to simplify design.

- Physical constraints like paths, axes, and barriers limit movement.

- Psychological constraints, like conventions, mapping, and symbols, influence user behaviour.

#### Efficiency/Usability: Memorability:

- High memorability makes user interfaces easier to learn and use.

- Parameters like location, logical grouping, conventions, and redundancy influence memorability.

#### Efficiency/Usability: Predictability:

- Predictability helps users anticipate the results of their actions.

- It is based on consistency, generalizability, conventions, familiarity, and location.

- Striving for consistency is important, but the designer must be certain it is the right choice.

- Generalizability helps apply previous experience to similar situations.

- Familiarity with menu names and options makes it easier for users to locate objects and functions.

#### Efficiency/Usability: Visibility:

- Visibility ensures that the user is aware of the system's components, processes, functionality, and feedback from user actions.

- To avoid overwhelming users, visibility should be combined with principles of progressive disclosure and simplicity.

- Overload should be avoided by carefully

## Grouping

- Involves arranging information in a way that helps users better understand it.

- Consider a weather app: Grouping data (like temperature, humidity, and wind speed) aids users in easily understanding and accessing information.

## Gestalt Principles of Perception

- Explains how we naturally perceive and group elements:

- Figure-ground: Think of the classic Rubin face/vase illusion. We see either the faces or the vase in the foreground as our minds differentiate between objects and their backgrounds.

- Proximity: When designing a webpage, placing related menu items close to each other makes it clear to users which options are grouped together.

- Similarity: If you're creating a chart, using similar colours for related data points helps users see patterns and relationships.

- Common fate: Imagine designing a car navigation system. When different elements move together, like icons for nearby points of interest, users understand they're related.

- Closure: If you're designing a logo, using shapes that suggest completeness, even if parts are missing, helps users perceive the whole picture.

- Good continuity: For a mobile app, designing smooth transitions between screens makes the user experience feel seamless.

- Area: Designers often use the principle of area to make buttons or important elements larger, indicating their significance.

- Symmetry: When designing a website layout, symmetrical elements around a central axis create a visually balanced and pleasing design.

- Surroundedness: Consider a photo gallery app. Highlighting the currently selected image and dimming the rest uses the principle of surroundedness to guide the user's focus.

- Prägnanz: In a messaging app, a simple and stable interface design encourages users to focus on the conversation.

## White Space

- Is an empty area on a page or screen.

- White space around text in a magazine article improves readability and aesthetics.

## Stimulus Intensity

- Means that we first notice attributes like colour, shape, or size before we process their meaning.

- For example, red highlighting in a dashboard draws attention to important alerts.

## Proportion

- Visually conveys hierarchy.

- Website headings in varying text sizes denote section hierarchy.

## Balancing Screen Complexity and Usability

- If a mobile app interface is too cluttered with options, users may find it confusing.

- An overly simplistic design might lack essential features.

- Balancing complexity and simplicity is crucial in mobile app design for a positive user experience.

## Resolution (closure)

- When users feel they've completed a task.

- For instance, in e-commerce, a successful purchase gives users this sense of task completion.

## Usability Goals

- e.g understandability

- Simple navigation in e-learning, or easy flight booking for travel sites.

### Simplicity

- Supports usability goals. For a travel booking website, simplifying the flight search process, with a clean and straightforward interface, enhances user satisfaction.

# **24/10/23 Colour**

- Colour is crucial in GUI design for aesthetics and information organisation.

- Proper colour use enhances user interaction but can also lead to confusion and errors.

## The Human Perceptual System

- Colour perception depends on how light interacts with objects.

- Reflected light waves' frequencies determine an object's colour.

- Cones in the retina detect colour, with red, blue, and green sensitivities.

## Colour Perception: Limitations

- Eyes are most sensitive to middle frequencies (green and yellow).

- Difficulty in perceiving colour in the periphery.

- Using colour to delineate shapes is less effective than contrast.

## Colour Deficiencies

- Approximately 8% of males and 0.4% of females have colour deficiencies.

- Most common is reduced green sensitivity (deuteranomaly).

## Individual & Cultural Issues

- Colour associations and expectations vary across cultures.

- Colour can evoke emotional responses.

- Important in interaction design, especially for global audiences.

## Using Colour in ID

- Colour aids in clarification, relation, differentiation, searching, comprehension, and enhancing tasks.

- Redundant coding can reinforce information and prevent issues.

## Colour Concerns for ID

- Limitations in distinguishing colours; recommend using 5-9 colours for memory tasks.

- Avoid incompatible colour combinations and take background colour into account.

## Technical Issues Concerning Colour

- Monitors vary in displaying colours; consider different environments.

