Feeling nostalgic yet?

a breakdown of the PC OS GUI window

Research I: Practice-led Image, Media and Communication Research
Lecturer: Lena Frei, Prof. Michael Renner

Character Count: 29428

Timo Rick
timo.rick@students.fhnw.ch

Master of Arts in Digital Communication Environments
HGK FHNW IDCE

Index

0. Abstract 3
1. Introduction 4
  1.1 Research Context and Motivation..... 4
  1.2 Research Question..... 4
  1.3 Glossary..... 4
  1.4 Scope and Limitations..... 6
2. Design and Operating System User Interfaces 7
  2.1 Aesthetics in Digital Culture and OS UI..... 7
  2.2 Beauty, Ugliness, Nostalgia and Taste in Digital Media..... 7
  2.3 Historical Overview of PC OS UI Design..... 8
    2.3.1 Early Graphical User Interfaces and Technical Constraints..... 8
    2.3.2 Skeuomorphism, Material Metaphors, and Depth..... 10
    2.3.3 The Shift Toward Minimalism and Flat Design..... 12
    2.3.4 Contemporary Trends and Post-Flat Aesthetics..... 12
3. Deconstructing the UI Window: Design Parameters 14
  3.1 Defining UI Window Design as a System..... 14
  3.2 Visual Parameters..... 14
    3.2.1 Device..... 14
    3.2.2 Background..... 15
    3.2.3 Window..... 16
    3.2.4 Header..... 18
4. The Interactive Prototype 19
  4.1 The Interactive Prototype as a Research Instrument..... 19
  4.2 System Architecture and Parameter Implementation..... 19
  4.3 Limitations..... 20
5. Conclusion 20
  5.1 Reflections on Practice-Led Methodology..... 22
  5.2 Future Research and Design Directions..... 23
6. References 24

0. Abstract

This research project examines shifting perceptions of beauty, ugliness, and nostalgia in personal computing operating system user interface window design through interactive exploration. By deconstructing historical interface aesthetics into visual and interactional parameters, the project investigates how aesthetic value emerges when these parameters are altered or recombined. An interactive prototype functions as both an analytical and experimental tool, questioning whether interface aesthetics can be systematically mapped or remain subjective and contextual.

1. Introduction

1.1 Research Context and Motivation

My engagement with operating system interfaces is rooted in long-term personal experience of growing up alongside desktop computing environments. The visual languages of past personal computing operating systems user interfaces formed part of my early encounters with digital media and remain strongly associated with periods of my child- and early adulthood. Over time, these interfaces have shifted from everyday tools to cultural artifacts, and I have become increasingly aware of how their aesthetics evoke nostalgia and emotional attachment in myself. This personal familiarity motivated a closer examination of how such visual forms become perceived as beautiful, outdated, or iconic. The project therefore emerges from the intersection of lived experience and design inquiry, using personal historical proximity to these interfaces as a starting point for critically investigating how aesthetic values in UI design are formed, remembered, and reinterpreted.

1.2 Research Question

How can playful, interactive exploration of OS UI window design reveal and challenge shifting perceptions of beauty, ugliness and nostalgia in digital interfaces?

Subquestions: How do users perceive aesthetic value when UI window design parameters are altered, exaggerated or recombined? How can an interactive prototype function simultaneously as an analytical tool for design research and as an experimental medium for engaging with UI perception and history? Can beauty and ugliness in interface design be mapped through design parameters? How do subjectivity, nostalgia and cultural context resist such mapping?

1.3 Glossary

Moving forward I will use the following abbreviations and technical terms which might need introduction:

CSS (Cascading Style Sheets)
A styling language used to control the visual presentation of HTML elements—such as layout, spacing, colors, typography, borders, shadows, and visual effects—separating appearance from structural content.

CSS Variables (Custom Properties)
Named, reusable values defined in CSS that allow centralized and dynamic control of visual parameters. They are especially useful in interactive prototypes, where visual attributes can be adjusted in real time via JavaScript.

GUI (Graphical User Interface)
A visual interface that allows users to interact with a computer through graphical elements such as windows, icons, menus, and pointers, rather than text-based commands.

HTML (HyperText Markup Language)
The standard markup language used to structure content on the web. In interface prototyping, HTML defines the structural elements (containers, text, controls) that form the basis of the user interface.

JS (JavaScript)
A programming language used to implement interactivity and logic in web-based systems. In UI prototyping, JavaScript enables user input to dynamically modify interface parameters, often by updating CSS variables or document object model (DOM) properties.

OS UI (Operating System User Interface)
The visual and interactive layer of an operating system through which users control, navigate, and manage the system.

PC (Personal Computer)
A general-purpose computer designed for individual, private use, typically running a desktop operating system.

PC OS (Personal Computing Operating System)
An operating system designed for personal computers, providing core system functionality such as file management, application execution, and a desktop environment.

PC OS UI (Personal Computing Operating System User Interface)
The specific visual and interaction layer of a personal computing operating system, encompassing elements such as the desktop, windows, menus, controls, and their aesthetic conventions.

Rendering
The process by which visual elements are computed and drawn to the screen, including text, shapes, images, shadows, and effects, influenced by software techniques and hardware capabilities.

Skeuomorphism
A design approach that imitates real-world objects, materials, or physical behaviors in digital form to make interfaces appear familiar and intuitively understandable.

UI (User Interface)
The point of interaction between a user and a digital system, encompassing visual elements, controls, and interaction patterns.

UI window
A bounded visual container within a graphical user interface that displays content and controls, typically movable, resizable, and layered relative to other windows.

WIMP Paradigm (Windows, Icons, Menus, Pointer)
A foundational graphical interface model based on windows, icons, menus, and a pointing device, shaping most desktop operating system interfaces.

1.4 Scope and Limitations

This paper (and especially the practice-led prototype) adopts a deliberately narrow focus, concentrating exclusively on the standardized window as a core element of OS UIs. The study examines and tries to break down the visual parameters of this component, both historically and through an interactive prototype. Other interface elements (like full application layouts, navigation systems or iconography systems) are not analysed in-depth. Likewise, while broader digital culture, media theory, and socio-technical contexts inform the conceptual framing, they are not the primary object of investigation.

In addition, the scope is limited on OS GUIs , especially of mass-produced GUI machines (e.g. widely available and popular personal computing operating systems).

2. Design and Operating System User Interfaces

OS UIs are foundational environments through which users perceive, navigate, and interpret digital systems. They function not only as technical control layers but as designed visual and interactive worlds that encode assumptions about clarity, efficiency, hierarchy, and modernity. Because OS UIs are repeatedly encountered over long periods, their aesthetic structures become deeply normalized and have shaped collective expectations of what digital environments should look and feel like. Design decisions at this level therefore influence not only usability but also cultural perception of technology and digital aesthetics itself. Research in human:computer interaction demonstrates that interface appearance affects perceived usability and trust, indicating that aesthetics operate as part of functional experience rather than superficial decoration (Tractinsky et al., 2000, p. 139–140).

2.1 Aesthetics in Digital Culture and OS UI

Aesthetics in digital culture emerge in the field of tension between the rules of established visual traditions and the restrictions of computational conditions. Central design principles such as hierarchy, contrast, balance, and rhythm are translated into pixel-based environments shaped by resolution, rendering methods, and interaction constraints. OS interfaces, and especially the OS UI window element, function as persistent frames that structure how digital space is perceived: title bars, content areas, borders, shadows and spacing form a visual language that becomes culturally legible through everyday use.

