Interaction Design and Architecture(s) Journal - IxD&A, N.66, 2025, pp. 66 - 84
DOI: 10.55612/s-5002-066-003

Designing counter-efficient artifacts: critical educational approaches in creative coding

Andrea Benedetti[1] and Tommaso Elli[2]

[1] Web Services and UX Office, Università degli Studi di Milano
[2] Design Department, Politecnico di Milano
tommaso.elli@polimi.it

Abstract. Digital technologies assist us in performing tasks in our daily lives thanks to interfaces and devices that are efficient and user-friendly, but conceal background operations that can jeopardise users' agency and digital well-being. This paper introduces a critical and educational setting to foster designers to make hidden technology features tangible and visible through the development of speculative creative coding projects. Design research provides critical approaches to the socio-technical implications of digital technologies in relation to users' cultures, backgrounds, and digital literacies. By discussing the results of a university course, the contribution identifies seven design strategies to de-emphasise digital efficiency and critically reflect on digital products. While the discovered techniques can support the work of educators and artists, the realisation of speculative projects supports students' critical thinking development. It is especially valuable as similar opportunities are often rare in professional environments.

Keywords: Design Education; Creative Coding; Critical Computing; Digital Design

1 Introduction

The paper presents an investigation using design education to critically challenge dominant paradigms in creating digital products. We focus on the emphasis that is usually put on solving tasks efficiently, making complex infrastructures or processes invisible, and the overall attention to ease of use in user interface and user experience (UI/UX) design. We present the results of a design course that encourages students to ideate speculative prototypes that, through the practice of creative coding, make hidden aspects of digital products more visible. By framing coding as a critical and expressive practice, rather than just a productive task, we analyse how students challenge mainstream interface design assumptions and reflect on digital technologies' socio-technical implications. The result is the identification of seven design strategies to foreground elements and processes concealed in digital products.

Design methodologies have an extensive history of promoting innovation in diverse businesses, especially when new products and services are ideated and developed. Widely adopted frameworks, such as the UX Pyramid, bring attention to

Interaction Design and Architecture(s) Journal - IxD&A, N.66, 2025, pp. 66 - 84
DOI: 10.55612/s-5002-066-003

properties such as functionality, reliability, usability, and pleasurability[1], helping to build objects around the needs of users that can perform tasks without confusion, too many steps, or unclear instructions. The Nielsen Norman Group introduces the concept of efficiency in the context of usability heuristics frameworks, suggesting that good designs balance flexibility for inexperienced users and efficiency for expert ones [1]. UX/UI designers, therefore, act as gatekeepers that decide what is relevant to potential users: they focus on improving the experience (i.e., what is perceptible), but they leave underrepresented or invisible the many processes that are part of the functioning of technologies. This may include technical aspects, but also the labour required to make them work (e.g., content moderation), user profiling, or data collection.

Visibility is a fundamental aspect in this regard. In his graphic novel "Hidden Systems", Nott [2] reflects on how invisibility affects our perception of *cloud computing*, a defining technology of our times. The disappearance of the infrastructural elements that enable cloud computing pushes us to believe that computers have shrunk from the size of rooms to a little mobile device such as the smartphone. While undoubtedly true, the author argues that mobile devices rely on currently the largest existing computational configuration: data centres, which sit far from dense city centres, yet they are constantly accessed by thousands of devices simultaneously. Data centres remain invisible to everyday users of cloud computing, and their invisibility extends to the energy, human labour, and resources they require. For example, the visibility or invisibility of these resources is a crucial aspect of new technologies such as artificial intelligence, which appears difficult to ground in terms of labour, energy and resources required [3–5].

While approachable and easy-to-use digital products are crucial in providing clear use cases of new technologies to the public, the overall obfuscation of their processes may create imbalances in the relationship between users, communities, companies, and policy regulators. A simple example can be found in the attention to mobile devices that potentially listen to conversations without consent, collecting information to formulate recommendations that will later reach users in advertising [6]. These episodes may result in users' anxieties that are not due to ignorance but rather are a consequence of a pervasive and invisible technology over which it is not possible to have full control.

1.1 De-emphasizing the efficiency paradigm of digital product design

In the context of this contribution, we refer to *the efficiency paradigm* as the aim of designing digital products that allow users to carry out tasks efficiently within intuitive interfaces by concealing their technological complexity. As efficiency, invisibility, and ease of use became the dominant qualities in designing digital products, many have tried to rethink this metaphorical connotation, proposing theories and approaches oriented towards finding alternatives. A first approach is the one found in Critical Technical Practice (CTP). Introduced by Philip Agre [7], CTP reconsiders the existing metaphors in technical practice to surface marginal takes on

[1] The UX Pyramid is a conceptual framework that outlines the hierarchy of user needs in product design, drawing inspiration from Maslow's Hierarchy of Needs. It emphasizes that foundational aspects like functionality and reliability must be addressed before higher-level considerations such as aesthetics and meaningfulness. This model has been discussed and elaborated upon by various design professionals and organizations over time; pinpointing a single origin is challenging due to its evolution through various interpretations over time.

technological development and to open new design spaces [7]. Design methodologies help extensively in reconsidering the social implications of technology. Additionally, Critical Design, referring to the theoretical framework of Critical Theory, proposes approaches that use defamiliarisation of the users with design objects [8] to give designers the opportunity to design fictional products that focus on problematic aspects of technological developments.

In the context of UX/UI design, we could apply similar approaches to reflect on the design of digital products and their components (e.g., users, their tasks, the interface to perform them, and the data they produce), re-articulating them to imagine new design paradigms. Judging from many examples that we encounter in everyday life, de-emphasising the efficiency paradigm appears to be already a necessity to preserve the well-being of users and policy applications. It is mostly done through forms of barriers: elements that stop or slow down interaction and that may look like applications or UI patterns to limit access according to usage times or parental controls. For example, "microboundaries" [9] are intentional designs that "provide a small obstacle […] that prevents us rushing from one context to another" [p. 5]. These interventions promote mindful interactions and behavioural change.