- Understanding colour systems (hue, value, chroma).

- Colour contrast and schemes affect legibility and user experience.

## Key Points

- Proper colour treatment is vital in UI design.

- Follow guidelines based on human visual receptor strengths.

- Test colour choices to ensure they enhance, not hinder, user performance.

# **24/10/23 Icons**

## Introduction:

- Icons are essential components of graphical user interfaces (GUI).

- Icons are small images used to represent objects (e.g., files or folders), functionality (e.g., print), or attributes (e.g., colours in paint programs).

- Icons can also convey critical system warnings and errors.

## Icons: Human Issues:

- Icons don't give explicit instructions; users infer their meaning.

- New users must learn to connect icons with functions.

- Icons can be both symbolic and functional.

- They should align with real-world associations.

- Users adapt and generalise icon use.

- GUIs rely on real-world metaphors, which influence users' expectations and perceptions.

- Carefully choose icons for efficient decision-making.

### Icons: Human Issues - Articulatory Distance:

- Users first respond to the physical attributes of icons and gradually make associations to understand their meaning.

- The articulatory distance measures the degree of separation between a user's observation and understanding of an icon's meaning.

- Deconstruction of an icon involves perceptual and cognitive levels of understanding.

- Examples include the envelope symbol for email.

### Icons: Human Issues - Icon Analysis:

- An icon analysis chart helps determine the complexity of an icon, avoiding ambiguous and personal associations.

- Simpler relationships between image and functionality result in smaller articulatory distances.

- Once learned, complex visual relationships can become second nature.

- Icons that do not convey any meaning can cause delays and errors.

## Using Icons in Interaction Design

### Using Icons in Interaction Design - Search:

- Users respond to the physical qualities of icons before their semantic associations.

- Icons should be sufficiently differentiated for effective search activities.

- Detail, colour, size, shape, and location influence the way users perceive icons.

- Properly designed location enhances the effectiveness of icons in search.

### Using Icons in Interaction Design - Real Estate:

- Icons save screen real estate, especially on small computing devices.

- Tooltips (short labels that pop up when users hover over an icon) can aid in learning icons.

- The choice between icons and text depends on factors like display size and users' computer literacy.

### Using Icons in Interaction Design - Conventions:

- Icon conventions should be employed when appropriate to enhance user understanding.

- Be cautious about the evolution of conventions, considering local vs. global acceptance.

- Examples of conventions include web icons for audio, secure connections, and home.

### Using Icons in Interaction Design - Context:

- Icons are perceived in relation to other screen elements, providing a frame of reference.

- Context is essential for understanding the meaning of icons.

- The power of context can dramatically affect how users interpret icons.

### Using Icons in Interaction Design - Globalization:

- Images in icons are linguistically agnostic, making them suitable for a global community.

- Consensus is necessary for symbols to work internationally, but finding agreement can be challenging.

- Understanding cultural meanings is crucial for global icon design.

### Technical Issues - Terminology:

- Terminology used to describe icons is often inconsistent and cryptic.

- Various types of signs, including phonograms, pictograms, abstract shapes, ideograms, and logograms, are used in icons.

- Icons often operate on multiple levels of meaning simultaneously.

### Technical Issues - Icon Grammar:

- Icon grammar involves principles that govern the internal structure of icons.

- Icons should follow observable, logical, predictable, and consistent grammatical rules.

- Icon grammar specifies how elements can be combined, their arrangement, and the representation of each element.

### Technical Issues - Deconstruction:

- Simple shapes and forms serve as the basic building blocks of complex symbols.

- Elements like basic shapes, indicators, graphical dynamics, arrows, and styles are used to construct icons.

- Combining symbols and indicators can create a rich vocabulary of actions and procedures.

### Technical Issues - Icon Size:

- Icons are always square and standardised at fixed dimensions.

- Icons should be created in various sizes and colour depths to suit different display environments.

- Designing icons for small sizes requires skill to convey information effectively.

## Key Points:

- Icons are crucial components of modern graphical user interfaces.

- Icons have advantages (recognition over recall, space conservation) and disadvantages (cultural issues, ambiguity).

- Icons' attributes like colour, detail, size, shape, and location impact user performance.

- Understanding the context of icons is vital for their design.

- Icons should be designed with cultural awareness, context, and global considerations in mind.

# **31/10/23 Text**

## Text: Human Issues:

- Examining how and why people read text differently on paper and screens.

- The goal is to design interfaces that tap into strengths and avoid weaknesses, making it easier for people to read on screens.

- Eye movement during reading involves saccades (quick movements) and fixations (pauses on areas of interest).