The evolution of PC OS UI design shows how aesthetic conventions emerge from a combination of technological capability, interaction paradigms, and cultural taste. Across decades, interface styles have oscillated between realism and abstraction, ornament and reduction. These shifts between aesthetics are not simply fashion cycles; they track changes in what hardware can display smoothly and what users have learned to interpret. Styles become markers of technological eras: glossy gradients and dimensionality once signified “advanced” graphics and modern computing, while later minimal interfaces came to signal efficiency, neutrality, and contemporary taste. Aesthetic conventions are not just formal trends but also responses to wider technological and cultural conditions, including what systems can render convincingly and what a culture reads as modern.(Manovich, 2002, p.63–66) In this sense, OS UI aesthetics participates in cultural narratives of progress; often presenting “new” as cleaner, flatter, faster.

2.2 Beauty, Ugliness, Nostalgia and Taste in Digital Media

Beauty and ugliness in digital media operate as evaluative judgments shaped by familiarity, expectation, and norms rather than intrinsic properties of form. In “what is beautiful is usable”, the authors Tractinsky, Katz and

Ikar demonstrate that people frequently infer usability from appearance, meaning that “beautiful” interfaces can feel more usable simply because they align with learned expectations of quality (Tractinsky et al., 2000, p. 139–140).

However, “ugliness” is not the same as dysfunction. A visually dense or deliberately awkward interface may still be effective, and in some design contexts ugliness can be expressive, ironic, or critical; exposing the constructed nature of “good taste.”

Taste itself is (amongst other factors) historically produced. Users learn what looks “right” through exposure to platform conventions, industry trends, and repeated interactions with dominant systems. Nostalgia adds a temporal layer to these judgments: past interface aesthetics can evoke emotional attachment and become desirable precisely because they signal a remembered era of technology use. Visual cues such as bevels, dithering, pixel grids, or obsolete icon metaphors often act as triggers of technological memory.

2.3 Historical Overview of PC OS UI Design

The historical arc of personal computing’s operating system user interfaces (PC OS UIs) begins with foundational research prototypes that shifted interaction away from command-line text toward graphical environments. Early work at Stanford and Xerox PARC in the late 1960s and early 1970s laid the groundwork: They experimented with mouse-driven interaction and integrated the desktop metaphor and pixel-addressable graphics, introducing the now-canonical Windows, Icons, Menus and Pointer (WIMP) paradigm that reoriented how users conceptualize and act within digital space. Although the Alto was never commercialized, its concepts directly influenced later commercial systems and established visual conventions still evident in OS interfaces today (Brock, The Machine That Transformed Computing; Soojung-Kim Pang, Making the Macintosh: The Xerox PARC Visit).

The following rundown is a brief summary of history, grouped into three stages and does not claim to be exhaustive.[1]

2.3.1 Early Graphical User Interfaces and Technical Constraints

The earliest graphical user interfaces were as much technological experiments as design innovations. Xerox’s *Alto* and later *Star* systems in the 1970 - 80s demonstrated that interactive graphical metaphors (icons, overlapping windows, menus and pointers) could make computing more intuitive than command-driven shells, but they were bound by the hardware constraints of the time: limited processing power, monochrome or low-color displays, and rudimentary graphics hardware necessitated minimal visual complexity (fig. 1).

[1] A more extensive rundown of the most influential systems from 1970–2025 was part of my research and can be accessed under: transfer.timorick.ch/practice-led-research/plr_research.jpg

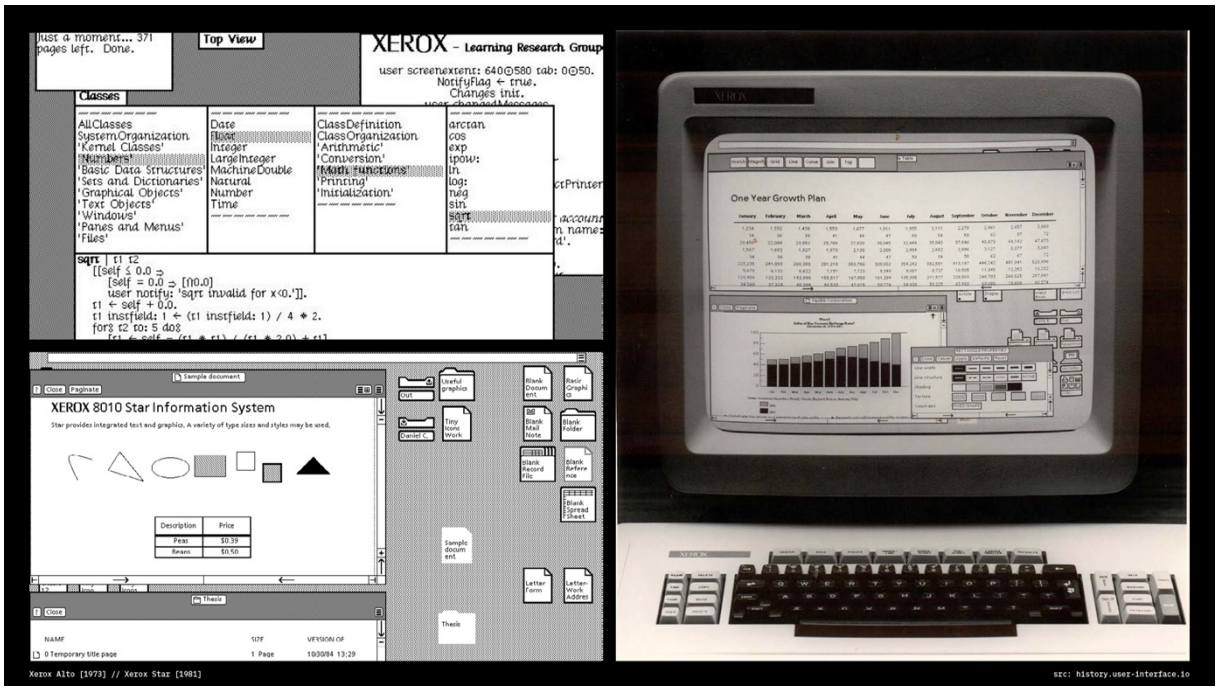

fig. 1: Xerox Alto (1973) & Xerox Star (1981)

Early commercial GUI OSs such as Apple Lisa, Macintosh and Windows 1.0 adapted these paradigms to constrained hardware while prioritizing clarity and quick recognition, using simple shading, bold borders, and restricted palettes to maximize legibility (Computer History Museum, 2023). These constraints shaped an aesthetic where economy of form was a necessity, not a stylistic choice; an aesthetic that later became associated with the “retro computing” look of early desktop environments (fig. 2–4).