These examples represent a push in the design practice, at times supported by important regulatory entities, aimed at changing the value system that currently defines the many digital products that we encounter daily, with the goal of making them more beneficial and healthful to their users. We call them "counter-efficiencies", or modalities of design that prioritise marginal, invisible, inconspicuous aspects that are disregarded when taking into consideration UX guidelines, to promote a more balanced relationship between digital products, users, and their social implications. We argue that design education can be a productive setting to develop these counter-efficiencies without being constrained by the productive requirements of the design industry.

2 Related Works: Creative Coding and Design Education

In this contribution, we propose design education as a site where it's possible to reflect on and de-emphasise the efficiency paradigm thanks to the use of Creative Coding as a critical practice for designers. With "Creative Coding," we refer to the use of programming as an expressive and conceptual medium, often applied in artistic, design, or experimental contexts [10]. While conventional software development focuses on efficiency and utility, creative coding prioritises exploration, aesthetics, performance, and the evaluation of design concepts. Tools commonly used in this context include Processing, p5.js, and openFrameworks, which enable practitioners to directly manipulate visuals, sound, and sensors through code. Common applications of creative coding include using computational thinking to produce generative artwork inspired by nature, physics, and mathematics [11–13]. This sets creative coding apart from block-based or low-code educational platforms like Scratch or App Inventor, which are primarily used to introduce computational logic rather than foster critical reflection or artistic expression.

As an established approach to coding education, Creative Coding is not a new or unexpected addition to design education curricula. Existing courses propose Creative Coding as an expressive medium for both designers and artists, establishing the need for hybrid figures in the landscape of digital arts, design, and new media [10]. As also noted by the "Future of Design Education Working Group," programming became an important asset for design students [11, 14]. In combination with other tools at their disposal (e.g., sketches, prototypes, heuristics, etc.), it increases their confidence when approaching technical matters related to digital products.

2.1 Creative Coding as a Critical and Educational Practice

The critical application of Creative Coding is situated within the broader framework of critical computing, which refers to an approach that interrogates the social, political, and ethical dimensions of computing technologies. As defined by Ko et al. [15], critical computing goes beyond teaching technical skills to foster critical consciousness about how digital systems can reinforce societal inequalities (i.e., racial bias, ableism, and gender discrimination) and how designers and engineers hold responsibility in shaping these outcomes. Rather than accepting technological systems as neutral or inevitable, critical computing encourages reflection on who benefits from technology, who is excluded, and how power is distributed through its design and use. According to Ko et al. [15], education in computing follows a neoliberal argument that urges providing youth with appropriate skills to serve the industry. The model corresponds to a “banking model of education” [16], in which students are seen as accounts to be filled with valuable skills [15]. In alternative to this approach, Freire, one of the most prominent proposers of critical pedagogy, suggests the use of dialogical teaching in which notions are discussed with students to demonstrate the existence of multiple perspectives on a specific matter (e.g., discrimination against black people in relation to face recognition technologies). This process helps individuals who are subjected to oppression to initiate and maintain their struggle towards freedom; they need to recognise that oppression isn't an inescapable reality but rather a confining situation that can be altered. This realisation is crucial, and it must serve as the driving force behind actions aimed at achieving emancipation.

If, conversely, computing is taught to transfer students a fundamental understanding of how our digital world operates and which actors it affects, educators might opt for alternative teaching and evaluation methods that reveal the implications of digital technologies in real-world contexts. Adopting critical perspectives acknowledges computing's immense influence on individual lives, but also prompts questions about how this power is applied and emphasises the responsibility of those in possession of such power [15]. Critical computing education includes a variety of different approaches: it can be delivered in different disciplines that go beyond STEM, like liberal arts [17]; it ranges at multiple educational levels, like K-12 [18, 19] and college [20–22]; it concerns various topics besides pure programming techniques, like AI [23] or the very terminology used in programming [24].

Critical application of creative coding in educational settings appears less common. There are examples that share traits of critical computing and creative coding and that are applied in unexpected conditions and for diverse targets; such examples demonstrate how programming with an expressive goal can be used as a vehicle for critical engagement. While such examples don’t strictly conform to the provided definition of creative coding, their authors still describe them as related to creative coding.

MacDowell et al. organised a ten-hour-long “make-a-thon” targeted at high-school female students to foster in them “maker mindsets” and “identities”. The brief was to leverage the MIT App Inventor[2] and the ZapWorks AR tools[3] to create mobile apps regarding issues that teen girls face in local or global communities [25]. Papavlasopoulou et al. designed a format for groups of children from eight to seventeen years old who are novices to coding. In a four-hour-long after-school

[2] The MIT App Inventor is an environment designed for nontechnical users that simplifies the creation of fully functional smartphones applications. https://appinventor.mit.edu/ (accessed 2024-11-15)

[3] ZapWorks released a series of tools to implement immersive 3D experiences for the web (AR, VR, XR). https://zap.works/ (accessed 2024-11-15)

workshop, they teach them how to develop a simple game coded in Scratch[4]. The workshop is also the occasion for inquiring if differences exist between boys' and girls' coding behaviours using eye-tracking and conducting interviews [26]. Sáez-López et al. run a remote and project-based university course that included students from different countries and briefed them to complete a group project with Scratch. In an educational framework characterised by "critical thinking" and "discovery learning", the course is designed to nurture the development of autonomy in problem-solving, knowledge acquisition and decision-making activities [27]. Dufva presents results of the establishment of a child-friendly hackerspace at the arts and crafts school Robotti (Käsityökoulu Robotti) that, while introducing coding and making to children, transfers critical stances on technology [28].

3 Methodology

Critical creative coding is applied in the course Draw with Code[5] to reflect and design to de-emphasise the efficiency paradigm. The course is designed to teach a comprehensive approach to coding in design-related activities, and it complements digital design courses predominantly focused on web development platforms that enable the design, build, and deploy websites without the need for writing code[6].

The course is elective and open to students enrolled in a Communication Design degree program, typically attended by students in the final year of the Bachelor's or the first year of the Master's level. Such students have a solid background in visual design and are acquainted with hand drawing, technical drawing, and computer-aided visual design (e.g., vector graphics, typographic composition, and 3D modelling). Creative coding is introduced to them as an alternative way to draw in order to capitalise on their visual design proficiency and provide them with some programming knowledge.

