Designing at 1:1 Scale on Wall-Sized Displays Using Existing UI Design Tools

1st Lou Schwartz
Luxembourg Institute of Science and Technology (LIST)
Esch-sur-Alzette, Luxembourg
lou.schwartz@list.lu

2nd Mohammad Ghoniem
Luxembourg Institute of Science and Technology (LIST)
Esch-sur-Alzette, Luxembourg
mohammad.ghoniem@list.lu

3rd Valérie Maquil
Luxembourg Institute of Science and Technology (LIST)
Esch-sur-Alzette, Luxembourg
valerie.maquil@list.lu

4th Adrien Coppens
Luxembourg Institute of Science and Technology (LIST)
Esch-sur-Alzette, Luxembourg
adrien.coppens@list.lu

5th Johannes Hermen
Luxembourg Institute of Science and Technology (LIST)
Esch-sur-Alzette, Luxembourg
johannes.hermen@list.lu

***Abstract*—Wall-Sized Displays have spatial characteristics that are difficult to address during user interface design. The design at scale 1:1 could be part of the solution. In this paper, we present the results of two user studies and one technology review, exploring the usability of popular, desktop-optimized prototyping tools, for designing at scale on Wall-Sized Displays. We considered two wall-sized display setups, and three different interaction methods: touch, a keyboard equipped with a touchpad, and a tablet. We observed that designing at scale 1:1 was appreciated. Tablet-based interaction proved to be the most comfortable interaction method, and a mix of interaction modalities is promising. In addition, care must be given to the surrounding environment, such as furniture. We propose twelve design guidelines for a design tool dedicated to this specific context. Overall, existing user interface design tools do not yet fully support design on and for wall-sized displays and require further considerations in terms of placement of user interface elements and the provision of additional features.**

Keywords-wall-sized display; UI design; design at 1:1 scale; user study; large scale display.

I. Introduction

This paper extends a previous study on the usability of Figma, a popular user interface (UI) design tool, for Wall-Sized Displays (WSDs) at the CENTRIC 2024 conference [1]. WSDs are increasingly used in public spaces to provide general or contextual information, provide entertainment, or for artistic purposes [2] [3]. WSDs are also applied in many research areas. Traffic management [4] and automotive design [5] benefit from their large display area. They also support data browsing and manipulation [6] [7], and are crucial for visualizing and interacting with vast amounts of data in natural sciences like physics, astronomy, chemistry, and biology [8] [9]. In the medical field, WSDs aid interdepartmental communication [10], optimize surgery room scheduling, and improve team transitions [11] [12]. They are also used for scheduling activities, such as conference organization [13], and support collaborative design tasks [14], including architectural design reviews [15].

WSDs are also referred to as vertical Large Interactive Displays (LIDs) or Large High-Resolution Displays (LHRDs). However, the notion of ‘large’ is typically not precisely defined and can be subjective [16]. Belkacem et al. defined an LHRD as a display that “*creates a coherent physical view space that is at least of the size of the human body and exhibits a significantly higher resolution than a conventional display*” [17]. According to Chen et al., WSDs improve user performance and satisfaction for tasks, such as model design, analysis, and visual data mining [18]. But, these new ways of viewing, collaborating and interacting differ from desktop and smartphone applications [16], because of their size, their resolution, the collaboration they foster, and the so-called *natural* interactions they enable, mainly through touch and gestures [19]. WSDs vary in terms of visualization technology, display setup (size, orientation), interaction modality, application objectives (productivity, entertainment, social interaction, games, and advertising) and location (city, office, education, conference, third place and cultural site) [19].

Based on previous work, we enumerate nine challenges raised by WSDs, each needing further research [16] [17] [20]: 1) *More interactions*: “natural” interactions with a mix of interaction methods [16] [20]. 2) *More users*: and improvements regarding how they collaborate [16] [17] [20]. 3) *More space (around)*: users’ movements and interactions in the space and the management of the inherent fatigue [17] [20]. 4) *Different usage durations*: for example, collaborative decision-making tasks require long sessions spread over time [20], but for public displays, the duration of use is often very short [16]. 5) *More complex content*: complex data representations, large quantities of complex data needed by experts [16] [17] [20]. 6) *More devices*: WSDs are often composed of several displays, but are also often supplemented by other devices (e.g., supplementary displays or sensors) [16] [17] [20]. 7) *Variable screen space and pixel count*: displaying more data simultaneously requires content to be designed differently to distinguish between and manage the visible, accessible and useful parts of the display [16] [17] [20]. 8) *Support for designers*: tools or methods for designing and testing are needed, as well as guidelines and best practices [16] [20], and 9) *Other concerns* arise like accessibility, compatibility, and portability from one WSD to another, and workflow management [20].

Supporting designers with the right tools and methods to design applications for WSDs is hence an open challenge. Similar questions have also been raised regarding the design of data visualization interfaces for WSDs, specifically [17] [21]. Overall, the challenge is three-fold: the difficulty of scaling visual elements, e.g., text, the limited availability of design software, and the lack of widely adopted design guidelines.

In this paper, we address the *designer support* challenge, i.e., the need for design and testing tools and methods [16] [20]. We look into the design of a UI prototype in WSD environments at 1:1 scale and seek to assess the suitability of existing design tools for this purpose. By 'UI prototyping', we mean the prototyping of the interface, functionalities, screen layouts and behaviors at the mid-fidelity level. We focus on using popular UI design tools, such as Figma [22] and Miro [23], to prototype UIs in WSD environments at 1:1 scale. We have no conflict of interest with any of them.

In the rest of this paper, Section II presents related work on methods and tools for prototyping for WSDs. Our research approach is detailed in Section III. Section IV describes a first user study using Figma to design for WSD at 1:1 scale. This allowed us to identify the required features to support design at 1:1 scale for WSDs. Section V compares existing design tools based on these features. Section VI presents a second user study, diving deeper into the use of another design tool (Miro) with different WSD setups and interaction means. Our observations are discussed in Section VII and used to draw implications for design, and for tooling improvements. The section also discusses the limitations of both user studies. Finally, Section VIII holds a general conclusion.

II. Related work

Many tools and methods have been proposed to support design for WSDs. Below we discuss paper prototyping, prototype development and mixed mockup techniques. We also cover briefly interaction techniques used in WSD environments.

A. Paper prototyping

Paper prototyping is a popular, validated and simple method of mocking up systems before programming [24]. It allows designers to explore, communicate and evaluate early interface designs with end-users or within the design team. A designer typically plays the role of the computer to simulate the behavior of the system by changing the pieces of paper shown to the participants. Numerous studies have used paper prototyping to design applications on a WSD (e.g., [13] [25] [26] [27]).

Bailey et al. used this method to prototype a multi-device environments (MDE) involving personal devices (e.g., computers or tablet), and a wide screen to share data and views [25]. They observed that A4 paper lacks accuracy for large displays due to scaling issues. Indeed, the size hampers reading text at a distance. WSDs require sheets of paper larger than A4, and text size should be adapted to the screen size. However, paper prototyping does not allow checking whether the user can quickly and easily detect where information is displayed and whether changes in the displayed content would be noticed, as the user can follow the facilitator's gaze direction and movements and see where she places the pieces of paper.

Paper prototyping can also be used to define the screen arrangements to compose the WSD [28].

However, paper prototyping is mainly used for UI prototyping on WSDs.

B. Functional prototyping

Mid-fidelity prototype development is also a common practice [29]. Indeed, in the following papers, for example, there are no indication on how the applications were designed, but a developed prototype is used for the studies within them [4] [6] [7] [14] [30] [31].

Some systems have been developed to support prototyping, among others, for WSDs. For instance, *DEBORAh*, is a front-end web-based software layer that supports the orchestration of interactive spaces combining multiple devices and screens, including WSDs [32]. Additionally, *jBricks*, a Java-based toolkit, enables the exploratory prototyping of interaction techniques and rapid development of post-WIMP applications for WSDs, particularly for tiled-displays [33].

C. Mixed mockup techniques

Finally, mixed mockup techniques, e.g., *Mini-studio* [34], consist in using a physical paper model of the system augmented with projected content. They are mainly used to prototype ubiquitous computing systems, but can also be used to mock up WSDs. *SketchStudio* is another example, which combines 2D devices with 3D characters, resulting in a 2.5D animated scenario design tool for rapid prototyping of systems involving multiple users and multiple components or devices [35]. Such methods and tools offer the advantage of allowing the interactions around the WSDs to be tested. However, they are not accurate enough for a prototype of the screen layout and content, especially in contexts where large amounts of data and high resolution are required [17].

Overall, the prototyping method is frequently used for designing WSD applications, but the way the design was conceived is usually not described. Among the few occurrences where the design process is documented, paper prototyping is the most commonly used method. We did not find any studies covering the design of a UI prototype on a WSD at 1:1 scale.

D. Interaction techniques for WSD environments

Various interaction modalities have been used with WSDs, ranging from classic mouse and keyboard to more advanced types of interaction leveraging touch, gaze, mid-air gestures, proxemics, handheld and wearable devices, and tangibles [17].

Mouse and keyboard interaction is readily available in virtually all interface design tools used in common desktop environments. Users may work from a distance while being seated. The physical keyboard is very convenient for text entry, but text may lack legibility at a distance. Virtual keyboards have also been studied in the context of mobile devices [36], horizontal tabletops [37] and virtual environments [38]. They can be spawned on vertical display surfaces too, but hardly any work investigates their usability in this context. While mouse interaction is faster and more accurate than touch [39], the mouse cursor is easily lost and clutching may be tedious when navigating across very large scenes.

The cursor can also be attached to other pointing devices, be it a touchpad or a handheld device such as remote controllers, or flysticks, or head-mounted displays through eye tracking. Handheld display devices, such as smartphones or tablets, can also be turned into pointing devices by attaching sensors to them, or by using them as touchpads to interact with the WSD.

Direct touch can also be used on touch-enabled WSDs. On the one hand, touch is an intuitive interaction modality that exhibits high user performance and acceptance scores [40]. On the other hand, users often have to move across the WSD, or step up close or away from the WSD, to look at details, or get the big picture, respectively [41]. They may also have a hard time reaching the upper or the lower part of the WSD [42].