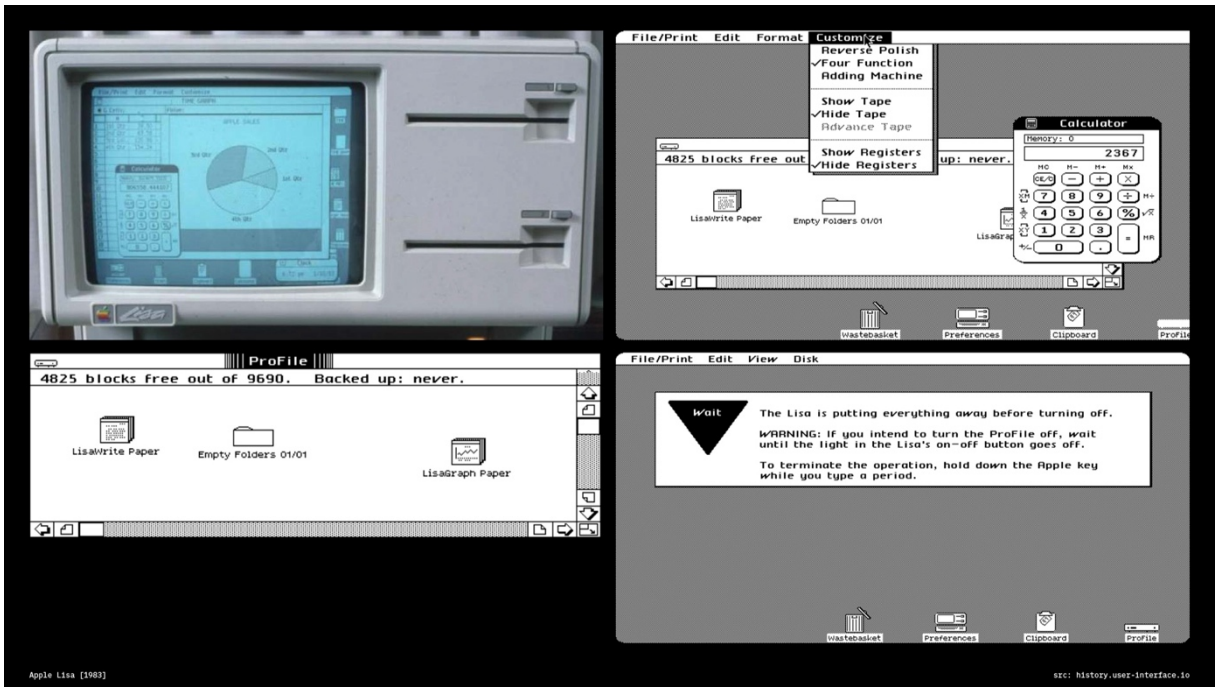

fig. 2: Apple Lisa (1983)

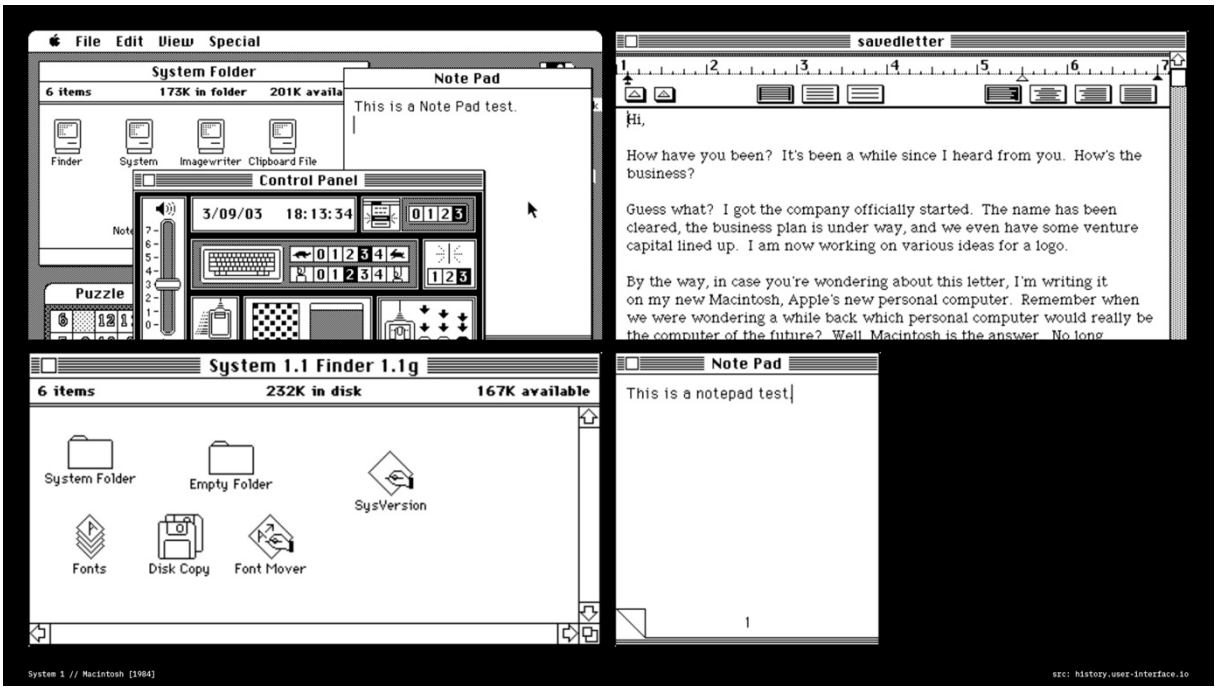

fig. 3: Apple Macintosh (1984)

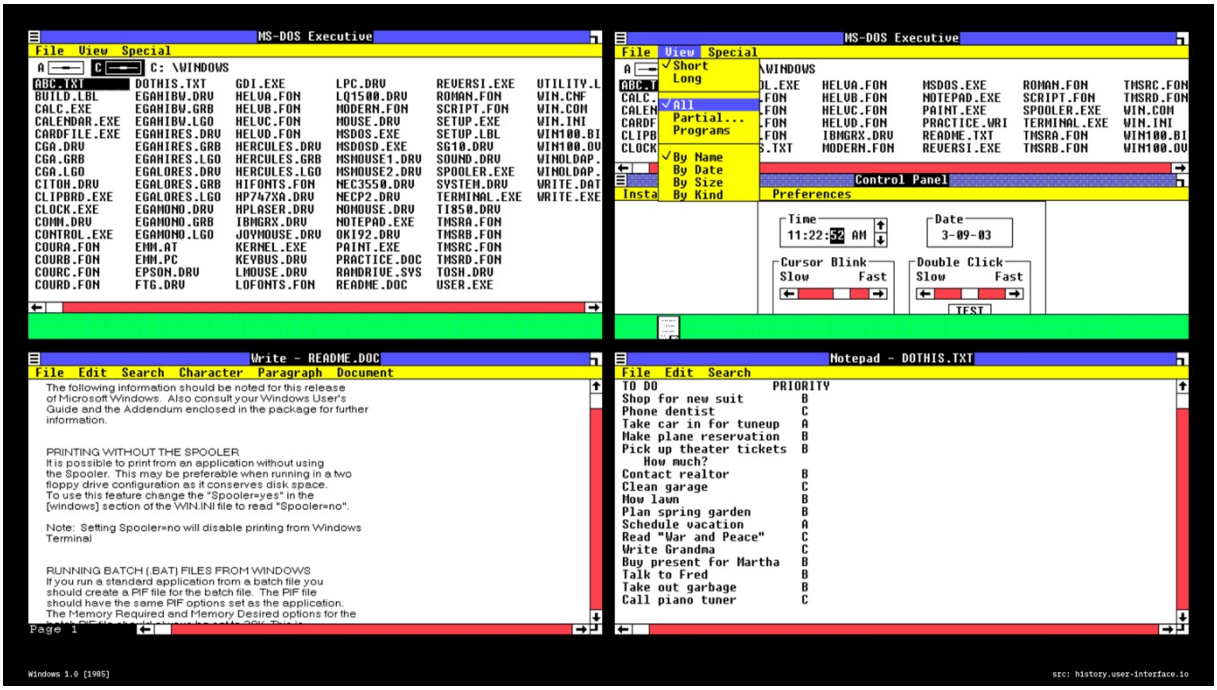

fig. 4: Windows 1.0 (1985)

2.3.2 Skeuomorphism, Material Metaphors, and Depth

As computing hardware became more powerful in the late 1990s and early 2000s, GUI design embraced visual richness and material metaphors intended to make digital environments feel familiar and intuitive. Apple's *Aqua* interface, introduced with Mac OS X in 2001 (fig. 5), exemplified this era by incorporating reflective effects, shadows, and translucency that suggested physical materials, thus grounding abstract digital operations in real-world visual references. The aim of these effects was not mere

decoration but to leverage users' existing visual cognition, using depth and texture to signal interactivity and hierarchy.