The course alternates lectures with individual hands-on activities in which students test their newly acquired skills; additionally, it requires students to work in small groups of 4 to 5 students to experience teamwork and peer learning. Most students have a foundational understanding of visual and interaction design and have been previously introduced to basic programming concepts in other courses. However, their familiarity with coding tends to be superficial, with only a few students possessing more advanced skills due to their personal interest. Taking this into consideration, the faculty has identified P5.js as an appropriate framework to reach the goals of the course: it is a JavaScript library tailored for designers, artists, and those without formal programming training and derived from Processing[7], from which it inherits the approach [29].

---

[4] Scratch is the most used coding enviroment designed for children; it is based on a block interface that allows creating digital stories, games, and animations. https://scratch.mit.edu/ (accessed 2024-11-14)

[5] "Draw with Code" is an elective course erogated at [School name redacted for blind review] as part of the Communication Design curriculum.

[6] For example Webflow: through an intuitive visual interface, users can create complex layouts and animations, while the tool automatically generates the underlying HTML, CSS, and JavaScript code. Additionally, the platform integrates a content management system (CMS) and e-commerce functionality, providing a complete solution for the creation and management of professional-grade websites. https://webflow.com/ (accessed 2024-11-14)

[7] Processing is a coding tool and learning platform that has fostered software and visual literacy in the arts since 2001. https://processing.org/ (accessed 2024-11-29)

The course runs over a three-month timeframe and presents a thematic brief (i.e., a concise statement that outlines the objectives, target audience, constraints, and deliverables for a project) that incorporates a critical or reflective component. Each group of students explores and translates it into a working prototype that forces students to implement their ideas using the capabilities acquired in the course. The group project is organised in three key moments: launch of the exercise, project reviews with the faculty, and the final exam. Reviews happen periodically during the course, with a higher frequency in the last month. Educators act as tutors, helping students develop awareness and development skills in the complex situations faced by students [30]. Once the lectures finish, students work for three to four weeks in their groups to develop a functional prototype to be demonstrated on the final exam date.

3.1 Designing a critical creative coding educational setting

In the latest edition, the faculty employed the course to explicitly challenge students to think critically about the way in which digital products are designed, with the aim of making them more aware of the diverse contexts that can prevent digital wellbeing or possibly cause harm. The theme of the "presence of technology" was used as a means to reflect on the possible implications of an invisible, utterly efficient and easy-to-use technology in people's everyday life. As described by Hallnäs and Redström [31], *presence* provides a starting point that encourages de-emphasising the efficiency paradigm, such as acknowledging the presence of objects even when they are not actively being used. Designing for presence directly counters the idea that computers and their functioning should be invisible, disappearing even when they are being used. When presenting *presence* as a framing for encounters with digital technologies, the authors put invisible computers in tension with an existential angle that reflects on everyday "acts of acceptance" of technologies [31] and inquires into "social technologies" [32]. Rather than concealing technological features, the theme encourages students to make them prominent and challenges the assumption that tools should become invisible through continuous use. During the launch of the exercise, students are prompted with three possible starting points for exploring the concept of presence: (1) involving more than one sense simultaneously through *sonifications* [33], (2) introducing barriers in the interaction through *counter-functional approaches* [34], and (3) requiring effort through *uncomfortable interactions* [35].

The projects developed within this educational setting are analysed to understand how the framework of presence affected the students in finding strategies that prioritise alternative approaches to the efficiency paradigm. Such strategies were identified through a deep analysis of projects from two perspectives: documentation containing the intentions and the communicative goals that students identified, and the interaction flows that they designed. By comparing these two design artifacts, we are able to reconstruct the reasons why certain features are included or excluded from the artefacts.

4 Results

Given the structure of the activities, students were able to explore many digital technologies, showing a strong interest in learning how they function, but also a strong sensibility in finding related issues that they wanted to address. Specifically, during reviews with the faculty, students presented problematic aspects emerging from current configurations of technology using presence as a framework. Some groups focused on the abstractness of infrastructures, others on the inexplicability of results from current machine learning algorithms. Reviews and collective discussion gradually focused students' reflections, allowing them to define the goal around which they would build a prototype.

A total of eleven prototypes were developed during these activities: of these eleven, three were considered inadequate due to their adherence to the theme, and one was not taken into consideration due to its simplicity in conceptualisation and realisation. All prototypes are developed to be featured potentially in art exhibitions or conferences, and they all integrate devices that are available in these spaces. Some prototypes rely on mobile devices and desktop computers connected to a projector, while others rely only on either mobile or desktop devices.

The projects in line with the expectations of the assignment were collected during the exam presentations to be analysed in detail, further tested, and dissected in the technical materials provided: the code written by the students and the accompanying documentation sheet, outlining design choices and challenges (Table 1).

Table 1. List of projects and communication goals identified by students.

Project	Communication goal
N01	To remind the presence of binary code, the basic language of every machine, through an encounters app
Parasite	To set temporal and spatial barriers that condition the experience within the website: users only perceive the presence of others if they are in the same spatial coordinates
Mermaid	To raise awareness of acceptance of the terms and conditions of use we encounter on the web
A Bot of People	To highlight other users' presence and interactions in the places we visit on the internet
Delph•E	To make tangible the presence of technology and profiling algorithms in everyday life
I'S ON U	To make users aware that they are creating data continuously and unconsciously, such as emotions and reactions which can be detected and recorded with and by technology, and this happens every time they interact with the machine
Just a Chat	A chat application in a physical space, in which people are the senders, the servers, the messengers and the receivers.

4.1 Case studies from the course

The following five projects are described in detail because they effectively represent the design strategies outlined in the next section. Each project also addresses one or more critical issues in contemporary digital technologies: (1) the opacity of system operations, (2) the ineffectiveness of consent mechanisms such as terms and conditions banners, and (3) the hidden social dynamics embedded in digital interactions.