Overall, many advanced interaction techniques may generate fatigue or muscle strain, e.g., gorilla arm [41], when used for a long time. They also differ significantly from widespread mouse and keyboard interaction, in terms of accuracy, speed, and their hedonic value [40], and may be less suited for certain tasks such as text entry. A more detailed discussion of these interaction techniques for WSDs can be found in the literature [17].

In this work, we evaluate the suitability of popular interface design tools, originally meant for the desktop, to design UIs for WSD environments at the 1:1 scale. Most design tools were not developed for WSD, nor optimized for advanced interaction modalities available in WSD environments.

III. Research approach

As noted by Lischke et al., when it comes to WSDs, it "*is often not possible to prototype in the original size*" [16]. Unlike using a desktop computer to design UIs for WSDs, prototyping in real size directly onto the targeted display could reduce complexity, give a sense of scale, and ensure that the target resolution is correctly achieved and exploited. It could also allow designers to check that the UI is visible at all distances and from all angles [17] [20]. Hence, our interest in prototyping on a WSD is based on supporting the design in real size.

A. Research questions

As part of understanding how to support the design in 1:1 scale for WSDs, we focus on three main research questions:

RQ1: Can a desktop optimized tool be used in a WSD environment to design at 1:1 scale?

RQ2: What would be the best interaction modality during the design at 1:1 scale on a WSD ? And what are the main issues raised by each interaction modality?

RQ3: What are the main features needed to support the design at 1:1 scale on a WSD ?

This research aims to extract initial guidelines for designing at 1:1 scale *on* and *for* WSDs. To answer these questions, we have set up two exploratory user studies and performed a comparative analysis of existing design tools. The first user study involved one participant who consecutively interacted through three distinct interaction methods (a keyboard and its embedded touchpad, direct touch on the WSD, or a connected tablet) and with two WSDs to understand whether and how a desktop optimized tool, Figma, could be used in a WSD environment to design at 1:1 scale. The outcomes of the first user study pointed us to important features, as a basis for doing a technology review and comparative analysis, and selecting a better tool, i.e., Miro, for the next user study. In the second user study, we observed the use of Miro by two participants designing at 1:1 scale on two WSDs with the same three interaction modalities.

B. Task

In both user studies, the task consisted in using a desktop optimized software (Figma or Miro) to reproduce a previously developed UI [43] and adapted for both WSDs as shown in Figure 1, more details can be found online [44] [45]. Before the test session, the participants discovered the design tool on a desktop computer for two hours, and the mock-up they had to reproduce.

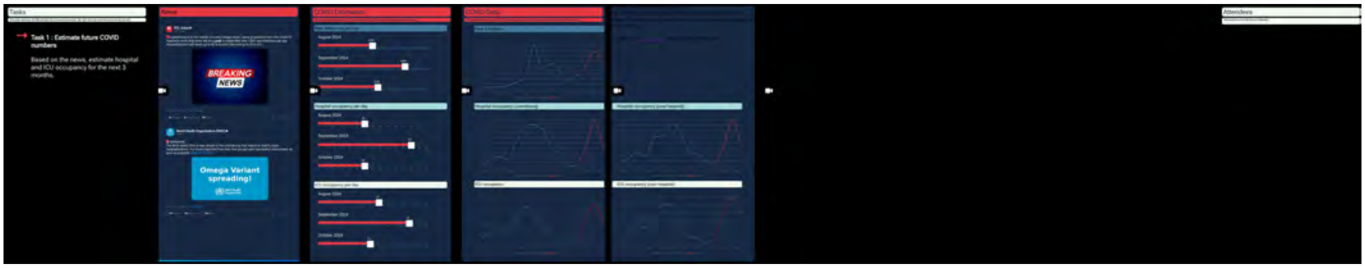
This UI was chosen because it comprises a variety of UI elements (text, sliders, graph, a social media feed). Reproducing an existing UI ensures that it is feasible, well adapted to the WSD environment, and allows the observation to be focused on the design software manipulation rather than the process of creating a new design.

C. System

The system consisted of a touch-enabled WSD displaying the design tool (either Figma or Miro) in a web browser window (Google Chrome [46]) in full-screen mode.

Two WSDs were used. First, WSD-IA (curved, diameter: 3.64m, height: 2m, composed of twelve 4K screens in portrait mode, among which eight are touch-enabled using infrared frames. The setup also included a height-adjustable table and a keyboard/touchpad, as shown in Figure 2.

Then, WSD-VW (flat, width: 7m, height: 2m, total resolution $13,152 \times 3,872$ pixels, composed of 24 HD screens with infrared frames enabling touch (see Figure 3). This WSD is located in a room containing three high tables (fixed-height) placed at each end of the WSD, with two mobile extended-height chairs each. The room also contains a large standard height table facing the middle of the WSD (about 3 meters away), and the display itself includes a virtual touch keyboard that appears at the bottom center of the WSD.



Tasks

Task 1 : Estimate future COVID numbers

Based on the news, estimate hospital and ICU occupancy for the next 3 months.

Figure 1. Screenshot of the prototype to replicate, for more information see [43] and [44] [45].

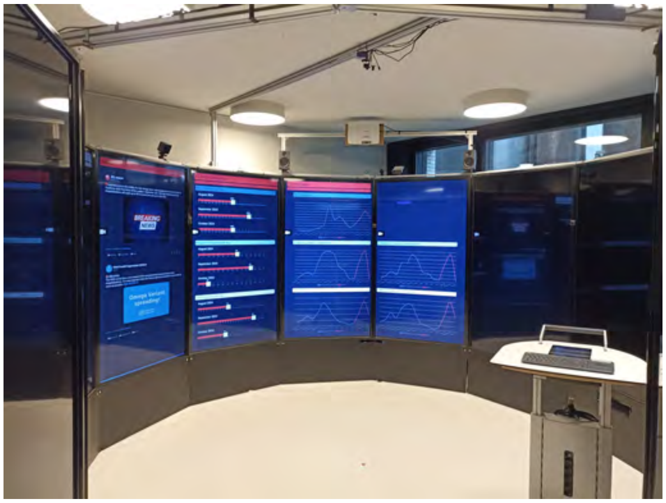


Figure 2. WSD-IA displaying the interface used for the task.

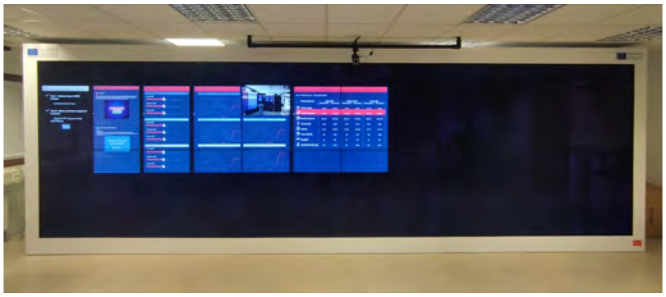


Figure 3. WSD-VW displaying a version of the interface used for the task.

D. Observation and analysis methods

Video cameras and microphones were used to record the sessions. For WSD-IA, three video cameras were used, one top camera placed in the middle of the room and attached to the roof, one in the center and attached to the top of the WSD (in front of the entrance), and one on the back, attached at the top of the entrance screens (the two screens, which can be rotated); see Figure 4 and Figure 5 for clarifications. For WSD-VW, two cameras were placed in the back, in front of both ends of the display for the first user study. They were completed by a third one at the middle back for the second user study. At the end of each session, the participant was invited to discuss and debrief with the facilitator. We analyzed thematically the comments made by the participants, both based on the debriefings and during the sessions themselves, to identify encountered issues. We additionally observed the moves and strategies the participants relied on.

IV. Prototyping at 1:1 scale on WSD with Figma

During the first exploratory user study, we used Figma, a mid-fidelity web-based prototyping tool for designing, collaborating, prototyping and transmitting content [22]. We chose it for its popularity [47] [48] and its availability as a ready-to-use web-based solution. Figma was tested on a WSD by one designer under several experimental conditions: two WSD settings (WSD-IA, and WSD-VW) with different arrangements and surroundings, and three different interaction methods to reproduce the interface: a wireless keyboard with a touchpad, direct touch on the WSD, and a synchronized tablet.

The main research question addressed by this first user study is: **can Figma**, as a desktop optimized tool, **be used in a WSD environment to design at 1:1 scale?**

In other words, we want to identify the challenges and opportunities raised by using an existing design tool to prototype the UI of an application built for a WSD, directly on the WSD.

A. Protocol

The **participant** is an expert in UI design and has participated in the design of several UIs for WSDs, but had never used Figma before. She was free to stop the session whenever she wanted (e.g., when it became too difficult) or after having finished reproducing the design. Since this first user study aimed to verify the feasibility of using Figma under these conditions before carrying out more in-depth studies, a single user was deemed sufficient. Conversation guide and detailed protocol are available in supplementary material online [44].

Figure 4 illustrates the interaction methods tested to interact with Figma to complete the **task** as described in Section III: respectively, a wireless keyboard with a touchpad, direct touch on the WSD, and a tablet synchronized with the WSD.

B. Results

In general, the participant appreciated the ability to design at a 1:1 scale, regardless of the interaction method and the WSD used, with the main advantage of being able to see in real time the final rendering on the target screen. Several issues are related to the lack of familiarity with Figma, as the use of widgets, components, and plugins seemed complicated, and were not used effectively by the participant. Also, some

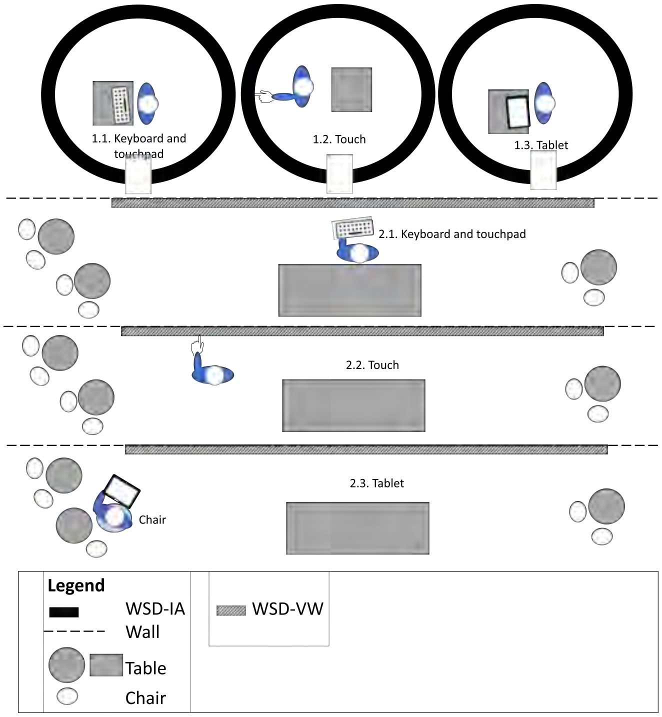


Figure 4. Experimental settings: at the top, the curved WSD-IA, at the bottom, the flat WSD-VW. The spatial relationships depicted here are illustrative and may not correspond to real-world dimensions.

problems are due to or emphasized by the circular nature of WSD-IA.