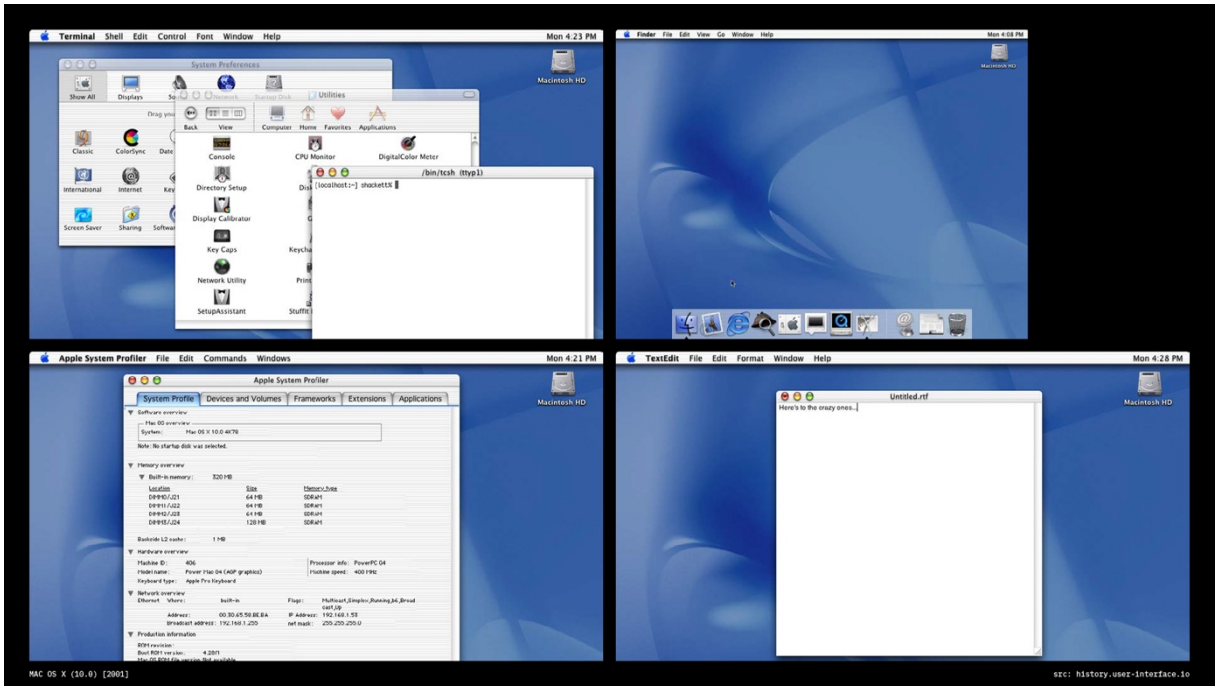

fig. 5: MAC OS X (2001)

This period coincided with similar tendencies in other operating systems and platforms; skeuomorphic icons and detailed visual cues proliferated especially in early smartphone design, with design languages such as iOS 1 and iOS 2 in 2006 often referencing physical objects such as books, notepads, and analog controls to signal functionality. (fig. 6) These material metaphors helped ease transitions for users new to graphical environments by anchoring digital concepts in familiar visual logic. By embedding metaphors of physicality into visual elements, these interfaces sought to make digital interaction more approachable and intuitively readable (Dayam, 2024).

fig. 6: iOS 1 & 2 (2006)

2.3.3 The Shift Toward Minimalism and Flat Design

After 2010, a cultural and technical shift away from heavy material metaphors gave rise to minimalist and flat UI styles that prioritized simplicity, clarity, and performance. This transformation was driven in part by the widespread adoption of mobile and touch interfaces, where limited screen real estate and the need for responsive performance made pared-down visual languages desirable. Microsoft's *Metro* design language emphasized typography, grid-based layouts, and bold color blocks, setting a precedent for flatter, less ornamented OS aesthetics . Around the same time, Apple's iOS 7 (fig. 7) departure from skeuomorphic design signaled a broader industry embrace of flat design principles, underscoring simplicity and interface clarity over detailed metaphors (Chakrabarty, 2025).

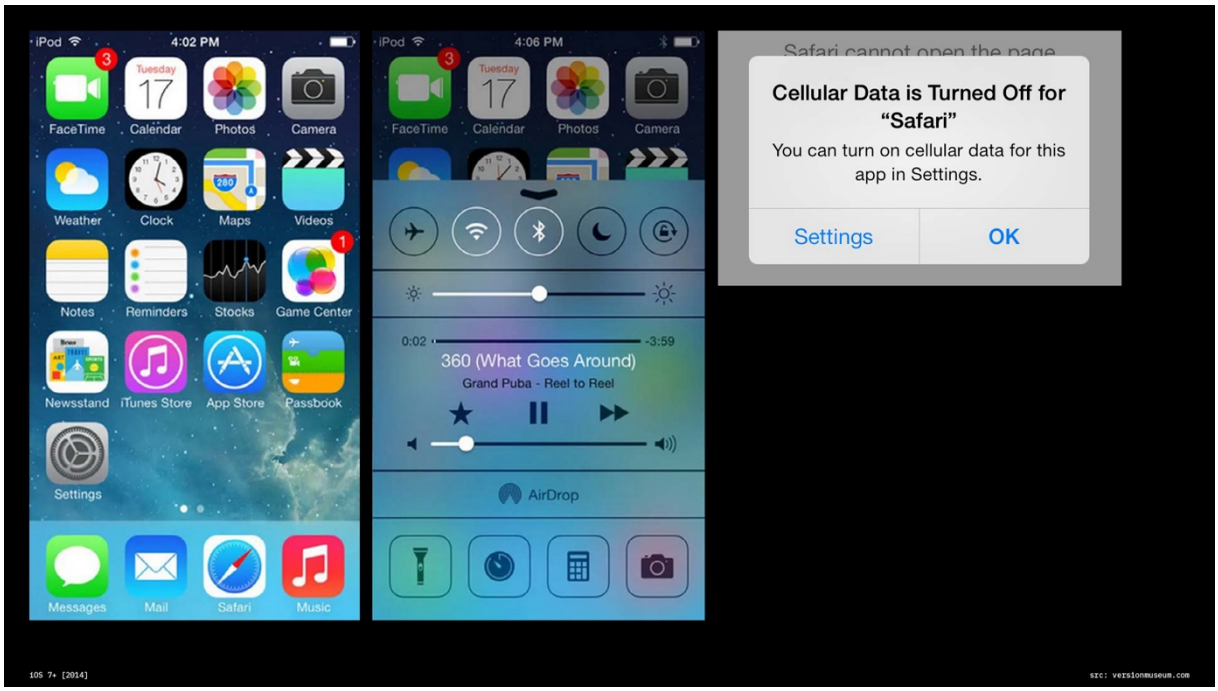

fig. 7: iOS 7 (2014)

2.3.4 Contemporary Trends and Post-Flat Aesthetics

Contemporary OS UI design practice has moved (somewhat) beyond the strict flat minimalism that dominated much of the 2010s and early 2020s, embracing translucency, depth, and dynamic visual layering as core aesthetic strategies. A leading example is Apple's Liquid Glass design language which uses glass-like translucency, real-time light refraction, and soft layered depth to create a richer spatial hierarchy than classic flat interfaces offered. This trend synthesizes the subtle depth cues of earlier "glassmorphic" and translucency trends, as seen in previous macOS and iOS iterations; with modern hardware-driven rendering (Apple Newsroom, 2025) (fig. 8).

fig. 8: macOS Tahoe (2025)

3. Deconstructing the UI Window: Design Parameters

3.1 Defining UI Window Design as a System

UI window design can be understood as a systemic structure that has remained conceptually consistent since its early articulation in the Xerox Alto and subsequent PARC research environments. The fundamental logic of a bounded visual container representing a workspace (framed, layered, and manipulable through pointing or touch) persists across decades of operating systems, from early desktop GUIs to contemporary desktop and mobile platforms. Although aesthetic treatments, interaction details, and navigation models have evolved significantly, the underlying structural model (content enclosed within a defined region, accompanied by controls, hierarchy, and spatial organization) remains stable. This continuity makes it possible to analyse OS UI windows as systems composed of parameters. By treating window design as a parameterized system rather than as a series of unrelated stylistic phases, differences between eras or platforms become legible as variations within a shared structural framework rather than entirely distinct paradigms.