- Reading involves two steps:

* distinguishing letters, words, shapes
* associating meaning

- Lowercase words are generally more readable due to distinct shapes, particularly ascenders and descenders.

- Identifying letters or words from a distance is easier in uppercase.

- Reading from screens and paper differs.

Consider the benefits of digital documentation and the advantages of paper over screens:

- Paper is more portable, doesn't require electricity, and offers spatial cues.

- Lowercase is more common and advantageous.

- Reading involves different activities, such as continuous reading, scanning, and screen reading.

- Paper is more flexible than electronic media, helps with spatial memory, and supports information retrieval.

- Consider spatial memory when designing to help users understand the document's structure.

- Spatial cues are lost when scrolling through electronic documents.

- Electronic documents are typically scrolling but can also be paged.

- The choice between scrolling and paging depends on the task, layout, and user preferences.

- Users often prefer paging, which allows spatial memory to be employed.

## Text: Using Text in ID:

- Different types of text in a UI serve various purposes.

- Commentary text informs users and makes system states or functionality visible.

- The most common form of commentary text is help text.

- Instrumental text does work and represents functionality directly related to user actions.

- Hyperlinks and menu options are examples of instrumental text.

- Legibility is an essential aspect of text presentation.

### Text: Using Text in ID - Legibility:

- Text must be clear, distinct, and have sufficient contrast for legibility.

- Environmental conditions must be considered in design.

- Visual acuity decreases with age, and not all users have 20/20 vision.

### Text: Using Text in ID - Readability:

- Readability = understanding text.

- It's affected by factors like line length, line spacing, formatting, margin width, scrolling, and grammatical issues.

- To maximise readability, language users will understand and avoid jargon.

- Consider formatting, margin width, and scrolling when designing.

- Users prefer medium line lengths, and margin width affects reading performance.

- Visual contrast and brightness are crucial for legibility.

- Text size, line length, and vertical spacing must be balanced for efficiency.

- Alignment can impact reading; left-aligned is preferable.

- Consider the age and visual acuity of users when selecting text size and typeface.

- Choose font typefaces carefully, considering readability and legibility.

- Consider screen resolution, visual acuity, and type of reading.

- Electronic documents may require scrolling, but users prefer paging.

- The length of a scrolled page can affect reading performance.

### Text: Using Text in ID - Technical Issues:

- Different font typefaces include serif, sans-serif, and cursive.

- Font selection and hyperlink presentation are evolving.

- Consider structural differences among languages, such as direction, alignment, delimiters, and diacritical marks.

- Linguistic concerns like puns, abbreviations, and formats should be handled carefully.

- Dynamic text presentation can be suitable for small displays.

- RSVP (Rapid Serial Visual Presentation) and TSS (Times Square Scrolling) are dynamic text presentation methods.

## Key Points:

- Text is crucial for human communication in HCI.

- Understanding human visual limitations and perception is essential for effective text use.

- Proper application of text contributes to a GUI's ease of use.

- Text is essential for UI usability.

# **31/10/23 UI Components**

## Introduction:

- GUIs consist of components (widgets) designed to support specific tasks.

- Major **WIMP** interface components: Windows, Icons, Menus, Pointers.

- Other components include lists, controls, display components, text entry components, tool containers.

## WIMP Interfaces: Windows:

- GUIs use rectangular windows to display application components.

- Types of window managers: OS + user interaction.

- Windows can be maximised, minimised, restored.

- Different window states: maximised, minimised, restored.

- Restored windows can have presentation styles: tiled, overlapping, cascading, interrupted cascading.

- Windowing systems use standard window components with consistent behaviour.

- Applications can be application-centric (MDI) or document-centric (SDI):

### 1. MDI (Multiple Document Interface):

- Example: Microsoft Word (older versions)

- Description: One main window with multiple document child windows inside.

### 2. SDI (Single Document Interface):

- Example: Notepad

- Description: Each document opens in its own separate primary window.

### 3. TDI (Tabbed Document Interface):

- Example: Web browsers like Google Chrome

- Description: Documents or web pages are represented as tabs in a single window for easy switching.

## WIMP Interfaces: Icons:

- Icons are graphical representations that can enhance user experience.

- Proper icon design is essential to avoid ambiguity.

## WIMP Interfaces: Menus:

- Menus are lists of options that provide access to functionality.

- Menus have titles and contain options.

- Menus can be pull-down, cascading, or pop-up.

- Menus should follow standard practices for structure, presentation, and behaviour.

- Advantages: easy access to functionality, should be ordered and not too long.

- Disadvantages: shift user focus, consider keyboard shortcuts.