Interaction Design and Architecture(s) Journal - IxD&A, N.66, 2025, pp. 66 - 84
DOI: 10.55612/s-5002-066-003

Just a chat. *Just a Chat* (

Fig. 1,

Fig. 2) is a chat application in a physical space, in which people are the senders, the servers, the messengers and the receivers, replicating a process that happens extremely fast and without interruptions. In the words of the students: *"this experience aims at making more understandable […] what happens daily through messaging apps, and across all devices that use client and server interactions. The project wants to recreate this intangible connection by making it tangible [and by making] people move along […] the path that data takes before it reaches its destination."* The project employs the transposition of a purely digital action into a physical interaction, revealing the invisible system behind it.

When starting the experience, participants are asked to join a waiting room where they are randomly assigned roles: one participant will act as the *Server*, while the others will act as *Clients*. The Server will stay in the centre with all Clients around it. After one Client has finished writing their message, it's packaged into a fake colour-coded enclosure, and the Server needs to collect the message by scanning the colour-coded package with their camera. Finally, to see the message in the group chat, all other Clients need to monitor the package with their camera from the Server: only then will the message appear in a traditional group chat interface.

Fig. 1. Setting of "Just a Chat". Participants in the experience can be clients (multiple) or servers (only one in each round). They use their mobile devices to enter a chat application, and they have to physically send messages to the server that delivers them to the recipient. Source: authors.

DOI: 10.55612/s-5002-066-003

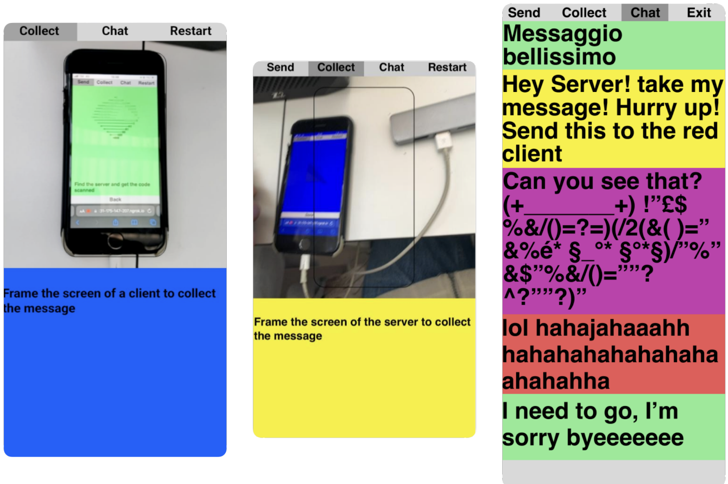

Fig. 2. Images of the UI of the prototype for "Just a Chat". Messages are delivered and collected thanks to visual codes that appear on mobile screens.

Mermaid. *Mermaid* (

Fig. 3,

Fig. 4) prompts users to interact without much thought with a Terms and Conditions pop-up that asks for access to the microphone and camera of a mobile device in a very long text. The goal of the projects was *"making visible what we cannot grasp with passive acceptance of Terms and Conditions to access any kind of online service."* With Mermaid, students want to *"make users think about the personal information they grant access to every day."*

The prototype is designed to be displayed in an exhibition, where a large canvas with a spinning vortex of colored particles is running. Users access a mobile website on their smartphone where they are prompted to very hastily accept the terms and conditions of a fictitious service, "Mermaid". After accepting, they see a graphical representation of their front-facing camera as a matrix of dots, and each frame captured by the camera is sucked into a vortex and sent to the larger screen canvas. At this moment, their camera and microphone feeds are unwillingly shared with all the other people interacting with the prototype.

After a first unaware interaction, as the users agree to the Terms of Service without probably reading them to their full extent, the microphone feed is recorded along with the feed of other participants, and the camera feed is visualised as colored particles on a shared canvas.

DOI: 10.55612/s-5002-066-003

Fig. 3. A mock-up used to present "Mermaid". The image displays the vortex of consents and personal information that are collected via the passive acceptance of websites' terms and conditions, which are too long and complex to be properly understood.

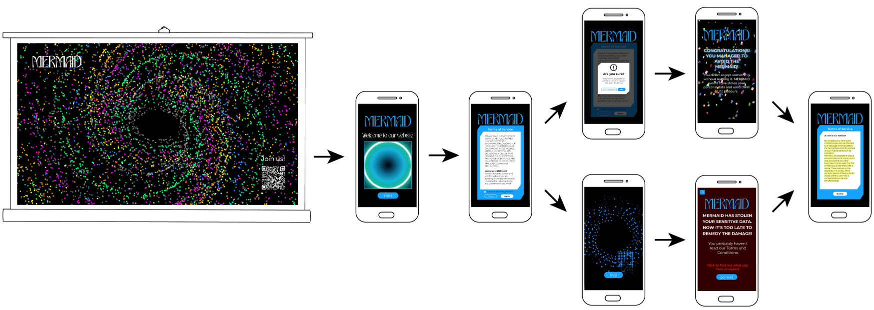

Fig. 4. Diagram of the interaction flow of the "Mermaid" prototype.

A bot of people. *A bot of people* (Fig. 5) focuses on the intimate experience we usually have with online websites, and it speculates on what would happen if it were less intimate than it currently is. In this case, students focused on *"showing the previous interactions of other users that are usually hidden, trying to bring a digital space where the illusion of the presence of others is visible and affecting your own experience."*
The user accesses a short narrative experience, where they are tasked to demonstrate that they are humans by answering visual prompts akin to CAPTCHAs. As they answer a series of simple questions, they can see other people doing the same, without knowing if they are real people or not. After the sequence of questions finishes, the user relives the experience as a passive observer, this time looking at the trace of their mouse movements.

The prototype collects the position of the users' mouse cursor on the screen and stores it in an online database. In this process of sedimentation, the actions performed by other users are positioned at the centre of the narrative, instead of being hidden as within traditional interfaces.

DOI: 10.55612/s-5002-066-003

Fig. 5. In the prototype "A bot of people", users can navigate a digital survey and be aware of the presence of other users who took the same path, thanks to the representation of the movement of their cursors.

N01. *N01* (Fig. 6,
Fig. 7) aims to make binary code, *"the basic language of every machine"*, visibly and experientially present. The project connects people through individual bits of information to highlight how the complexity of digital communication emerges from a simple system of 0s and 1s. It reflects on the richness of outcomes that can be generated from minimal inputs.