3.2 Visual Parameters

To systematically examine the shared structural logic of UI window design across historical periods and platforms, visual attributes are organized into a set of interrelated parameters. I have grouped these parameters into layers and sublayers to reflect how different visual factors operate at varying levels of perception, from the overall display context to the detailed treatment of interface elements.

3.2.1 Device

An important contextual layer in the perception of graphical user interfaces is the implied or represented device through which the interface is experienced. Even when an interface is shown abstractly, its proportions, curvature, and visual framing often evoke specific hardware conditions, which in turn shape expectations of interaction and visual style. For analytical purposes, this layer is subdivided into aspect ratio and device-specific display characteristics, both of which act as strong visual indicators of technological context.

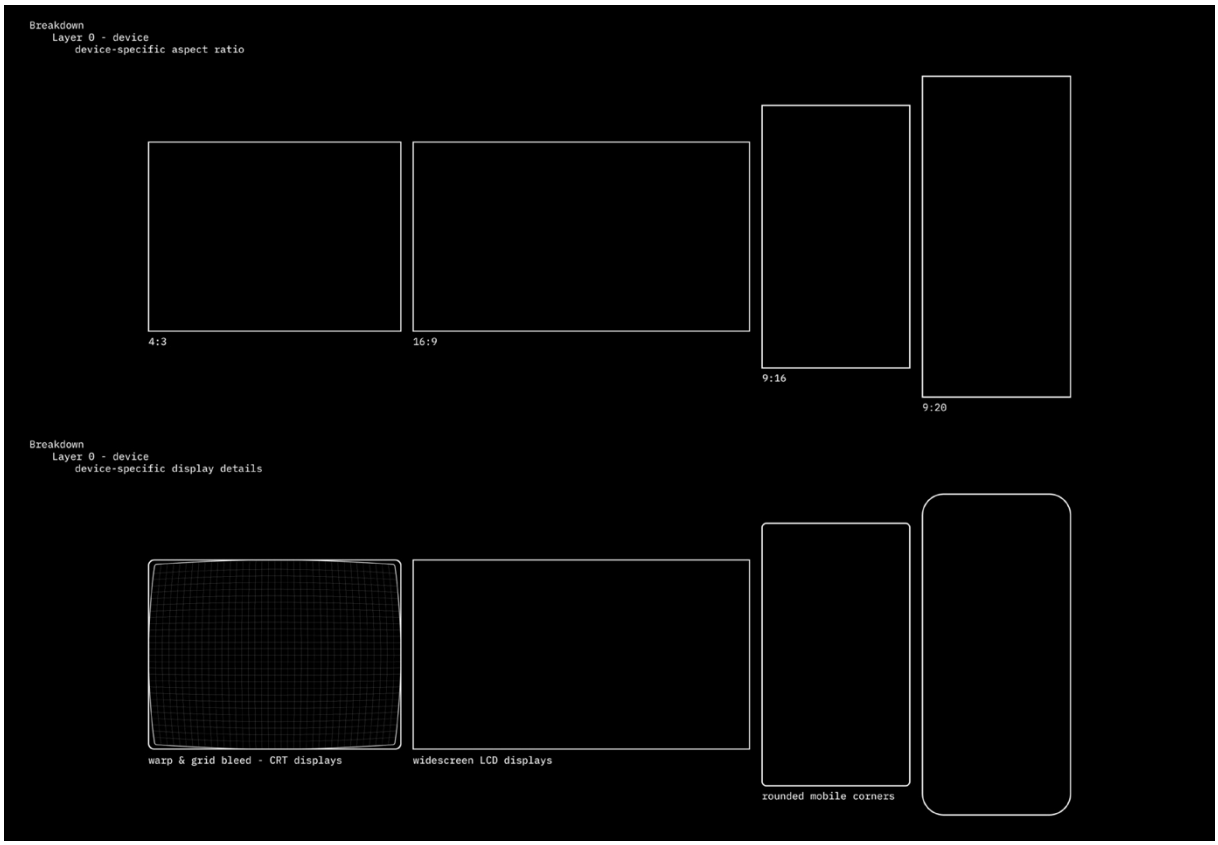

fig. 9: Definition of visual parameters: Layer 0: Device

3.2.1.1. Aspect Ratio

Aspect ratio functions as a key visual signal of technological era and device category. Ratios such as 4:3 are commonly associated with older CRT-based desktop monitors, while widescreen formats such as 16:9 reflect contemporary desktop and laptop displays. In contrast, tall portrait ratios are characteristic of modern smartphones. These proportions affect not only layout possibilities but also the perceived historical and functional context of an interface. As a result, aspect ratio operates as both a structural constraint and an aesthetic parameter, shaping spatial organization and users' assumptions about the device environment.

3.2.1.2. Device-Specific Display details

Beyond overall proportions, specific display characteristics further contribute to the implied device context. Rounded screen corners are commonly associated with contemporary mobile devices, while pronounced edge curvature or visible screen "warp" evokes older CRT displays. Incorporating these parameters into analysis highlights how interface aesthetics extend beyond window design itself to include the visual conditions of the display medium.

3.2.2 Background

Within the prototype, the background functions as a contemporary reinterpretation of the desktop metaphor, representing the broader spatial

and cultural context in which interface windows are situated. Although often treated as a neutral surface, the desktop background carries strong aesthetic and historical connotations that shape how an interface is perceived. Allowing users to define background colors or upload custom images enables exploration of how contextual framing alters aesthetic interpretation. For instance, referencing the characteristic green tones associated with Windows 95 or employing iconic system wallpapers such as the “Bliss” image from Windows XP immediately situates the window within a recognizable historical narrative.