1) General observation about Figma: The configuration of the **Figma environment** was not always suited for WSDs. For instance, the properties of a selected object are placed on the right-hand side of the display (see Figure 6.a), the main menu is placed at the very top (see Figure 6.b) and dialogue boxes open in the middle of the display. The user must also move the cursor across the entire display or physically walk to the desired location to modify, e.g., element properties (see Figure 5.d), which is tiring on the long run. Below, we discuss in more detail the issues related to each interaction method.

2) Interacting with a wireless keyboard with a touchpad: The session lasted one hour for WSD-IA and ten minutes for WSD-VW. On both WSDs, the participant would sometimes have a hard time finding the cursor on the large display.

Concerning the *WSD-IA*, the menu and items list were displayed on the leftmost screen (one of the two screens also used as a door), and the selected item's properties on the rightmost screen (the other screen forming the side by side door). To avoid turning her head from the extreme left to the extreme right too often, the participant closed the doors to look at them both at the same time. She also placed herself to create an angle that allowed her to see the menu, the properties of the selected object, and the work area at a glance (see Figure 5.c). As the session was short, and all UI elements were tightly grouped on the left, the position was acceptable. But the user could not maintain this position while working in the middle. In this configuration (menu on the left and properties on the right), the participant turned her head and body a lot, which could possibly be painful and exhausting. At first, the participant carried the interaction device, see Figure 5.a. After 15 minutes, she felt tired and placed it on a table, see Figure 5.b. Another problem was the impossibility to move the cursor directly from the WSD's leftmost side to its rightmost side. The participant had to move the cursor all the way around the WSD, which is tiring, despite the physical proximity of both sides of the WSD due to its circular arrangement (see Figure 5.d). To avoid turning her head too much, the participant did not follow the cursor with her eyes when it was behind her back.

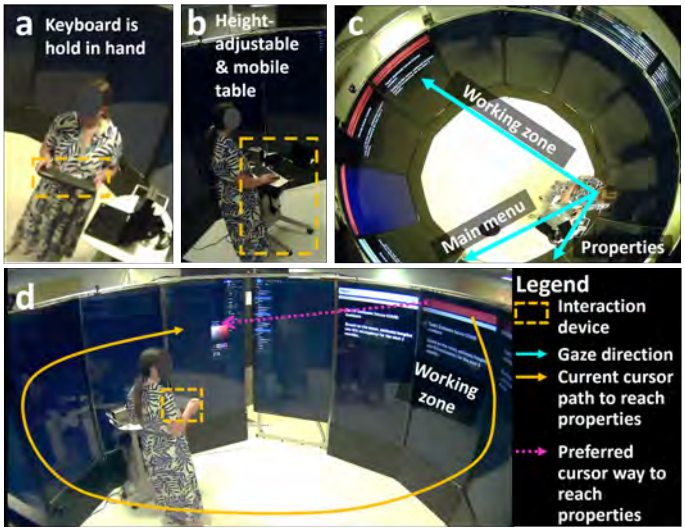

a Keyboard is hold in hand
b Height-adjustable & mobile table
c
Working zone
Main menu
Properties
d
Working zone
Legend
Interaction device
Gaze direction
Current cursor path to reach properties
Preferred cursor way to reach properties

Figure 5. Observations made when interacting with Figma and keyboard+touchpad. a) At first, the participant held the keyboard. b) Use of a table to put down the keyboard. c) Lots of head rotations to see all the important areas. d) WSD-IA does not allow you to cross directly from the left screen to the right screen.

In the *WSD-VW* condition, the font size of the Figma interface was problematic. The flat WSD-VW was too wide to read text labels when standing at the opposite side of the display. Hence, to modify a property's value, the user had to walk frequently across the WSD-VW to the properties area, where she rested on a table next to it. Then she leaned on the middle table for comfort and stayed at a certain distance from WSD-VW to get an overview. The fatigue due to walking around, eye strain due to the text size, and carrying the keyboard led the user to stop the test after ten minutes. Because the WSD-IA is twice as large as the WSD-VW in terms of horizontal resolution, the virtual navigation felt more painful in WSD-IA. This is also the case because the cursor and its progress were always visible in WSD-VW, and the session was shorter.

3) Interacting using direct touch on the WSD: The session with WSD-IA lasted twenty minutes, and the session with WSD-VW was interrupted after ten minutes.

In the *WSD-IA* condition, to manage physical fatigue (neck strain and gorilla arm syndrome [49]), the user tried to work

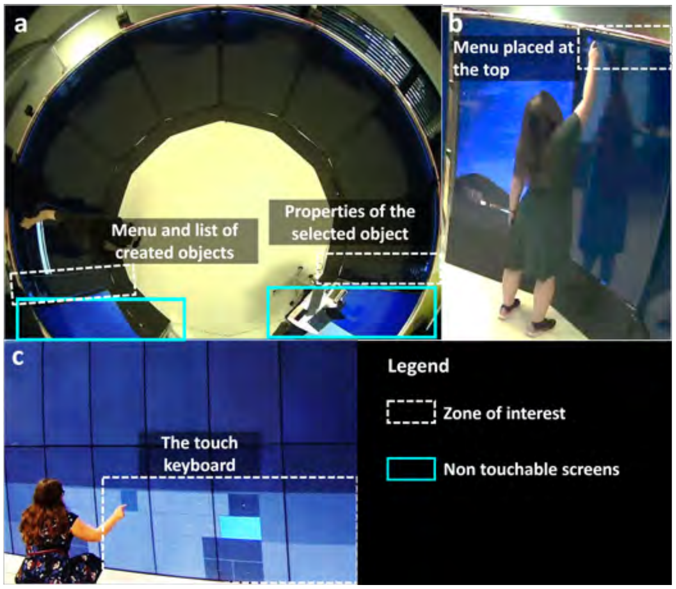


Figure 6. Observations done when interacting with Figma and touch. a) The touch space is occupied by the Figma interface on the left (list of created objects) and the right (properties). b) The menu is too high. c) WSD-VW's touch keyboard is not comfortable; the user had to crouch.

at a lower scale by zooming in on the work area without minimizing the Figma window. Although the menu remained too high and objects' properties too far away, objects could be moved with smaller movements and were better placed relative to the user's field of view, generating less neck pain. Hence, the advantage of working at 1:1 scale was temporarily lost. Only the middle eight screens of WSD-IA support touch, so we resized the browser window used for Figma to fit within these screens. As Figma's interface elements (list of objects and properties) take up space themselves at both ends of the window, the design space was further reduced, by about one extra screen for interface elements from both ends combined. Hence, the total design space available to the participant was reduced compared to the actual interface they had to reproduce. Consequently, the alignment with the display tiles could not be maintained, as shown in Figure 6.a. When using touch, the participant had trouble moving an object across tiles. The wireless keyboard was used to input text or values and was either held by the participant or placed on the table.

On *WSD-VW*, the properties panel was too far away from the work area, but unlike WSD-IA, when a property was changed on WSD-VW, the result was not visible from the user's position. So, she stepped back to check, e.g., whether the font size is large enough. The top menu was out of reach, and WSD-VW's virtual keyboard was not suitable for entering more than one word due to its design (position at the bottom and large size, see Figure 6.c). After ten minutes of use, the participant complained from the gorilla arm syndrome.

4) Interacting on a synchronized tablet: The same Figma project was loaded onto the tablet and onto the WSD. The UI elements were created, moved and adjusted on the tablet.

We observed that the participant mainly looked at the tablet to add UI elements, move them around and set parameters, see Figure 7.a. Then, the participant looked at the WSD to check, e.g., the position and size of the UI elements, the readability of text, and colors, see Figure 7.b. A main issue was the impossibility to select several UI elements at the same time on the tablet, as they are superimposed. The session with WSD-IA lasted ninety minutes, whereas the session with WSD-VW was interrupted after twenty minutes.

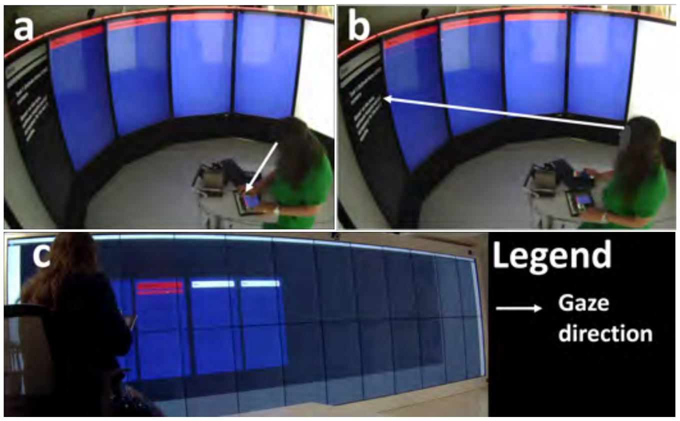


Figure 7. Observations done when interacting with Figma and a tablet. a) The participant modified the prototype on the tablet. b) Then, the participant checked the result on the WSD. c) The participant sat on a chair.