The experience is designed for an art exhibition setting and features a playful, nostalgic aesthetic inspired by old love calculator websites. Users begin by logging into the web-based interface with a name or nickname, which personalises their interaction and saves the final result in an online gallery. Participants exchange binary messages by tapping a heart icon, which triggers sounds on the receiving device. When they meet in person, they scan each other's screens to establish a connection and complete the interaction. While the interface is lighthearted and humorous, it prompts reflection on how invisible digital processes are rooted in basic binary logic.

Interaction Design and Architecture(s) Journal - IxD&A, N.66, 2025, pp. 66 - 84
DOI: 10.55612/s-5002-066-003

Fig. 6. The home screen of N01, a digital installation that transforms user messages into binary code, expressed through synchronised sound and visuals to reveal the hidden language of machines.

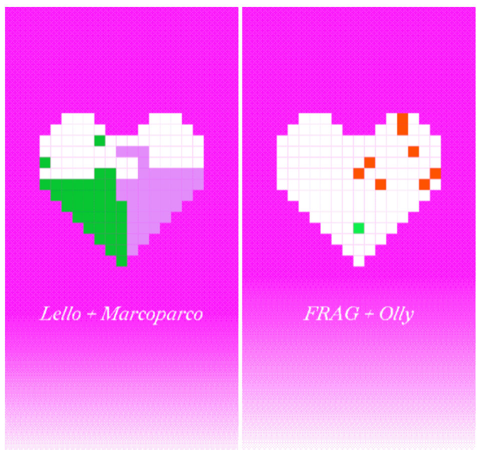

Fig. 7. The data provided by both users is stored and used to generate a digital artwork: a pixelated heart split into two halves, one representing each participant. The heart's colour corresponds to the value assigned by the server during the session, while the name displayed is taken from the user's initial input on the website.

Delph-E. *Delph·E* (Fig. 8, Fig. 9) explores the role of profiling algorithms by simulating a digital oracle and inviting reflection on whether algorithms have become the modern equivalent of the Oracle of Delphi. The project makes tangible the presence of surveillance technologies and the construction of user identity through data. Within the experience, users interact with a website that collects both explicit and implicit data: facial expressions are analysed in real time, while subtler metrics—such as time spent watching animated GIFs—are tracked without the user's awareness. These inputs are then used to generate stereotypical and exaggerated representations of the user, parodying how platforms simplify identity to deliver tailored recommendations. By intentionally presenting the profiling as crude and overgeneralized, *Delph·E* exposes how minimal behavioural cues can lead to distorted identity models, making visible the hidden mechanics of data-driven personalisation.

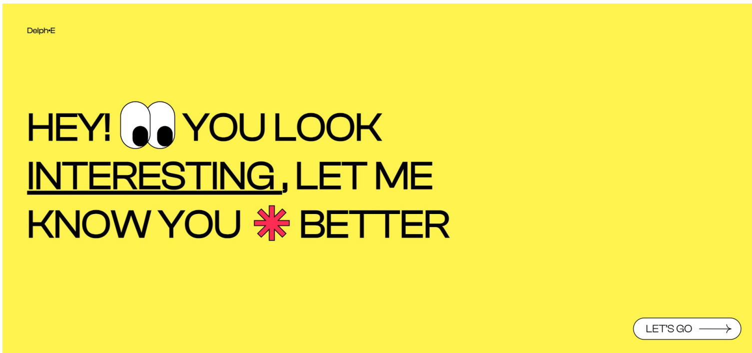

Fig. 8. Delph•E is a speculative web experience that exposes how algorithms profile users by turning subtle behaviours into exaggerated, stereotypical identities.

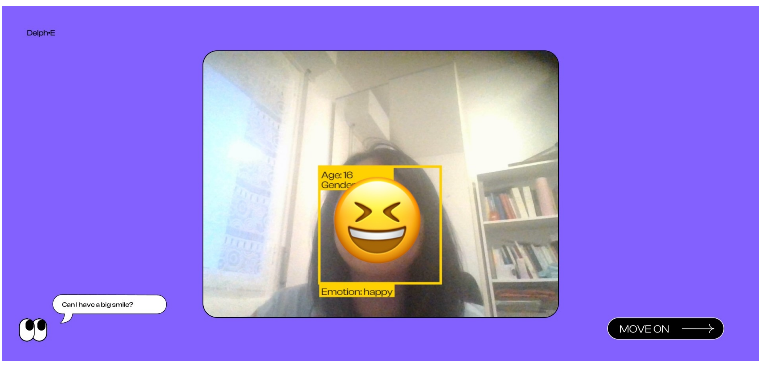

Fig. 9. Delph·E profiles users by analysing facial expressions and interaction patterns, such as time spent viewing content, to infer preferences and construct a simplified identity.

4.2 Techniques to design for presence

From the analysis of students' projects, it's possible to outline seven different techniques that are used by students to design for presence (Table 2).

Table 2. Techniques to design for presence

Technique	Description
Unaware interaction	Designers reuse a "dark pattern" to create estrangement in the use of a technology and to create a context of use that totally changes meaning. If a dark pattern is used to exploit the inattention of the user of a technology, in this case the purpose is to show the consequences

Interaction Design and Architecture(s) Journal - IxD&A, N.66, 2025, pp. 66 - 84
DOI: 10.55612/s-5002-066-003

	of an action taken, sometimes unconsciously.
Sedimentation	Designers collect and expose other users' interactions with a digital product. Through this strategy, data collected by other users are exposed and become visible, creating a communal sense of the use of a product. Sedimentation becomes an "additional layer" in the interaction flow.
Physical interaction	Designers create a physical relationship between the technology and its use, requiring actions to be taken from a digital to a physical space.
Visualization	Designers use information and data collected by a specific product as input to create generative artwork. With this strategy, information gathered through technologies becomes tangible parameters detached from their utility (their value).
Glitches	Designer construct systems of interactions aimed at errors. With this strategy, errors are contemplated as a desired result of using a digital product. Specifically, students use old, crude, and unsophisticated technologies and embrace their technological limitations that produce coarse results.
Sonification	Like *Visualization*, designers use sonification to introduce a multisensory dimension to the enjoyment and use of technology.
Unveiling	Through an interactive approach, designers explain how a specific digital product functions by embedding the explanation in the product encounter.