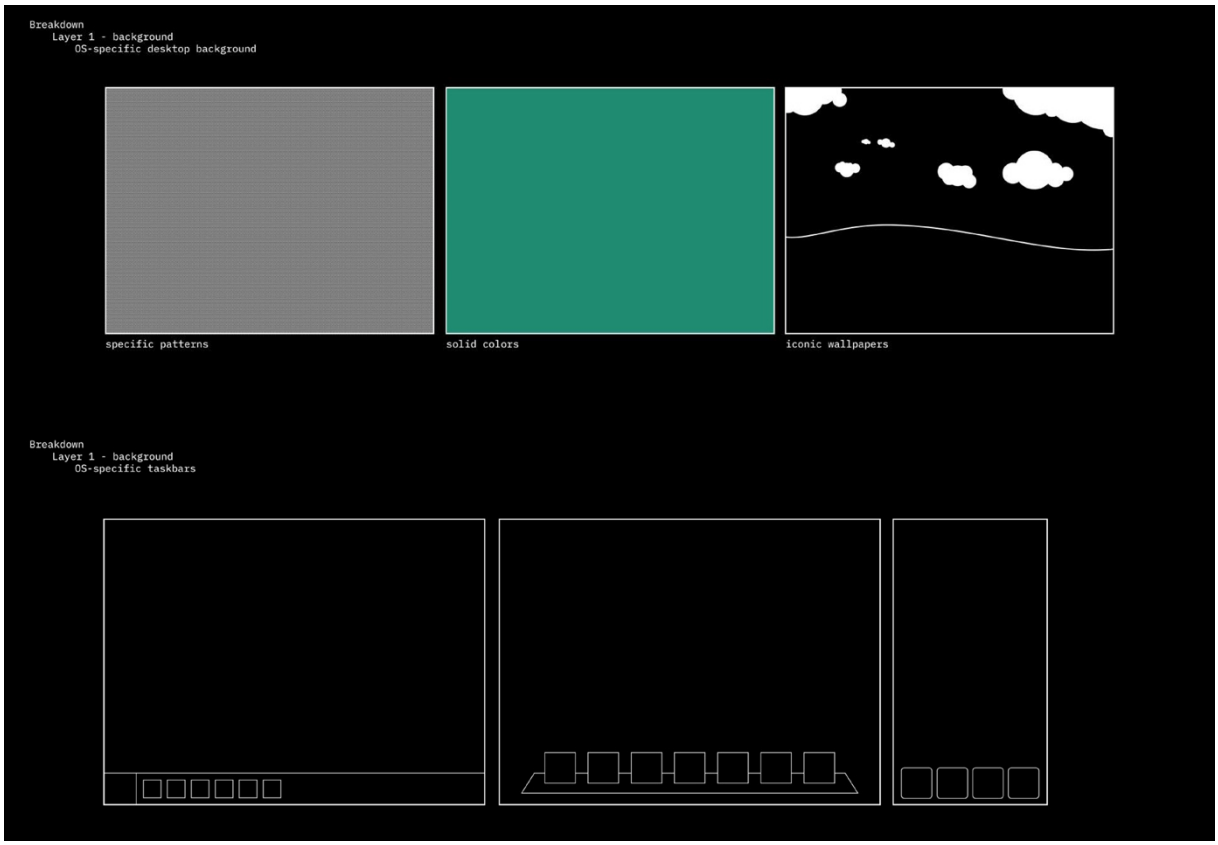

fig. 10: Definition of visual parameters: Layer 1: Background

3.2.3 Window

The window itself is divided into an outer frame and inner content area, both having further subparameters. This distinction allows visual and spatial attributes to be analysed at multiple levels, from the container that defines the window as an object to the layout conditions governing its internal organization (fig 11).

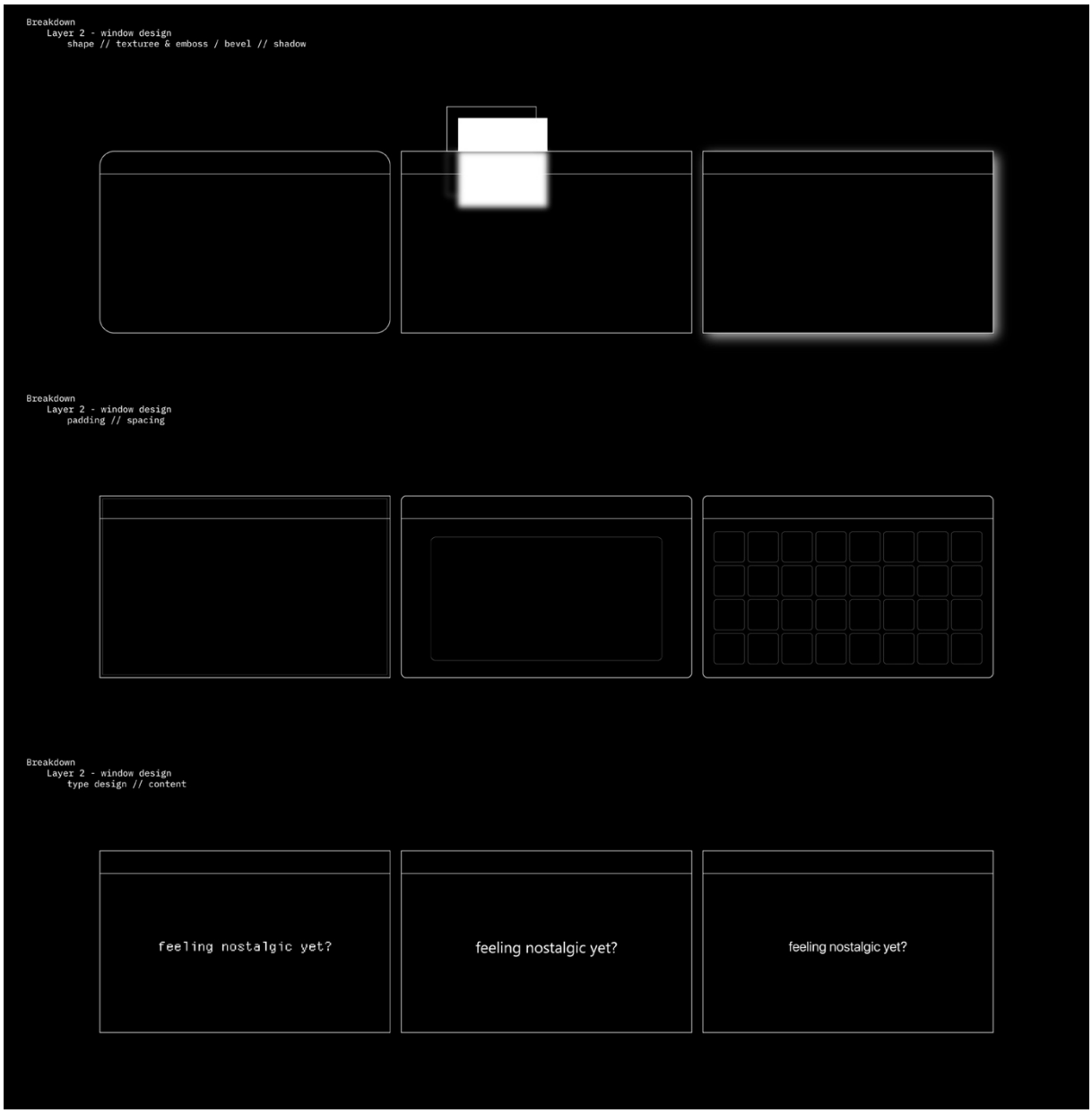

fig. 11: Definition of visual parameters: Layer 2: Window

3.2.3.1 Spacing

Both the outer frame and the inner content area have individual padding and margin which can be set. These variables strongly influence visual density, hierarchy, and perceived modernity: tightly packed layouts often evoke earlier interface paradigms or utilitarian software environments, whereas generous spacing is commonly associated with contemporary minimal aesthetics.

3.2.3.2 Typography

Typography functions simultaneously as an informational and aesthetic parameter. Adjustable properties include font size, color, and alignment, all of which shape legibility, tone, and stylistic identity. Different typographic treatments can evoke specific historical moments or platform

conventions; for example, bitmap-like system fonts suggesting early GUIs or clean sans-serif typefaces associated with modern minimalism.

3.2.3.3 Borders

Border parameters such as thickness, corner radius, color, and opacity can be set individually. Sharp-edged, high-contrast borders may recall earlier desktop interfaces, while subtle or rounded treatments align with contemporary design trends.

3.2.3.4 Window Background

Adjustable properties include color, opacity, and additional effects such as drop shadows or background blur, which can simulate glass-like translucency. These variables are central to expressing different aesthetic paradigms, from opaque, high-contrast surfaces typical of early GUIs to semi-transparent, layered effects associated with contemporary “glass” or material-inspired interface styles.