On *WSD-IA*, the user had difficulties to position the UI prototype on the WSD correctly. Although the additions and changes to UI elements were synchronized between the tablet and the WSD, the viewports were independent, so positioning the UI had to be done from the WSD directly. This led to the use of an extra wireless keyboard equipped with a touchpad. This happened when the participant closed the project and reopened it. The designed UI was centered horizontally and vertically on the WSD. So, the participant wanted to properly reposition the designed UI to continue the design. The participant also used the touchpad to select a group of UI elements to save them as a new reusable UI element. She placed the tablet on the height-adjustable and mobile table. She felt that WSD-IA and tablet configuration was the most comfortable.

On *WSD-VW*, the participant sat down and placed the tablet on the table, see Figure 7.c. But, as the table was not well positioned and too heavy to be moved, she preferred to hold the tablet in her hand, which was tiresome.

C. Preliminary observations

The duration of the test sessions varied widely, from ten to 90 minutes. The most comfortable condition seems to be WSD-IA with a tablet and a height-adjustable and mobile table. But the problem of multiple selection and correct positioning of the prototype on the WSD needs to be solved. Overall, the main issues were: (I1) physical fatigue, (I2) reachability of Figma elements, (I3) readability of the Figma interface, (I4) the vast interaction surface, (I5) when a project is reopened, objects are moved to the middle, and

(I6) the partial occlusion of the WSD by Figma's UI elements, which is not a perfect 1:1 scale.

To tackle these issues, we outline several design solutions. (I1) could be reduced by managing the physical environment and providing a height-adjustable and movable table to place the interacting device, chairs, and by supporting interaction at a distance. (I2) could be improved by offering floating context menus and value input fields, by opening dialogue boxes close to the work area or by using a smaller interaction device as a tablet or a laptop. For (I3) and (I4), the size of the design tool interface elements should be adapted. For (I4), a bigger cursor should be used as well as pointing facilitation techniques [50] to reach the opposite end of the WSD. (I5) could be solved by fixing the created objects in their positions and reloading them in exactly the same position. To achieve 1:1 scale (I6), the design tool interface should be hidable or movable.

Overall, issues (I2), (I3), (I5) and (I6) show that Figma may not be adapted to prototype at 1:1 scale on WSDs, and that another tool incorporating the outlined design solutions, might be better suited.

V. Comparative analysis of existing design tools

This first user study enabled us to identify the problems with using Figma as a 1:1 scale design tool for WSDs. Based on these results, we then set up a list of required features for such a tool. Below, we first list these features and then analyze 19 different design tools, meant for UI design for computers and smartphones, taking into account the listed features.

Based on our observations, we identified the following main features of tools suited for designing at 1:1 scale on WSD:

- The main menu must be reachable from the work area (not at the top or bottom) [related to I1, I2, I4].
- The properties of the selected object must be reachable from the work area [related to I1, I2, I4].
- Pop-ups must open near the work area [related to I1, I2, I4].
- The font size of the design tool UI elements must be large enough (or adjustable) [related to I3].
- It must be possible to conceal or move the design tool interface elements, i.e., menu, list of created objects, and properties of the selected object [related to I6].
- It must be possible to interact from a synchronized tablet [needed for our user study].
- When reopening a previous project, it must preserve the positions of the created interface elements, as they were previously. [related to I5]
- The workspace size must be large or adaptable to the typically high resolution of WSDs.

Based on this features list, we analyzed 19 design tools. We performed a search using the engine Google with the keywords "design tool", "mockup tool" and "UI design tool" between July and December 2024. We collected the design tools from direct results or from online articles about such tools [48] [51]. We then eliminated the tools that did not support the creation of mock-ups, and the ones that could not be tested and used for free. In the end, we selected 19 tools, which were analyzed against the listed features. Many tools set fixed positions for their menus, and for object properties panels. None of the tools in the selection enabled users to conceal or move the design tool's interface elements.

As our WSDs run on a Linux operating system, we also had to narrow down our list to multiplatform tools, which indirectly led us to mainly consider web-based solutions.

Table II lists the design tools and our analysis of their characteristics against the desirable features. Concerning the maximum surface size of the workspace, this information was rather complex and at times impossible to find for some design tools, so some figures listed in the table might not be accurate. As most of the examined tools provide a web-based version, thus giving access to the zoom functionality of the web browser, we did not check the font size of the design tool's interface.

Based on the features they offer, we pre-selected three multiplatform design tools (Bubble, inVision, and Miro) to test them more in-depth on one WSD (WSD-IA), with the three interaction methods used in the user study, to ensure their usability for the second user study. With inVision, we ran into issues when using touch on our WSD, so we had to reject it. Additionally, we noticed that the solution was discontinued shortly after our tests. As for Bubble, we experienced some synchronization latency when using the tablet, which is why we discarded it as well. We therefore chose Miro as design tool, because it satisfied most of the requirements, including running in a web browser, showing object properties in a context menu, reopening a project at the same position, retractable interface by using the preview mode, and offering the option to open the same project on a tablet as on the WSD.

VI. Prototyping at 1:1 scale on WSD with Miro

Based on the findings from the first user study and the feature matrix, we ran a second user study with some modifications in the protocol compared to the first study, as explained below.

The research questions are adapted from the first study. RQ1: **Can an existing design tool**, that was conceived for desktops, **be used in a WSD environment to design at 1:1 scale?** And RQ2: **What are the main guidelines to support design at 1:1 scale in a WSD environment?**

A. Protocol

Miro was tested as a mockup tool for WSDs by two designers under several conditions: two WSD settings (WSD-IA, and WSD-VW), and three different interaction methods: (1) a wireless keyboard with a touchpad, (2) touch on the WSD completed by a wireless keyboard for entering textual and numeric values, and (3) a synchronized tablet.

Sessions were limited to one hour. But, participants were free to stop the session earlier, if they felt too tired, or uncomfortable, or if they completed the task.

The two **participants** were respectively an expert in UI design (p2) and in visualization design (p1). They both participated in the design of several UIs for WSDs, although (p2) is

TABLE I. Main issues and possible solutions.

	Issues	Keyboard/touchpad		Touch		Tablet		Possible solutions
		WSD-IA	WSD-VW	WSD-IA	WSD-VW	WSD-IA	WSD-VW	
I1	Physical fatigue	Yes frequent head and torso rotations	Yes hand-held, much walking	Yes Menu too high, much walking	Yes Menu too high, much walking, touch keyboard too low	No placed on a table	Yes hand-held	A height-adjustable movable table on which to place the interaction device, a chair for sessions longer than 90 minutes. Interacting at a distance.
I2	Figma's elements reachability	Yes Properties too far from the working zone	Yes Properties too far from the working zone	Yes Properties too far from the working zone	Yes Properties too far from the working zone	No	No	Floating context menus and value input fields near the working zone, allow the participant to move across the display without scrolling, open dialogue boxes near the working zone, and use a smaller interaction device.
I3	Readability	No	Yes UI element not visible when modifying its parameters. Figma interface not readable.	No	Yes UI element not visible when modifying its parameters	No	No	Adapt the size of the Figma elements, use a curved WSD instead.
I4	Huge interaction surface	Yes It takes a long time to reach the extremities.	Yes It takes a long time to reach the extremities.	No	No	No	No	Adapt the size of the Figma elements, use a curved WSD instead. Bigger cursor, pointer acceleration to reach the opposite side of the display.
I5	Not fixed position of UI elements	Yes When reopening the project.	Yes	Yes	Yes	Yes	Yes	Fix the UI elements in their position and restore their position after reloading the project. The Figma design space should match the targeted WSD.
I6	1:1 scale not perfect	No	No	Yes Part of the necessary design surface masked by Figma elements.	No	No	No	Possibility to conceal or move the Figma interface elements such as the menus.

TABLE II. Features matrix of the UI design tools. Cells in gray highlight instances where features or requirements we identified are supported by the corresponding design tool. NA: not applicable. * No source for inVision as the design tool was stopped after our review. Adobe XD is now in maintenance mode. ** E.g., delete.

Name	Max surface size	Supported platforms	Main menu position	Object properties position	Dialog boxes position	Reopens at the same position	Retractable interface	Synchronization with a tablet	Selected
Adobe XD* [52]	50,000x50,000 px	macOS, Windows, Android, iOS, browsers (Chrome, Firefox, Edge, Safari)	fixed on the top and left	fixed on the right	middle	unknown	unknown	unknown	no
Axure RP [53]	20,000 px	Windows, macOS, Android, iOS	fixed on the top	fixed on the right	middle	unknown	unknown	unknown	no
Balsamiq [54]	20,000x20,000 px	Windows, macOS, browsers	fixed on the top	fixed on the right	middle	yes	partially + preview mode	possible with the cloud version	no
Bubble [55]	max width 5,000 px	browsers	fixed on the left	floating and movable	unknown	no	partially + preview mode	Possible with the opening of the same project, but lag	yes
Canva [56]	8000 x 3125 px	browsers, Windows, macOS, iOS, Android	fixed on the left	fixed on the top some actions attached to the selected object**)	middle	no	partially + preview mode	Possible with the cloud version	no
Excalidraw [57]	unknown	browsers	fixed on the top	fixed on the left	unknown	yes	partially	using collaboration feature	no
Figma [22]	no limit	browsers	fixed on the top	fixed on the right	middle	no	partially + preview mode	Possible with the opening of the same project or using collaboration feature	no
Framer [58]	max width 1280px	browsers, macOS, Windows	fixed on the top and left	fixed on the right	middle	no	preview mode	Possible with the cloud version,	no
InVision*	unknown	browsers	fixed on the bottom middle	attached to the selected object	top left	no	unknown	Possible with the opening of the same project,	yes
JustInMind [59]	unknown	Windows, macOS	fixed on the top	fixed on the right	middle	unknown	unknown	using collaboration feature	no
Miro [23]	unknown	browsers, iOS, Android, macOS, Windows	fixed on the left	attached to the selected object	middle	yes	preview mode	Possible with the opening of the same project, or using collaboration feature	yes
MockFlow [60]	unknown	browser, macOS, Windows	fixed on the left	fixed on the left	unknown	no	no	unknown	no
Mockplus [61]	no limit	browsers, Android, iOS, Windows, macOS	fixed on the top and left	fixed on the right	middle	yes	preview mode	using collaboration feature (only visualization, no modification)	no
Penpot [62]	no limit	browsers	fixed on the top	fixed on the right	middle	yes	yes + preview mode	using collaboration feature	no
Proto.io [63]	unknown	browsers (Chrome, Safari, Firefox)	fixed on the right	fixed on the right some actions attached to the selected object	middle	no	preview mode	unknown	no
ProtoPie [64]	unknown	Windows	fixed on the top	fixed on the right	middle	unknown	unknown	unknown	no
Sketch [65]	unknown	macOS	fixed on the top	fixed on the right	unknown	unknown	unknown	unknown	no
UXPin [66]	25000x25000 px	browsers, macOS, Windows	fixed on the left	fixed on the right	middle	yes	preview mode	Possible with the opening of the same project, but lag	no
Webflow [67]	max width 10,000 px	browsers	fixed on the left	fixed on the right	middle	no	preview mode	unknown	no