Techniques' usages are not mutually exclusive. Cases are encountered that mix them to achieve a complex communication goal (Table 3). For example, "N01" layers *physical interactions*, *sonification*, and *visualisation* to interpret expressively the theme of binary code. Again, "Mermaid" uses *sedimentation*, *unaware interaction* and *visualisation* to reflect on the role of long pop-up menus that contain crucial information to use services and websites. However, effective communication does not necessarily require multiple techniques: "Just a chat" uses just one technique and gets its point across successfully.

Table 3. Emerging techniques and their distribution in prototypes.

Project	Communication goal
N01	Physical interaction – Sonification – Visualization
Parasite	Physical interaction – Sedimentation – Visualization
Mermaid	Sedimentation – Unaware interaction – Visualization
A Bot of People	Sedimentation – Visualization
Delph-e	Unaware interaction – Glitches
I'S ON U	Unaware interaction
Just a Chat	Physical interaction – Unveiling

Interaction Design and Architecture(s) Journal - IxD&A, N.66, 2025, pp. 66 - 84
DOI: 10.55612/s-5002-066-003

5 Discussion

The prototypes developed within the scope of the course underline the effectiveness of framing Creative Coding as a Critical Computing learning framework. Analysing the prototypes, students demonstrate proper understanding of technical notions along with the capacity of employing them to communicate their own complex point of view on the matter. If we re-examine principles of critical pedagogy, the "Draw With Code" setting allows students to consider hidden or unexplored implications of digital technologies that can hamper digital wellbeing, through conversations with the faculty.

Compared to other related coursework, "Draw with code" uses Creative Coding as a critical practice rather than an artistic one. The difference lies in employing technical notions to communicate reflections to a larger public, rather than using them to research exclusively their aesthetic value. The brief is crucial to encourage this type of activity and to provide students a clear yet open target to reach (

Table 4). The brief on the theme of "presence" provided a first robust starting point for students to reflect on the digital technologies that they might design in the future. The techniques developed by students in response to the theme reveal a deep understanding of the socio-technical implications of digital technologies.

Table 4. The course "Draw with Code" is framed as a critical educational setting.

Format	Participants	Goal	Brief	Tools
A three-month long course structured in weekly and interactive lectures, individual assignments and group projects.	University students attending a degree communication design, with interest in digital products.	Teach basic programming to design students; frame coding as a creative practice akin to drawing; use coding as a critical practice to inquire digital technologies	Design and develop a prototype that helps in understanding technology and its components, rendering it more "visible"	P5.js

The techniques developed by students can be meaningfully contextualised within the broader design and HCI literature. For instance, the use of *unaware interaction* mirrors, but subverts, what Gray et al. [36, 37] and Mathur et al. [38] define as "dark patterns," transforming them from deceptive into revelatory mechanisms that can be exploited by designers to show the consequences of inattention. Similarly, *physical interaction* recalls the "design frictions" proposed by Benford et al. [35] and Cox et al.[9, 39], where intentional obstacles promote reflection or behavioural change. The techniques of *sedimentation*, *sonification* and *visualisation* align with Hallnäs and Redström's idea of "soniture" and "informative art" part of the slow technology conceptual framework [38], where aesthetic and sensory engagement prompt users to notice hidden system behaviours. Finally, the *unveiling* technique relates to the goals of critical pedagogy and transparency in computing education as described by Ko et al. [20] and Freire [21]. If we compare these techniques to the classical notion of efficiency, which hides unnecessary complexities and speeds up interaction when needed, these techniques define new counter-efficiencies that communicate previously invisible aspects of digital technologies. These counter-efficiencies directly disregard conventional notions found in Human-Computer Interaction: while the pyramid sits on functionality and aspires to pleasure, counter-efficiencies prioritise time-consuming ways of interacting with prototypes in favour of a chance to explain to users how digital products work.

Additionally, the development of prototypes provides an opportunity to tackle the learning of complex technologies through an experiential approach that has proven to be effective in design curricula. Learning by doing [40] is a recurring theme that can be found in many educational contexts, including less formal ones where hacking is the primary form of engagement with complex technologies [41, 42]. In the context of design education, students can explore the inner workings of digital technologies and develop the skills to communicate them to non-experts. These explorations result in an increase of technological and digital literacy for students and the public alike, demystifying the opacity of most technologies, which conceal human, technical and environmental costs [5, 43].

Finally, a developed sense of awareness of the socio-technical implications of digital technologies resulted in the development of techniques useful to address common issues related to digital wellbeing: privacy concerns, transparency, and the social nature of digital technologies. Counter-efficiencies can prioritize aspects that reveal whenever data is being collected, and to which use: “Mermaid” could be repurposed in digital applications as a form of alternative to long Terms and Conditions of Service; “Just a chat” could be repurposed to disclose how messaging protocols work, and “A bot of people” could be repurposed in scenarios where seclusion due to digital technologies is enhanced.

The discovered techniques can be meaningfully applied in other creative coding settings, supporting the work of educators and artists. From the perspective of the design industry, the experience of conducting speculative projects in a creative coding course is highly valuable because it supports the development of both critical thinking and a deep understanding of technology, transforming students into conscious and skilled practitioners. This experience is especially meaningful, as similar opportunities are often rare in typical professional environments.

6 Conclusion

This contribution presents “Draw with code”, a Creative Coding class in the design curriculum of [Redacted for blind review], as a critical pedagogical framework that aims at reflecting upon the prioritisation of efficiency, ease-of-use and invisibility related to the introduction of digital technologies. Borrowing from Critical Computing, Science and Technology Studies, and Critical Design, the course provides “presence” as a framework to define counter-efficiencies to reflect on technology, the design of digital products and those aspects that can affect users’ digital well-being.

Thanks to seven prototypes developed by students, we identify seven different techniques – mixed between each other in each prototype – that provide an outlook on how to frame digital technologies as “present” when they are most of the time conceived to be invisible. Designed to engage with the public in exhibition spaces, these prototypes have proven to be a valuable design challenge for students. First, they had to become comfortable with a specific digital technology of their choice (such as messaging apps, web sockets, or legal notes), and then they developed an effective strategy to engage with non-experts. These techniques take inspiration from the literature, such as counter-functionalities and uncomfortable interactions, while incorporating some novel aspects, such as repurposing dark patterns and the sedimentation of interactions from other users of the same prototype.