3.2.4 Header

The header represents a structurally and functionally distinct region of the window, typically housing title information and primary controls. It is therefore treated as a separate layer composed of a main header area and navigation elements (fig. 12).

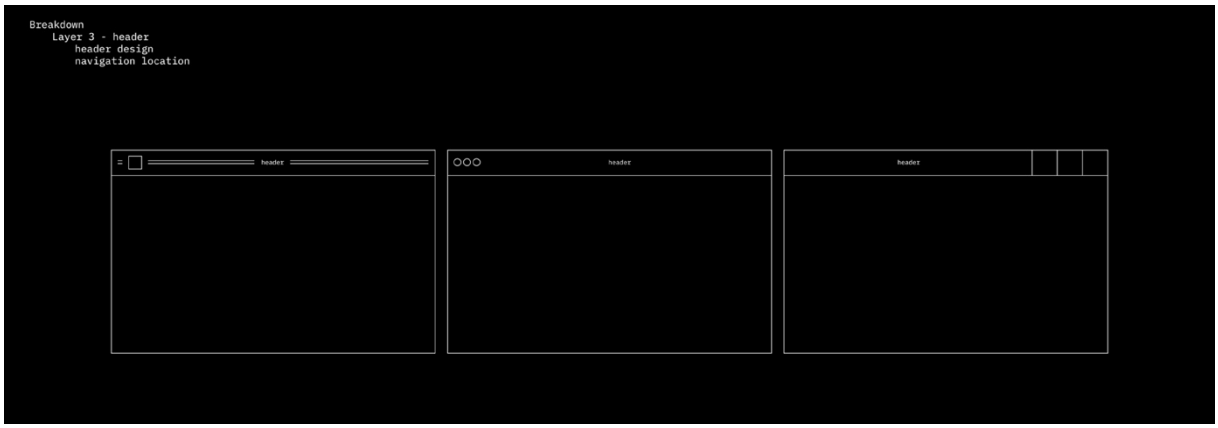

fig. 12: Definition of visual parameters: Layer 3: Header

3.2.4.1 Header Area

The main header area has the same individual parameters as the window body: Typography, background color, borders, and spacing options.

3.2.4.2 Navigation

Navigation elements within the header are treated as individual components with adjustable radius, size, border properties, and background color. These parameters enable recreation of historically distinctive navigation systems, including the rounded red, yellow, and green window controls associated with macOS.

4. The Interactive Prototype

The practical core of this research is an interactive web-based prototype[2] that models a simplified OS UI consisting of a background and a standardized window. Rather than reproducing a specific historical system, the prototype abstracts the window as a modular design object whose visual and formal properties can be manipulated. Through an integrated control interface, users can adjust parameters such as sizes, aspect ratios and proportion, color schemes, typography, border styles, corner radii, and the presence and intensity of drop shadows. These variables correspond to aesthetic characteristics that have historically differentiated OS UI styles. The prototype therefore functions as a controlled environment in which interface aesthetics can be isolated, recombined, and observed outside the functional complexity of full software systems.

By foregrounding visual and structural parameters, the prototype translates interface history into a manipulable design space. Users can reconstruct recognizable historical styles (such as early low-color, high-contrast GUIs, skeuomorphic depth-heavy windows, or flat minimalist configurations) while also producing hybrid or entirely novel combinations. This generative capacity is central to the research aim: to expose aesthetic judgments of “beautiful” and “ugly” as contingent outcomes of parameter relationships rather than inherent properties of specific styles.

4.1 The Interactive Prototype as a Research Instrument

This project adopts a practice-led approach aligned with research through design, where making is not merely a means of illustrating theory but a method of inquiry in its own right. The interactive prototype here operates as a form of knowledge production: by constructing a system that embodies theoretical concepts (such as parameterization, historical referencing, and aesthetic variability) the prototype investigates interface aesthetics through action and iteration.

4.2 System Architecture and Parameter Implementation

I developed the prototype as a hand-coded web application using HTML for structural markup, CSS for visual styling, and JS for interaction logic. The simulated OS window is constructed as a modular DOM structure consisting of nested containers that correspond to the parameters defined earlier in the study (background, window frame, header, content area, navigation elements). Rather than hard-coding visual styles, the system relies extensively on CSS variables, which function as a parameter layer between structure and appearance. Core visual attributes (such as color values, spacing units, border thickness and radius, font size, shadow intensity, opacity, and blur) are defined as variables at the root level, enabling centralized and continuous manipulation of the interface’s visual state.

---

[2] Accessible under practiceledresearch.timorick.ch.

JS provides the mediation layer that connects the user-operated control interface to the visual system. Input elements such as sliders, color pickers, and numeric fields are mapped to specific CSS variables, allowing parameter changes to be reflected in real-time. This architecture supports both discrete adjustments and fluid transitions, making the prototype suitable for exploratory, iterative interaction.

The control interface itself is deliberately designed with minimal visual styling and neutral layout conventions in order to avoid imposing its own aesthetic character onto the experiment. Its functional clarity ensures that it operates as an instrument panel rather than as a competing design object, keeping the analytical focus on the evolving window configurations. The prototype is rounded out with a possibility to download the current iterative design as a snapshot image.

4.3 Limitations

Despite its analytical value, the prototype operates within a deliberately constrained scope. It models only a very simplified, isolated window environment and does not reproduce the full complexity of OS ecosystems, including multitasking behavior, animated feedback or performance constraints. Furthermore, the parameters set (while broad) cannot encompass all historical, technical, or cultural variables that shape interface perception. The abstraction of UI elements into adjustable, web-based properties is heavily simplified. This prototype should therefore be understood as exploratory and iterative rather than universally generalizable across all interface contexts.

5. Conclusion

This research set out to examine how perceptions of beauty, ugliness, and nostalgia in OS UI window design can be explored through interactive parameter manipulation. By tracing the historical development of OS UI aesthetics and deconstructing the window into a system of visual and interactional parameters, the study demonstrated that interface styles often perceived as coherent historical “looks” can be understood as variable configurations within a shared structural framework. The interactive prototype translated this framework into a maluable environment, enabling the recombination of said parameters. Through this process, aesthetic judgments emerged not as fixed properties of particular eras or platforms but as relational effects produced by specific constellations of design variables (fig. 13).