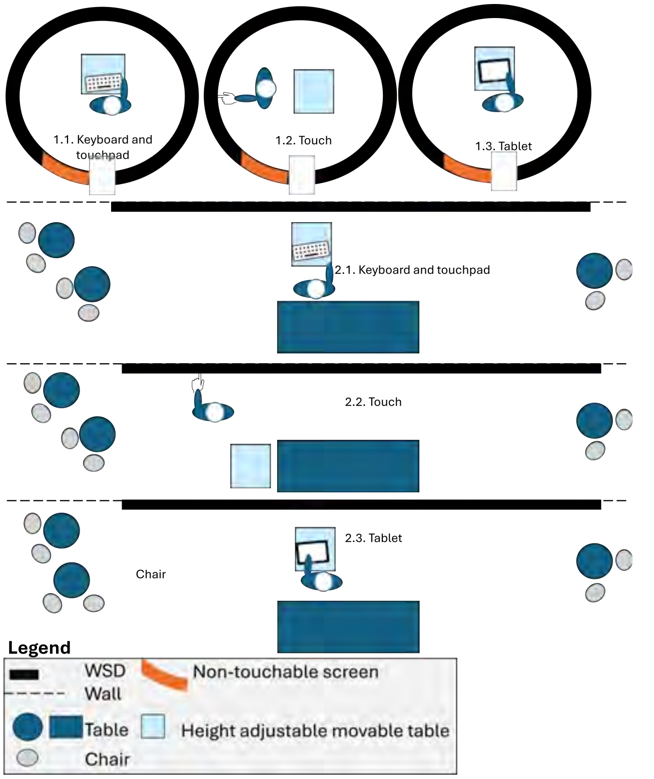


Figure 8. Experimental settings for the second study: on the top the curved WSD-IA, on the bottom the flat WSD-VW. The spatial relationships depicted in this figure are illustrative and may not correspond to real-world dimensions.

more experienced in UI design per se. They both had already used Miro beforehand as a collaborative mind mapping tool, but not for UI design, with p2 who was competent using Miro and p1 self-declaring as more novice.

Like in the first study, the **system** used for this second user study consists of a touch-enabled WSD displaying the design tool, Miro in this case, through the Chrome web browser. The zoom level in the browser was set to 200%, for the main menu items (located on the left and vertically centered) to be wide enough to be usable. We used the same two WSDs as previously: WSD-IA (curved) and WSD-VW (flat).

Learning from the first user study, some adaptations were made on both WSDs. As for WSD-IA, additional infrared touch frames were put in place, increasing the number of touch enabled screens to eleven. As text input with the virtual keyboard on WSD-VW was quite cumbersome, and to ensure a fair comparison between the conditions between both WSDs, participants were also provided with a physical keyboard for the touch condition, but its use was limited to text input. We have also provided a height-adjustable movable table with WSD-VW, to allow users to put down the interaction device anywhere in the room (see Figure 8).

The **task** consisted in using Miro to replicate the same UI as in the first study (see Figure 1). The participants first discovered Miro on a desktop computer for two hours, with the goal of reproducing another small UI similar to the main one in terms of components to be drawn. Detailed task, protocol, and discussion guide are available in supplementary material [45].

TABLE III. Order of test conditions for the second user study using Miro.

Condition ID	WSD	Interaction method
Participant 1 (p1)		
IA-Keyboard-p1	WSD-IA	Keyboard
IA-Tablet-p1	WSD-IA	Tablet
IA-Touch-p1	WSD-IA	Touch
VW-Touch-p1	WSD-VW	Touch
VW-Keyboard-p1	WSD-VW	Keyboard
VW-Tablet-p1	WSD-VW	Tablet
Participant 2 (p2)		
IA-Tablet-p2	WSD-IA	Tablet
IA-Touch-p2	WSD-IA	Touch
IA-Keyboard-p2	WSD-IA	Keyboard
VW-Keyboard-p2	WSD-VW	Keyboard
VW-Touch-p2	WSD-VW	Touch
VW-Tablet-p2	WSD-VW	Tablet

We used similar interaction methods as before, i.e., a wireless keyboard with a touchpad, touch on the WSD with a keyboard, and a tablet synchronized with the WSD (see Figure 8). We changed the order in which participants used the interaction methods, as shown in Table III. For logistic reasons, the tests were run on WSD-IA first, and then on WSD-VW.

At the end of each session, the facilitator debriefed with the participants to discuss their feedback and suggestions for improvement. The debriefing collected spontaneous open-ended feedback first, and then leveraged sentence completion [68] regarding first impression, the utility, usability, and user experience of the test condition, as well as its capacity to meet the needs and desired improvements. See the conversion guide and user study guide provided in supplementary material [45].

B. Results

Designing in real size was appreciated by participants, it enhances the understanding of object dimensions, placement, and the use of space ("*the model on a large scale gives more precise impression of the objects' size, of their position and the use of space.*" p1, "It is great to see the result directly in real size on the targeted display." p2).

1) General observations:

a) General observation about Miro: In Miro, the main menu that allows to create objects has a fixed position on the left side of the workspace and is centered vertically. The context menu for a selected object is displayed in the vicinity of the object, as shown in Figure 9. Nobody complained about the position of the two menus. The context menu allowed to modify properties and perform actions on the selected object, such as deleting or duplicating it, while the main menu on the middle-left side allowed the creation of objects ("*It is convenient that a part of the functionalities [object properties] are placed in the working zone.*" p2). The only exception to the above is for p2 in condition touch on WSD-IA ("*It's not working well.*" p2, "*It's not comfortable.*" p2). We however note that the mock-up to reproduce mainly used the leftmost parts of the WSD. Had UI elements from the mock-up been placed in a more central position (or even towards the right side of the WSD), it is likely that the position of the main menu

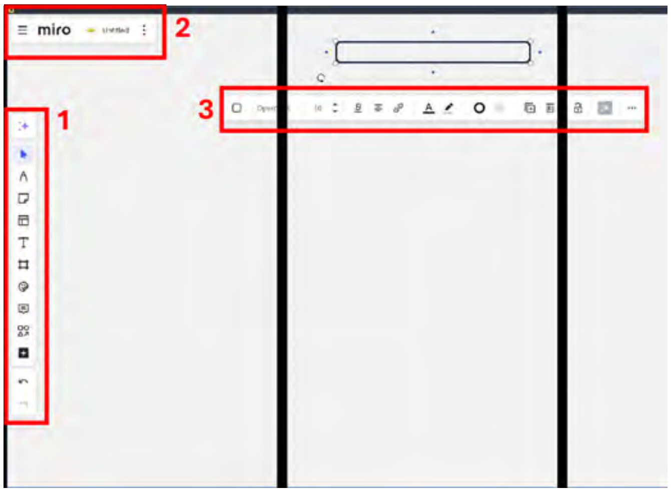


Figure 9. 1) shows the left menu used to create objects, 2) is the menu used to set some parameters for the workspace or go back to the list of projects in Miro, and 3) is the context menu of the selected object.

would have been more of an issue. The lack of complaints on the menu placement could also be due to the fact that the mock-up only contained a moderate number of objects that the participants must create. Another explanation could come from a strategy used by both participants to mainly rely on duplicating an existing object then completely modifying its properties, rather than creating a new object using the menu. This behavior could also be a reaction to the problematic placement of the menu. The context menu was used a lot, even though it sometimes stopped working for no apparent reason. The top menu, which allows users to set some parameters of the workspace, was only used to change the background color of the workspace.

The most critical aspect of Miro noted by both participants is that it does not provide a rich objects library. There are for example no chart elements ("*The available objects are very basic, so everything must be laboriously drawn by oneself.*" p1). Both participants complained that the minimum font size is blocked at 10. It was observed that, when an object was created (or duplicated) partially outside the current viewport (the portion of the workspace that is visible on the WSD), then Miro adapted the view to include that and all the objects ("*The zoom level changes when I copy/paste something out of the view, to put everything in view.*" p2). This disturbed the participants. During the session on WSD-VW with p1 and the touch condition, a modal popup window opened in the middle of the WSD, blocking all interactions. It took p1 a whole minute to notice it, which was frustrating. Both users liked the ease of changing the background color of the workspace.

b) Participants' strategy: The participants often used a strategy consisting in subdividing the mock-up in panels or zones, using a lot of copy and paste actions, and working on main structural elements before tackling the details, for all conditions. With both the touch and the keyboard conditions, p1 started by creating objects at the center level, to interact at comfortable height. He then panned the mock-up to move the objects up. P2 also used this strategy but only when interacting with touch ("*It is practical to be able to pan the mock-up, because I was able to work in the middle of the screen at the beginning, and then move everything up.*" p2).

c) Task duration: One-hour sessions were judged acceptable, but chairs were deemed mandatory for longer sessions.