While Creative Coding has already proven its ability to improve confidence and fluency with programming, we argue that it can do the same with technological and digital literacy, both in design students and the public that engages with these artefacts. These techniques, which are currently highly speculative, may have repercussions in the commercial development of applications. If integrated in existing products, they may prioritise usually hidden aspects of digital technologies, such as privacy, transparency, or their social dimension, opening new outlooks on digital wellbeing.

Acknowledgements. The authors would like to express their deepest gratitude to the students who took part in the course, dedicating time, creativity, and commitment to learning coding as communication designers. Their engagement was fundamental to the development of this research. A special thank you goes to Dr. Michele Mauri, coordinator of the course, for his guidance, supervision, and for making the publication of this work possible.

Credits for the mentioned projects. N01: Davide Agostinelli, Elisa De Benedettis, Martina Esposito, and Giulia Gnessi. Parasite: Maria Cecilia Buonocunto, Deborah Franceschini, Stefano Gubiolo, Matteo Visini, and Giulia Zago. Mermaid: Clara Di Bella, Irene Fazzari, Alessandra Palombelli, and Elisa Tudisco. A Bot of People: Alessandro Raimondo, Anna Maria Spada, Davide Saladino, and Bianca Maria Lucini. Delph•E: Elisabetta Como, Paola Pia Palumbo, Benedetta Riccio, Zeya Wu. I'S ON U: Andrea Corsini, Beatrice Ulivi, Sabrina Morelli, Simone Cerea, Yousef Taffal. Just a Chat: Federico Bobbo, Francesco Bonetti, Sarah Cosentino, Matteo dell'Agostino, and Enrico Isidori.

CRediT author statement.
Andrea Benedetti: Conceptualisation, Methodology, Investigation, Resources, Data curation, Writing – original draft, Writing – review and editing. **Tommaso Elli**: Conceptualisation, Resources, Data curation, Writing – original draft, Writing – review and editing.

References

1. Nielsen, J.: Enhancing the explanatory power of usability heuristics. In: Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94. pp. 152–158. ACM Press, Boston, Massachusetts, United States (1994). https://doi.org/10.1145/191666.191729.
2. Nott, D.: Hidden systems: water, electricity, the internet, and the secrets we use every day. Random House Graphic, New York (2023).
3. Tubaro, P., Casilli, A.A., Coville, M.: The trainer, the verifier, the imitator: Three ways in which human platform workers support artificial intelligence. Big Data & Society. 7, 205395172091977 (2020). https://doi.org/10.1177/2053951720919776.
4. Newlands, G.: Lifting the curtain: Strategic visibility of human labour in AI-as-a-Service. Big Data & Society. 8, (2021). https://doi.org/10.1177/20539517211016026.
5. Crawford, K.: Atlas of AI: power, politics, and the planetary costs of artificial intelligence. Yale University Press, New Haven (2021).
6. Kröger, J.L., Raschke, P.: Is My Phone Listening in? On the Feasibility and Detectability of Mobile Eavesdropping. In: Foley, S.N. (ed.) Data and Applications Security and Privacy XXXIII. pp. 102–120. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-22479-0_6.
7. Agre, P.E.: Toward a Critical Technical Practice: Lessons Learned in Trying to Reform AI. In: Social Science, Technical Systems, and Cooperative Work. Psychology Press (1998).
8. Bardzell, J., Bardzell, S.: What is "critical" about critical design? In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 3297–3306. ACM, Paris France (2013). https://doi.org/10.1145/2470654.2466451.
9. Cox, A.L., Gould, S.J.J., Cecchinato, M.E., Iacovides, I., Renfree, I.: Design Frictions for Mindful Interactions: The Case for Microboundaries. In: Proceedings of the 2016 CHI