WIMP Interfaces (Windows, Icons, Menus, Pointers): Pointers

- Usage: Pointers, like the mouse cursor, are employed for user interaction in graphical user interfaces.

- Context-Sensitive: Pointers change shape to provide context-sensitive information. For instance, a hand cursor indicates a clickable link on a web page.

- Cursor Hinting: Cursor hinting, like an hourglass or spinning wheel, informs users about system states or ongoing processes.

- Foveal Vision: Cursors are designed to stay within the user's foveal vision, allowing for precise and context-specific interactions.

- Active Window: Pointers lose context when moved outside the active window, emphasising the window's relevance to the cursor's behaviour.

## Other Components:

- Lists include listboxes, comboboxes, spinners, and sliders.

- Controls consist of command buttons, toolbar buttons, and hyperlinks.

- Display components include scrollbars and splitters.

- Text entry components comprise text boxes and text fields.

- Tool containers encompass toolbars and tool palettes.

## Key Points:

- Interaction designers work extensively with GUIs.

- Improper use of screen components can hinder user tasks.

- Logical assumptions drive user actions; designers need to understand user tasks.

- Innovation should not be stifled by standardisation.

# **14/11/23 Speech & Hearing**

## Human Perceptual System:

- GUIs can become visually overloaded, making auditory stimuli a valuable addition.

- Humans can efficiently use both visual and auditory channels simultaneously.

- Redundant coding (assigning multiple stimuli to an element e.g. text, sound and images) can enhance memory and user experience.

### Human Perceptual System: Hearing:

- Hearing helps us be aware of what's going on in places we can't see.

- Auditory presence (sounds that make you feel like you're there) and sound localization (knowing where the sound is coming from without seeing it) are vital for creating realistic auditory displays.

- Head-Related Transfer Functions (HRTFs) are used to replicate real acoustic environments in virtual auditory settings.

### Human Perceptual System: Speech:

- Speech is a significant part of human interaction with the world.

- Advantages include natural interaction, ease of communication, and multitasking.

- Disadvantages include language barriers and slower processing compared to reading.

### Human Perceptual System: Non-Speech:

- Non-speech auditory cues inform about the success of actions and can be processed quickly, e.g. ‘ding’ in a game = good and ‘boop’ = bad.

- Non-speech sounds are universal and don't rely on language.

- However, they can be ambiguous and need to be learned and familiar.

## Using Sound in Interaction Design:

- The decision to use auditory stimuli depends on the user's tasks.

- Redundant coding can enhance memory and user experience if done thoughtfully.

- Positive and negative auditory feedback should align with the user's task and context.

### Speaking Sounds: Speech Applications:

- Talking to the computer can help us write, record, do tasks, or work with others. For example, you can ask your computer to write a story for you.

### Other Kinds of Sounds: Non-speech Applications:

- Some sounds are like real things (animals, cars), while others are made up by people (music). We use these sounds to show things on the computer, like when a button makes a sound. It's like the computer talks to us using sounds. For instance, when you click on a button, you might hear a "ding" sound.

### Talking Pictures: Auditory Icons:

- These sounds are like little pictures that help us understand and do things. They need to be designed carefully so we don't get confused. Imagine a camera icon making a "click" sound when you take a picture.

### Musical Sounds: Earcons:

- These are like short musical messages for the computer. They use things like high or low sounds, different instruments, and loud or soft parts. They help us with menus and buttons. For example, when you press a button, you might hear a short tune.

### Sounds in Menus: Hierarchical Earcons:

- These special sounds show if something is part of something else, like a folder inside a folder. They tell us a lot of stuff without using many sounds. Think of a menu with folders, and when you open one, it plays a special sound.

### Many Sounds Together: Parallel Earcons:

- Sometimes, the computer plays lots of sounds at once to tell us big stuff. For instance, when you get a lot of new messages, your computer might make different sounds at the same time.

### Rules for Making Sounds: Guidelines for Earcons:

- When we make sounds, we need to think about how they sound, like the kind of sound and how long it lasts. We also need to think about where the sounds come from (left or right) and how loud or soft they are. For example, a loud "ding" might mean something important, and a soft "ding" might mean something less important.

### Musical vs. Talking Pictures: Earcons vs. Auditory Icons:

- We can use both sounds that talk like words and sounds that are like pictures. It depends on what we want to do. We should think about what sounds make sense in different places and for different people all around the world. For example, a phone can have both a ringtone (a musical sound) and a camera click (an auditory icon) for taking pictures.

## Key Points:

- Auditory displays enhance user experience and efficiency in human-computer interaction.

- Humans have the ability to manipulate sound, comprehend complex musical structures, and use auditory cues effectively in interaction design.