2) Interacting with a wireless keyboard equipped with a touchpad: In WSD-IA, p1 left the keyboard on the table and turned his head and torso to see what he was doing, particularly when he created the last panel of the mock-up, which was behind his back, see Figure 10.a. Once, he turned around the table without moving it. In WSD-VW, p1 always left the keyboard on the table including when he moved (five times) to the extreme right to change the zoom level from the zoom menu (at the opposite end of the WSD, in the bottom right corner) or to create the last panel of the mock-up. In this case, p1 rolled the table until he was facing the zoom menu or the new work area (see Figure 10.b). P2 noted that she moved more using the keyboard with WSD-VW than with WSD-IA. In WSD-IA, p2 remained at the table and turned it to face the work area (see Figure 10.g and h). P2 mainly leaned over the table (see Figure 10.d), but she often held the keyboard on her arm during the last half of the session. P2 kept the keyboard in her hands most of the time when doing modifications like creating an object, moving or resizing it, changing its properties or zooming and panning. To enter text, p2 mostly put the keyboard on the table. She showed some signs of physical fatigue halfway through the session (movements with the neck, massages of the neck, see Figure 10.e. In WSD-VW, p2 moved the table with the keyboard on it to face the panel she was working on, and halfway through the session, sat down for 15 minutes (see Figure 10.f). But p2 also often stood up while holding the keyboard in her hands. Occasionally, p2 stepped back to get an overview.

In WSD-IA, p1 started by creating the left panel (Tasks), then he created the title and subtitle of each panel. He looked for some chart objects or chart icons. As he did not find any, he instead went on to focus on creating the sliders. In WSD-VW, p1 adopted another strategy. He first created one big blue rectangle to serve as background for all panels. Then, he created the other objects on top of it. This choice generated difficulties later on, when trying to select objects without selecting the background shape. He created the titles in the middle of the WSD, then panned the workspace to place them at the correct height. He built the interface panel by panel from the left to the right. He sometimes zoomed in to work on details (e.g., icons). He had some trouble grouping and ungrouping objects. In WSD-IA, p2 started to design objects directly where they were supposed to be placed. After creating the titles, p2 designed panel by panel from left to right, before coming back to the charts and sliders to add some details. At the end, she added blue boxes to create the background of each panel. In WSD-VW, p2 created the objects at a lower position than their final placement, then panned the workspace to adjust the height. After first creating the titles, she then designed the remaining objects panel by panel from left to right.

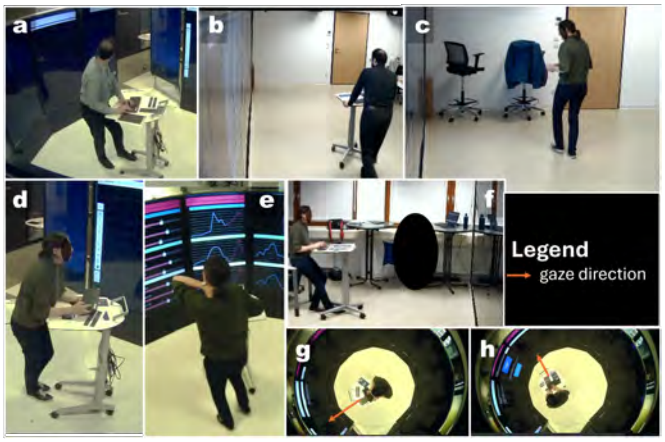


Figure 10. Using Miro with keyboard+touchpad. a) P1 turns his torso to look behind in WSD-IA. b) P1 moves the table with the keyboard on in WSD-VW. c) P2 carries the keyboard in her hands in WSD-VW. d) P2 leans on the table and tilts the keyboard with her hands in WSD-IA. e) P2 massages her neck in WSD-IA. f) P2 sits down in WSD-VW. g) and h) p2 turns the table to face the work area.

P2 used keyboard shortcuts a lot, to delete, copy, paste, but also to create new shapes, text boxes, or zoom in and out. P2 and p1 used keyboard arrow keys to place objects. This facilitated a precise placement of objects ("*I tried to be accurate with the position of the objects from the paper mock-up, taking into account the position relative to the various screens of the wall.*" p1). It also allowed participants to be more precise in the layout of the mock-up itself relative to the workspace. ("*With the keyboard, positioning the scene correctly is easier.*" p1). The keyboard was also appreciated for entering text. P2 also found it easier to draw graphs curves using the keyboard touchpad than with the tablet or in the touch condition.

The participants disliked the touchpad because it lacked precision ("*It was OK with the touchpad, but for smaller things it would be better with a mouse.*" p2). P1 complained about triggering actions inadvertently, such as an unintended pinch leading to a change of zoom, but unwanted panning, and selecting/deselecting actions also occurred ("*Of all the modalities tested, the keyboard was the most frustrating one, due to the touchpad's extreme sensitivity. As a result, this modality is difficult to use alone, it would be better to complement it with other modalities.*" p1). That's why participants asked for an option to change the zoom level independently of the panning. P1 and p2 in general prefer to use a mouse instead of the touchpad. It however remains unclear how a mouse would be dealt with when a participant walks around with the interactive device in hand, since a mouse might be difficult to use without being placed on the table. P2 also used some shortcuts to avoid using the touchpad, such as those enabling the creation of a new shape or a new text box at the cursor's position ("*But it's not nice with the touchpad, so I tried to avoid using it, and used shortcuts instead.*" p2). The participants were uncomfortable when inputting text ("*The table height was too low, which hurt my wrist. I tried to raise it, but it was already at the maximum height. In the end, I carried the keyboard to be more mobile, more flexible and to avoid wrist pain.*" p2) The workspace moved a lot inadvertently for both participants, who wished they could lock the zooming and panning.

3) Interacting using direct touch on the WSD: P1 preferred not to move the table. He mainly left the paper mock-up and the keyboard on the table ("*I didn't move the table, I preferred to have the papers in my hand in front of the display when I really need them. I didn't want to be cluttered with the table.*" p1). P2 preferred to move the table closer to the WSD to let the paper mock-up on it and to have quick access to the keyboard when needed, while maintaining access to the left menu on the WSD. She often moved the table to follow her process of working from left to right, but also sometimes pushed the table out of her way (see Figure 11.b). We also observed that both participants stepped back from time to time to get an overview (both participants did so for WSD-VW and only p2 for WSD-IA; "*We can't check the exact position of the created objects; to do this we need to move back.*" p2). They both knelt down a few times in WSD-VW to interact with the zoom menu, and p1 also did so to quickly modify some objects ("*I had to kneel down, that was more physically demanding than with other modalities.*" p1, see Figure 11 g and h). Touch was the modality requiring the most movement. Some physical fatigue arose, particularly in WSD-VW for p2 where she worked closer to the WSD ("*It hurts my neck when I am so close to the display and I look up. And it is warmer when we are close to the screens.*" p2).

We also observed that p1 and p2 first created the top objects of the mock-up at arm's height (vertically centered on the WSD), then moved the workspace upwards to correctly place the different elements "*I moved [panned] the designed interface so that it was at my height.*" p1, see Figure 11 a, b and c). On WSD-VW, both participants made use of the zoom in and out capabilities several times to design the smallest objects and then return to the initial view (see Figure 11 e then f). P2 noted that it was tiring to work at the top or the bottom of the WSD, especially when using the zoom widget at the bottom right (see Figure 11.g). They both created the titles first and then worked panel by panel. This eased the interactions and decreased the difficulty of working on such a huge surface by staying in a limited area and decreasing the amount of physical movements ("*I designed panel by panel, so I had the feeling of working only locally.*" p1).

A notable difference between WSD-IA and WSD-VW is that the participants moved back more often and further back, to get a bigger picture with WSD-VW than with WSD-IA ("*I needed to move back to have a global view.*" p1, when interacting on WSD-VW). It was also noted by p2 that touch on WSD-VW was more tiring than on WSD-IA. This could be explained by the fact that in WSD-VW, the zoom widget was far from the users when they stood in the main working area. To use it, they were forced to kneel down and press buttons, which is uncomfortable. However, they used the zoom widget more often on WSD-VW than on WSD-IA (only one time on WSD-IA, by p2), which might be due to the issues encountered

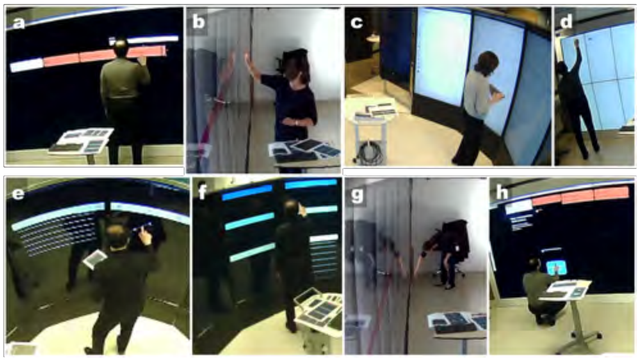


Figure 11. Using Miro with touch. In a) b) and c) participants create the objects at the WSD center. e) shows p1 zooming in and panning to design details, and then in f) repositioning the workspace correctly. In a) and f) p1 leaves the table in the middle. In b) p2 puts the table closer. d), g) and h) show moments when the participants interact too low or too high.

with touch-based zooming and panning actions, which were sometimes too fast and sometimes laggy ("*It is more tiring than with WSD-IA because of the zoom button at the bottom right.*" p2). Overall, both participants had the feeling that touch worked better on WSD-VW than on WSD-IA.

The touch modality helps the most with the sense of space. It is also more instinctive and rapidly allows to place the created objects near their target location, even if the positioning is rarely immediately perfect ("*For some actions it is nice to do them directly on the display, e.g., when I add a text box, I can place it directly where I want.*" p2). The feeling of working directly on the interface was liked by participants ("*We can change directly on the display.*" p2).

Many issues appeared with touch on both WSDs. The sensitivity of the touch-based gestures was not adapted; sometimes it was too sensitive, sometimes not responsive enough. The technology used to recognize touch events was questioned ("*The touch needs to be improved.*" p1). Indeed, participants wondered whether the unexpected zooms, pans, and deselections/selections were triggered by pre-touch when a finger, sheet of paper or sleeve inadvertently entered the infrared detection zone of the touch frames (e.g., pointing a finger while thinking "*I wonder if it is due to a finger entering the detection zone of the infrared frame.*" p1, "*being very close to the display is sometimes enough to be detected as a touch.*" p1, "*It was disturbing when my pointing finger was close to the surface, because I was thinking before acting and then a touch was detected.*" p1). Both participants had trouble placing and resizing objects, particularly for the first object with this modality ("*I still have some issues with the touch behavior, notably the resizing of the objects and text fields.*" p1). This was particularly the case when objects were tiny ("*When the manipulated objects are too small, it is impossible or very hard to modify them, like with the sliders or the curves.*" p2). Furthermore, grasping tiny objects such as the handles used for resizing objects is complex and sometimes even impossible (p1 & p2). Modifying tiny objects therefore requires zooming in, and it then becomes difficult to readjust to the right zooming level and to position the mock-up in the workspace afterwards ("*When objects are very small, like the one composing the video camera icon, it was complicated to position them. So, I zoomed in. Then, I used the zoom menu to go back to a 100% zoom level.*" p2). Hence, the participants asked for an option to restore to the initial view, or at least to be able to go back to a given fixed view of the workspace.