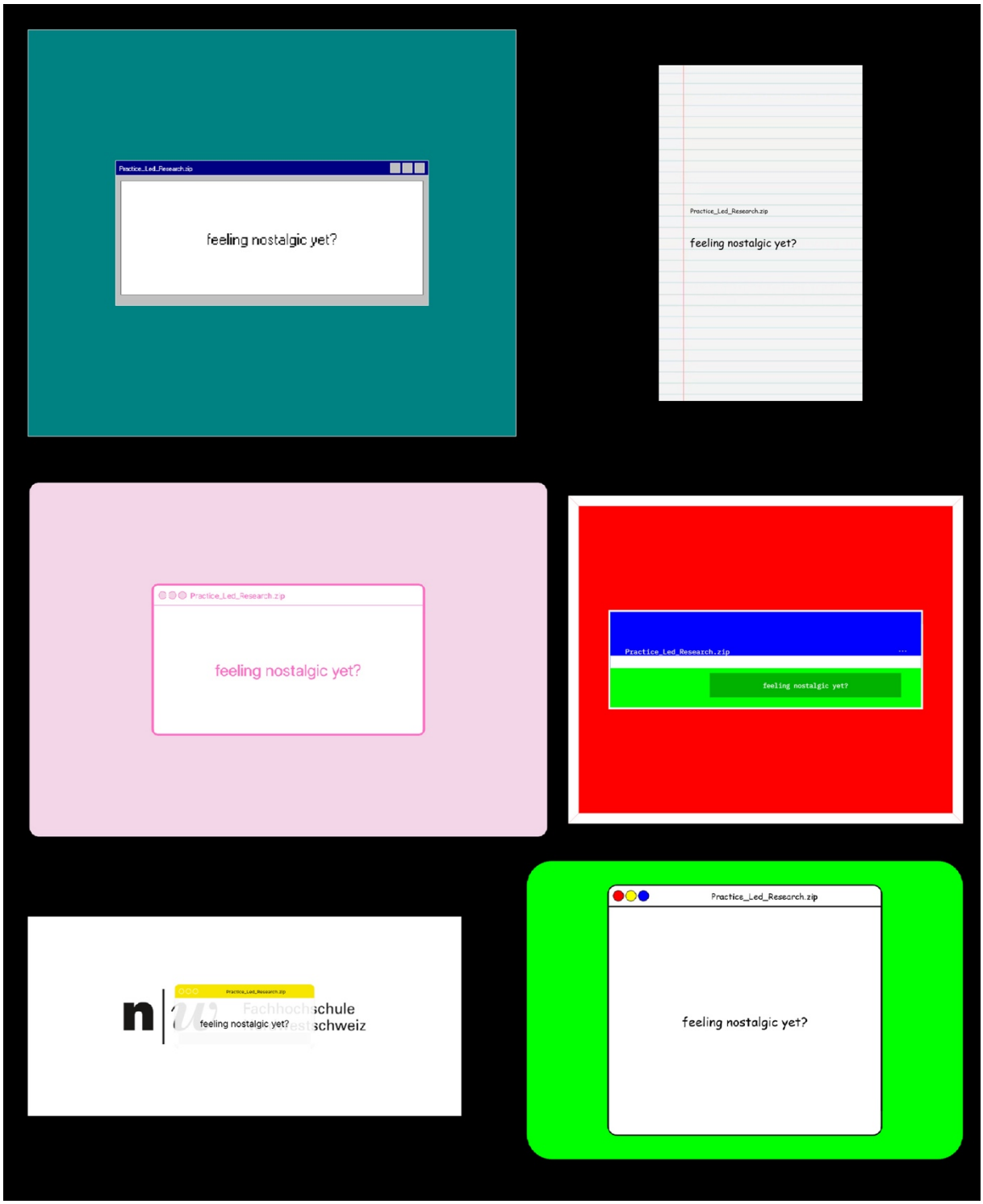

fig. 13: Some experiments with the interactive prototype.

This suggests that categories such as beautiful and ugly in interface design are historically situated and context-dependent rather than absolute. Familiar parameter combinations tend to be read as coherent or appropriate, while unusual or hybrid configurations can appear strange, excessive, or “wrong,” even when they remain structurally functional. Some of the more interesting window designs, challenging both our understanding of nostalgia as well as beautiful, ugly and uncanny interface design,

resulted in the process of playing around without a goal or transitioning from one historical design into the other; such as a window with the iconic navigation system of classic Mac OS GUIs in combination with the previously mentioned “Bliss” Background of Windows XP (fig. 14).

fig. 14: Snapshot with the interactive prototype, in transition of replicating a Windows XP and Mac OS BigSur interface.

At the same time, the prototype revealed how quickly perceptions shift when users recognize historical references or nostalgic cues, underscoring the role of memory and cultural conditioning in aesthetic evaluation.[3] By framing UI windows as parameterized systems, the research bridges design history, aesthetic theory, and hands-on experimentation, offering a way to analyse interface aesthetics beyond trend-based description.

5.1 Reflections on Practice-Led Methodology

The practice-led approach was central to this investigation, as my parameters to analyse historical OS UI got refined through the act of designing, implementing, and interacting with the prototype itself. The methodology highlights the situated and interpretive nature of design research: knowledge produced through prototyping is experiential, contingent, and shaped by the specific system constructed. As such, the practice-led method proved particularly suited to investigating aesthetic perception, where understanding often arises through seeing, comparing, and manipulating forms directly.

[3] A timelapse of the prototype in use can be seen here: transfer.timorick.ch/practice-led-research/plr-timelapse.mp4

5.2 Future Research and Design Directions

Future research could extend this work by broadening both the analytical parameter set and the functional scope of the interactive prototype. While the current system focuses on core visual variables of the standardized UI OS window, subsequent iterations could introduce a greater depth of adjustable options such as additional surface effects like gradients, embossment, inset shadows, and other material simulations. Similarly, introducing more diverse navigation and control element configurations would both allow closer reconstruction of historically specific interaction paradigms, as well as further options to break out of the collectively predefined assumption of what user interfaces must be.

Improvements in usability of the prototype itself also represent an important design direction. Replacing hexadecimal color input with integrated color picker tools would lower the barrier to experimentation, while grouping parameters into more clearly structured categories would establish visual and conceptual hierarchy within the control interface.

Additionally, incorporating ready-made presets that emulate iconic historical interfaces would support comparative exploration, allowing users to toggle between known configurations and experimental hybrids. This could be paired with brief contextual annotations, enabling the prototype to function not only as a generative tool but also as a reflective instrument that connects parameter manipulation with historical design contexts.

Beyond interface refinement, future studies could place greater emphasis on longitudinal and user-centered research. Observing how aesthetic judgments evolve during extended use, and how different user groups respond to historically coded visual cues, would deepen understanding of nostalgia, familiarity, and taste formation.

6. References

6.1 Figures

fig. 1–5: The History of User Interfaces. [Collection of images]. Retrieved October, 2025, from https://history.user-interface.io/

fig. 6–7: Version Museum. [Collection of images]. Retrieved October, 2025, from https://www.versionmuseum.com/

fig. 8: Wikipedia. [Collection of images]. Retrieved October, 2025, from https://en.wikipedia.org/wiki/MacOS_version_history#macOS_26_Tahoe

fig.9–14: Own work

6.2 Bibliography

Apple Newsroom. (2025, June 9). *Apple introduces a delightful and elegant new software design*. https://www.apple.com/in/newsroom/2025/06/apple-introduces-a-delightful-and-elegant-new-software-design/

Brock, D. C. (2023). *The Machine That Transformed Computing*. https://spectrum.ieee.org/xerox-alto

Chakrabarty, P. (2025, January 24). THE EVOLUTION OF UI DESIGN: A DECADE IN REVIEW AND THE ROAD AHEAD. *UX Planet*. https://diptoch.medium.com/the-evolution-of-ui-design-a-decade-in-review-and-the-road-ahead-dca4506e0e0b

Computer History Museum. (2023, January 19). The Lisa: Apple's Most Influential Failure. *CHM*. https://computerhistory.org/blog/the-lisa-apples-most-influential-failure/

Dayam, P. (2024, July 30). *Skeuomorphismus. Als Icons noch eine Geschichte erzählten*. Digitec. https://www.digitec.ch/de/page/skeuomorphismus-als-icons-noch-eine-geschichte-erzaehlten-34124

Manovich, L. (2002). *The Language of New Media*. MIT Press. https://dss-edit.com/plu/Manovich-Lev_The_Language_of_the_New_Media.pdf

Metro (design language). (2026). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Metro_(design_language)&oldid=1333048257

Soojung-Kim Pang, A. (2003). *The Xerox PARC Visit*. Making the Macintosh: Technology and Culture in Silicon Valley. https://web.stanford.edu/dept/SUL/sites/mac/parc.html

Tractinsky, N., Katz, A. S., & Ikar, D. (2000). What is beautiful is usable. *Interacting with Computers*, *13*(2), 127–145. https://doi.org/10.1016/S0953-5438(00)00031-X