Conference Extended Abstracts on Human Factors in Computing Systems. pp. 1389–1397. ACM, San Jose California USA (2016). https://doi.org/10.1145/2851581.2892410.
10. Maeda, J.: Creative code. Thames & Hudson, New York, N.Y (2004).
11. Benedetti, A., Elli, T., Mauri, M.: "Drawing With Code": the Experience of Teaching Creative Coding as a Skill for Communication Designers. In: 12th International Conference on Education and New Learning Technologies. pp. 3478–3488, Online Conference (2020). https://doi.org/10.21125/edulearn.2020.0982.
12. Shiffman, D.: The nature of code. Selbstverl., s.l. (2012).
13. Bohnacker, H., Gross, B., Laub, J., Lazzeroni, C., Frohling, M.: Generative design: visualize, program, and create with JavaScript in p5.js. Princeton Architectural Press, New York (2018).
14. Offenhuber, D., Mountford, J.: Reconsidering Representation in College Design Curricula. She Ji: The Journal of Design, Economics, and Innovation. 9, 264–282 (2023). https://doi.org/10.1016/j.sheji.2023.04.005.
15. Ko, A.J., Beitlers, A., Wortzman, B., Davidson, M., Oleson, A., Kirdani-Ryan, M., Druga, S., Eversone, J.: Critically Conscious Computing: Methods for Secondary Education. (2024).
16. Freire, P.: Pedagogy of Freedom: Ethics, Democracy, and Civic Courage. Rowman & Littlefield (1998).
17. Guzdial, M.: Designing Computing Education Courses for Liberal Arts and Science Students. In: Proceedings of the 27th ACM Conference on on Innovation and Technology in Computer Science Education Vol. 2. p. 614. Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3502717.3532129.
18. Kafai, Y., Jayathirtha, G., Shaw, M., Morales-Navarro, L.: CodeQuilt: Designing an Hour of Code Activity for Creative and Critical Engagement with Computing. In: Proceedings of the 20th Annual ACM Interaction Design and Children Conference. pp. 573–576. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3459990.3465187.
19. Kafai, Y., Proctor, C., Lui, D.: From Theory Bias to Theory Dialogue: Embracing Cognitive, Situated, and Critical Framings of Computational Thinking in K-12 CS Education. In: Proceedings of the 2019 ACM Conference on International Computing Education Research. pp. 101–109. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3291279.3339400.
20. Lee, C., Soep, E.: Beyond Coding: Using Critical Computational Literacy to Transform Tech. (2018). https://doi.org/10.15781/T24J0BF37.
21. Lee, C., Soep, E.: None But Ourselves Can Free Our Minds: Critical Computational Literacy as a Pedagogy of Resistance. Equity & Excellence in Education. 49, 480–492 (2016). https://doi.org/10.1080/10665684.2016.1227157.
22. Morales-Navarro, L., Kafai, Y.B., Gregory, K.T.: Is Computational Empowerment Promoted in the Hour of Code? Analyzing Opportunities for Conceptual, Creative, and Critical Engagement in the Design of Introductory Computing Activities. Presented at the ACM International Conference Proceeding Series (2022). https://doi.org/10.1145/3564721.3565949.
23. Wang, N., Wang, X., Su, Y.-S.: Critical analysis of the technological affordances, challenges and future directions of Generative AI in education: a systematic review. Asia Pacific Journal of Education. (2024). https://doi.org/10.1080/02188791.2024.2305156.
24. Vogel, S.: "Los Programadores Debieron Pensarse Como Dos Veces": Exploring the Intersections of Language, Power, and Technology with Bi/Multilingual Students. ACM Trans. Comput. Educ. 21, 28:1-28:25 (2021). https://doi.org/10.1145/3447379.
25. MacDowell, P., Ralph, R., Ng, D.: App Making for Pro-Social and Environmental Change at an Equity-Oriented Makeathon. In: Proceedings of the 7th Annual Conference on Creativity and Fabrication in Education. pp. 1–8. ACM, Stanford CA USA (2017). https://doi.org/10.1145/3141798.3141806.
26. Papavlasopoulou, S., Sharma, K., Giannakos, M.N.: Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. Computers in Human Behavior. 105, 105939 (2020). https://doi.org/10.1016/j.chb.2019.03.003.
27. Sáez-López, J.-M., Miyata, Y., Domínguez-Garrido, M.-C.: Creative Coding and Intercultural Projects in Higher Education: a Case Study in Three Universities / Codificación creativa y proyectos interculturales en Educación Superior: Un estudio de caso en tres universidades. RIED. 19, 145 (2016). https://doi.org/10.5944/ried.19.2.15796.

28. Dufva, T.: Creative Coding at the arts and crafts school Robotti (Käsityökoulu Robotti). Presented at the CEUR Workshop Proceedings (2018).
29. McCarthy, L., Reas, C., Fry, B.: Getting started with p5.js: making interactive graphics in JavaScript and Processing. Maker Media, Inc., San Francisco, CA (2015).
30. Shor, I.: Education is Politics: Paulo Freire's critical pedagogy. In: Paulo Freire. Routledge (1992).
31. Hallnäs, L., Redström, J.: From use to presence: on the expressions and aesthetics of everyday computational things. ACM Trans. Comput.-Hum. Interact. 9, 106–124 (2002). https://doi.org/10.1145/513665.513668.
32. Bijker, W.E., Hughes, T.P., Pinch, T.J. eds: The social construction of technological systems: new directions in the sociology and history of technology. MIT Press, Cambridge, Mass. (2012).
33. Hallnäs, L., Redström, J.: Slow Technology – Designing for Reflection. Personal Ub Comp. 5, 201–212 (2001). https://doi.org/10.1007/PL00000019.
34. Pierce, J., Paulos, E.: Counterfunctional things: exploring possibilities in designing digital limitations. In: Proceedings of the 2014 conference on Designing interactive systems. pp. 375–384. ACM, Vancouver BC Canada (2014). https://doi.org/10.1145/2598510.2598522.
35. Benford, S., Greenhalgh, C., Giannachi, G., Walker, B., Marshall, J., Rodden, T.: Uncomfortable interactions. In: Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12. p. 2005. ACM Press, Austin, Texas, USA (2012). https://doi.org/10.1145/2207676.2208347.
36. Gray, C.M., Chivukula, S.S., Lee, A.: What Kind of Work Do "Asshole Designers" Create? Describing Properties of Ethical Concern on Reddit. In: Proceedings of the 2020 ACM Designing Interactive Systems Conference. pp. 61–73. ACM, Eindhoven Netherlands (2020). https://doi.org/10.1145/3357236.3395486.
37. Gray, C.M., Kou, Y., Battles, B., Hoggatt, J., Toombs, A.L.: The Dark (Patterns) Side of UX Design. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 1–14. ACM, Montreal QC Canada (2018). https://doi.org/10.1145/3173574.3174108.
38. Mathur, A., Kshirsagar, M., Mayer, J.: What Makes a Dark Pattern... Dark?: Design Attributes, Normative Considerations, and Measurement Methods. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–18. ACM, Yokohama Japan (2021). https://doi.org/10.1145/3411764.3445610.
39. Gould, S.J.J., Chuang, L.L., Iacovides, I., Garaialde, D., Cecchinato, M.E., Cowan, B.R., Cox, A.L.: A Special Interest Group on Designed and Engineered Friction in Interaction. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–4. ACM, Yokohama Japan (2021). https://doi.org/10.1145/3411763.3450404.
40. Özkar, M.: Learning by Doing in the Age of Design Computation. In: Dong, A., Moere, A.V., and Gero, J.S. (eds.) Computer-Aided Architectural Design Futures (CAADFutures) 2007. pp. 99–112. Springer Netherlands, Dordrecht (2007). https://doi.org/10.1007/978-1-4020-6528-6_8.
41. Lodato, T.J., DiSalvo, C.: Issue-oriented hackathons as material participation. New Media & Society. 18, 539–557 (2016). https://doi.org/10.1177/1461444816629467.
42. Pater, R.: Caps Lock: how capitalism took hold of graphic design and how to escape from it. Valiz, Amsterdam (2021).
43. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., Dennison, D.: Hidden Technical Debt in Machine Learning Systems. In: Proceedings of the 28th International Conference on Neural Information Processing Systems (2015). https://doi.org/10.5555/2969442.2969519.