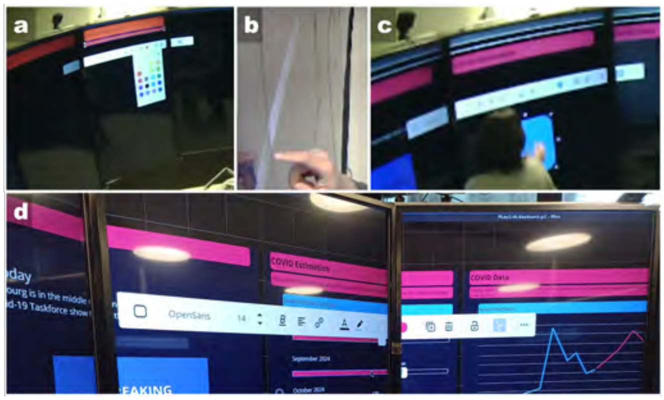


Figure 12. Bezel issue on WSD-IA (a, c and d), and on WSD WSD-VW (b).

Positioning the objects on different layers (z-index) was also problematic. Ensuring that each object was in the right layer was sometimes tricky. But, the main issue was that, when participants wanted to select an object in the foreground, a background object was often selected instead ("*When we are drawing objects composed of simple forms, it is complicated to specify what should be in the front/background.*" p1).

The amount of unexpected zooming, panning, and deselecting/selecting actions hampered work progress. In reaction, both participants had to frequently zoom and pan. They asked for the possibility to perfectly re-center the mock-up and resize the workspace based on the WSD dimensions ("*I need a button to rescale the scene.*" p1).

Screen bezels between the tiles composing the WSDs made the touch interaction harder. It was difficult to drag an element across screens because the bezel in between would interrupt the gesture ("*Objects need to be large enough to be moved across screens. Smaller bezels between two screens would improve the touch reliability.*" p1). Sometimes, the bezel occluded partially the options in the context menu (see Figure 12). Participants worked around it by first moving the selected object to ensure that the whole context menu was visible, then repositioned the object in the right position. Bezel-related issues were more raised by the participants with WSD-IA than with WSD-VW, likely because they are wider for that WSD.

Drawing straight lines was not possible using touch (p1 & p2 "*Drawing a straight line is complicated with touch because of finger tremor.*" p1).

While typing with the keyboard was appreciated, using an additional device poses a classic parking problem or requires the user to carry it around ("*The physical keyboard is better than the virtual keyboard on the tablet, and I was able to input* *more text. But the fact that the keyboard is placed on the table is not comfortable. Why not searching for a solution to have the keyboard closer without having to hold it in the hands, like attaching it to the left arm?*" p2).

4) Interacting on a synchronized tablet: Both participants moved very little during their sessions with the tablet. In WSD-IA, p1 and p2 stayed at the table and mostly did not move it. They both held the tablet during about half of the session, but p1 did so in the first part (see Figure 13.b), while p2 rather did that in the second half. They both changed their holding strategy because of the pain they felt in their arm (p1, see Figure 13.e) or neck (p2). They both showed signs of physical fatigue in the second half of the session (see Figure 13.d). P1 started the WSD-IA session with the tablet in his hand, then put the tablet on the table, complaining about the weight of the tablet. After a while, since the position at the table was not comfortable either, p1 took the tablet back in his hand. For WSD-VW, p1 left the tablet on the table throughout the session. None of the participants moved the table extensively, and they did always face the area they were working on. For instance, p1 created the rightmost panel on WSD-VW while staying at the table that was placed at the left of the WSD, as shown on Figure 13.b. P1 also sometimes held the tablet to move objects (e.g., title and subtitle of the rightmost panel) from a screen in his back to a target position he was looking at, without gazing at the objects during their movement. When the tablet was placed on the table, they both put their other hand (the one they were not using for interaction) on the table (as shown in Figure 13.a and g), except for p2 when she was entering text and therefore needed both hands (see Figure 13.f). P2 changed the tablet orientation from landscape to portrait in both WSD environments, for the whole second half of the session for WSD-IA and during 15 minutes in the beginning of the second half of the session for WSD-VW, before switching back to landscape in that case. P2 often tilted the tablet.During the WSD-VW session, p2 sat down on a table for the first fifteen minutes of the session (see Figure 13.c), then on a chair until the end of the session (see Figure 13.h). With WSD-VW, they both moved a little to check results on the WSD, leaving the tablet on the table. They both looked at the tablet for the creation and modification of objects, and then regularly checked the results on the WSD. As for the placement of objects, they partially looked at the tablet, mainly when the titles had already been created (as they then had a reference), but also looked at the WSD, although the lag between updates on the tablet and their effect on the WSD impeded these checks. They both opted to directly position the objects precisely on the WSD. They both also worked with the tablet by zooming in on part of the working area.

The most interesting aspect of this modality is that the viewports of the tablet and that of the WSD are dissociated. This allows the user to zoom in considerably on the tablet while keeping the view on the WSD unchanged. This was noted as the main advantage of the tablet modality by both participants ("*Dissociating the viewport of the tablet and that of the wall, especially for the zoom and pan*" was pointed out as the reason that makes the system easy to use by p1, "*The decoupling between zoom and pan on the tablet and on the wall is interesting. As the tablet is tiny, I needed to zoom a lot.*" p1 "*The workspace on WSD-VW never moved, this is really comfortable.*" p2).

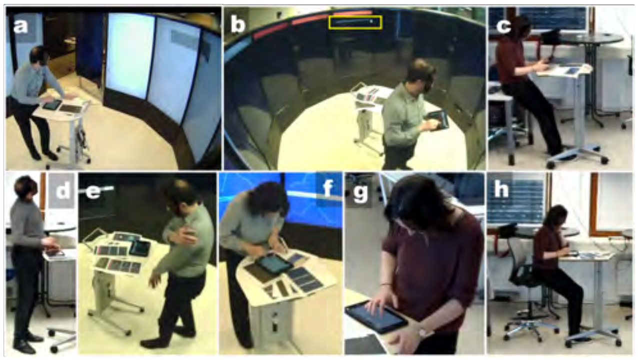


Figure 13. Using Miro with a tablet. a) and g) participants placed the tablet on the table and their non-interactive hand, b) p1 holds the tablet in hands, c) and h) shows p2 sitting down on a table and on a chair, d) and e) shows that p1 is uncomfortable and stretches f) shows p2 entering some text with two hands with the tablet placed on the table, in b) the yellow rectangle shows the object manipulated by p1 that is in his back.

This zoom in also helped to place elements more precisely (as touch is more reliable on the tablet than on the WSD) ("*With zoom [on the tablet], it becomes possible to position smaller elements.*" p2). The ability to zoom in considerably cancels out the fat finger effect observed with the touch modality (p1) [69]. Creating small objects was experienced differently by each participant. For p1, the ability to zoom on the tablet without affecting the view on the WSD enabled him to work on the smaller objects and their details. However, p2 still found the tablet inadequate to design tiny objects ("*I avoided working on the very small details with the tablet, it was not adapted.*" p2)

P2 appreciated a lot the tablet condition, particularly on WSD-VW ("*It was with the tablet that I was able to do the most. It works best.*" p2).

The most negative aspect regarding the use of the tablet was the delay between actions performed on the tablet and their effect becoming visible on the WSD (p1 & p2, "*The tablet seemed slow, it didn't respond right away." p2, "There is a delay between moving something on the tablet and seeing it move on the WSD.*" p2). P1 additionally observed disconnections on the tablet at times.

Both participants found the tablet rather heavy, which made it difficult to carry it around. They therefore put it on the table, but the ergonomics of the table were not good enough, which led to pain in their wrist, neck, back and shoulders. The pain levels reported by the participants were more intense with WSD-VW. We note that the table was not exactly the same as that of WSD-IA, which could explain the increase in complaints ("*The tablet is heavy to carry. So, carrying it in my hands didn't suit me. So, I put it on the table. But in this position, my shoulder and neck muscles were hurting. I should have raised the table up, but I didn't think about it.*"

p1, "*I had a bit of pain in my wrist due to the position while interacting, especially while entering text.*" p2).

Positioning the objects precisely with the tablet was complicated ("*You can't place [the objects] as precisely as you would like.*" p1). Grouping objects was not easy either, using touch input on the small surface of the tablet. The guiding and snapping lines to help with aligning objects were visible only when interacting on the tablet. P1 would have preferred to see them on the WSD as well.

The working surface on the tablet is small, especially when the virtual keyboard is opened ("*But it is very small: the visible surface is very small, and the virtual keyboard takes a lot of the available space.*" p2). For this reason, p2 changed the tablet's orientation from landscape to portrait, to see the workspace better when the virtual keyboard was opened.

Participants deemed drag and drop difficult on the tablet, so was multiple selection. The drag and drop of an object was in fact accelerated depending on the distance on the tablet. However, as the surface is small, the acceleration quickly becomes too pronounced, which disturbed p1 ("*When I moved an object, it seemed like the speed of moving was changing depending on the distance.*" p1).

It was also noted by p2 that the available colors on the WSD and the tablet seemed slightly different, making the selection of the right color harder ("*It was harder for me to find the right colors. Maybe it's because they are not exactly the same on the tablet and on WSD-VW, or perhaps they are not perceived the same way.*" p2).

Last but not least, both participants did not like to type text with the tablet; they preferred the physical keyboard. The predictive text input of the tablet's virtual keyboard was not working well. Indeed, after p1 copied a text box, deleted the text in the new text box, and started to input new text, the predictive text input proposed a word starting with the first letter of the previous text ("*It's a weird behavior, and it poses a problem with the strategy I used for copying/pasting and then modifying.*" p1, "*Typing is more painful than with a keyboard. But with a more efficient automatic text completion system, it would be faster. But, here, it is not efficient enough. When I duplicated a text box and then deleted its text to enter new text, it seemed that the system was keeping the first letter in mind, although it was deleted, as it was still considered by the text completion system.*" p1).

To improve the use of the tablet, both participants suggested using a stylus for more precision, and a keyboard, mainly to facilitate text entry ("*A stylus to better move the small elements and a physical keyboard.*" p2). They also suggested using a wider tablet and would have liked a more ergonomic table.

5) Suggested improvements: Participants proposed different features and options to improve the 1:1 scale design on a WSD.

- an option to lock and unlock zoom on WSD (p1 & p2);
- an option to lock and unlock panning on WSD (p1 & p2);
- a button to center & zoom the design within the WSD viewport (p1 & p2);
- an enriched objects' library (p1 & p2);
- an improved touch detection system (p1 & p2);
- more flexibility regarding the font size (p1 & p2);
- the ability to move the context menu to work around occlusions by the bezel (p1 & p2);
- freehand drawing with an AI-based conversion of hand drawings into standard UI elements ("*It would be interesting to be able to draw by hand, to make sketches.*" p1, "*Hand drawing with an AI that recognizes drawings and normalizes drawn shapes.*" p1, "*Some AI assistance, which gets what I'm doing and enables me to go faster.*" p1);
- speech recognition to dictate text and for simple commands ("*As well as using some vocal commands, for instance to duplicate an object or change the color*" p1);
- limiting the number of sub-menus, particularly in the context menu, to 2 or 3 at most (p1);
- limiting pop-ups as much as possible; those that cannot be avoided should not appear in the middle-center part of the screen, but instead appear in the work area. (p1);
- a smart feature that recognizes series of actions that are repeated, to ask the user whether and how many times to repeat them ("*a smart function that, for example, if we repeat the same gesture [action] several times, then the system asks if we should repeat it again and how many times, e.g., for the labeled ticks along sliders.*" p1);
- a wearable keyboard (p2);
- supporting mid-air gestures ("*And maybe find a solution to move less [than with touch], like using mid-air pointing instead of touch, to support working from a distance.*" p2)
- adding specific actions for each tile composing the WSD, such as changing that screen's background ("*It would be interesting to be able to change the background for each screen*" p2), that would require to declare the tiling configuration of the WSD.

VII. Discussion

In comparison to the first user study, the sessions of the second one lasted longer (the full hour in all cases) and participants managed to advance much further in the creation of the mock-up. Hence, we can say that with Miro, participants could work more efficiently, and this tool can be considered as better adapted for the design in 1:1 scale on WSDs than Figma. Our observations have shown that this improvement was mainly due to the main menu being placed in the middle part on the left side, and the context menu directly next to the selected object that allows to select properties. User feedback further revealed that the context menus were appreciated for offering quick access to common actions (e.g., duplicate, delete, change the z-index, modify object properties). The most liked functionalities were the ability to duplicate objects, to make multiple selections, to position objects in different layers, and to modify the background color. As compared to the first user study conducted with Figma and its right-anchored menu, the physical fatigue was considerably reduced.

Despite this general improvement over Figma, the second user study also showed that some issues still remain with Miro, such as the position of the zoom widget placed in the bottom right, which was disliked by all participants.

In addition, the second user study has shown that the interaction modality and the WSD configuration has impacts onto the experience of the user, which we will discuss further in the following sections.

A. Benefits and issues with each modality

All three interaction modalities used in the user studies were deemed familiar by both p1 and p2, which made them easier to interact with the WSDs ("*It resembles the interaction modalities that we know on the desktop. It embraces the great metaphors we are accustomed to.*" p1, "*It uses standard interaction methods.*" p2). However, we found that in each of the conditions, participants faced issues and appreciated other features. Based on our observations, we categorized them into pros and cons, described below and summarized in Table IV.

P1 preferred the touch modality for the sense of acting directly on the WSD, then placed the tablet as second choice, with the keyboard last ("*My favorite modality is the touch, although I experienced some glitches that caused problems.*" p1, "*With the keyboard, positioning the objects is more precise and entering text is easier, but with the tablet I have better control on the zoom level on the display, and no unwanted zoom.*" p1). P2 preferred the tablet, then ranked the keyboard second owing to the ability to use shortcuts, and ranked the touch modality last. Touch was the most tiring interaction modality for both participants, because it required more movement. Concerning the keyboard, participants would have liked to use it with a mouse rather than a touchpad. As for the tablet, participants would have preferred a larger tablet, and would have liked to use a stylus as well as a physical keyboard.

When an interaction device was used in the given condition, p1 generally preferred to leave it on the table. When movement was necessary, he preferred to move the table, e.g., when accessing the zoom menu at the right bottom on WSD-VW ("*I preferred not to carry the keyboard in hand for comfort. I preferred to move the table, that was less painful.*" p1). In such situations, p2 preferred instead to carry the device around for more flexibility and to avoid wrist pain linked to entering text at an uncomfortable angle ("*I carried the keyboard with me in order to be more mobile, more flexible and to avoid wrist pain.*" p2, "*Having the tablet on the table was eventually an uncomfortable position for the neck, so, in the end, I preferred to hold the tablet in my hand.*" p2). Ultimately, both users complained of physical fatigue and that the positioning of the interaction devices was a pain-inducing issue. Interaction devices also divided the participants' attention between the device and the WSD, typically when entering text ("*[I] divided my attention between the display and the keyboard.*" p1).

Our results also show the main issues related to each interaction modality. In the keyboard + touchpad setting, the touchpad was too sensitive, which resulted in erratic movements of the workspace and created frustration. Also, typing text induced wrist pain due to the uncomfortable typing position, but this may be solved with a tilting table. Direct touch on the WSD induces physical fatigue and pain in the arm and neck. It lacks precision, particularly in the manipulation of tiny objects, and requires a lot of zoom and pan interactions. Some issues may be related to the touch hardware we are using, as it lacked robustness according to the participants. Finally, the tablet weight induces pain in the arm and neck, that could be alleviated by a tilting table, the working surface is too small, and a fat finger effect was observed, the long-range drag and drop is difficult, and lastly, there is a synchronization latency between the tablet and the WSD.

Overall, the participants would have preferred to have the ability to choose and switch between several interaction methods throughout the design session.

B. Difference between both WSDs

Our observations also indicate some differences between the two WSD settings. First, bezel issues were raised more frequently with WSD-IA than WSD-VW, likely because they are wider on WSD-IA, even if they are more numerous on WSD-VW.

The position of the zoom widget was disliked by participants on WSD-VW, although they did not comment on it on WSD-IA. This is probably due to the fact that it was displayed in the back of the participants on WSD-IA, and went unnoticed. Only p2 used these options on WSD-IA and only once, with the keyboard modality.

The physical ergonomics play an important role and must be handled with care. The height-adjustable and movable table in WSD-VW was liked and used, even if it was not ergonomic enough. Indeed, the participants were quite tall and the maximum height of the table in WSD-VW was insufficient for them. P2 adapted the table height in WSD-IA for more comfort. Participants wished they could tilt the table, especially with the keyboard and the tablet conditions to avoid wrist, shoulder, neck and back pain. P2 sometimes sat down in WSD-VW ("*I was not too tired, but maybe it was because I sat down.*" p2 when using the keyboard on WSD-VW). But, none of the participants sat in WSD-IA, likely because available chairs were behind the screens, i.e., outside the participants' field of view. It was noted by p2 that being able to sit down in WSD-VW was less tiring when using the keyboard and the tablet, and offered better positioning for entering text.

Finally, the space around WSD-VW was appreciated ("*It was nicer with [WSD-VW] than with [WSD-IA] because there was more space, and I sat down. I was able to arrange my desk better.*" p2) and led to a better user experience, likely because participants realized they could use a chair and sit down.

C. Implications for design

Based on our results from the two user studies, we can propose several **design guidelines** for creating a WSD design tool:

- Main actions should be reachable from the work area (e.g., creation of a new object, selected object properties, deletion, layering management).

TABLE IV. Pros and cons with each interaction modality used to design at scale 1:1 on a WSD.

	Keyboard & Touchpad	**Touch**	**Tablet**
Benefits	- Precise positioning (using arrow keys) - Easier text entering - Use of shortcuts - Lighter than a tablet	- Better sense of space - Instinctive, direct, and immediate	- Dissociated viewports of tablet and WSD, allowing two distinct views, - Easier work from a distance, reducing physical fatigue as it is possible to sit down - Easier work on tiny objects
Issues	- Touchpad not appreciated - Not comfortable and wrist pain - Workspace moves a lot - Unintended action triggering	- Physical fatigue - Arm and neck pain - Not so robust - Unintended action triggering - Not so precise (tiny objects cannot be done) - Requires a lot of zoom and pan	- Arm and neck pain - Weight of the tablet - Fat finger effect - Lag between the two views - Working surface is small - Long-range drag and drop is difficult - Virtual keyboard not appreciated

- Dialogue boxes should be avoided, or opened near the work area.
- All design tool elements should be legible from near and far, and actionable with the used modality.
- It should be possible to use the entire WSD surface for design and that the design tool elements do not get in the way, e.g., by making the design tool menus retractable, or movable. All context menus should be movable to avoid the bezels.
- A distinct device with a screen, like a tablet, should provide a second view for zooming in and carrying out design actions.
- It should be possible to lock the viewport of the workspace, with independent settings for zoom and pan actions.
- A possibility for resetting the viewport to a defined setting should be provided, as reopening the design at the same position.
- The system should support very large workspaces (gigapixel resolution).
- It should be possible to change the background color.
- It should be possible to duplicate (multiple) objects.
- Expert interactions, e.g., shortcuts, should be supported.
- There should be an enriched objects' library with advanced widgets, graphs, and icons to choose from.

Overall, regarding the different interaction methods, and based on our results, the best approach seems to be to **provide a mix of modalities**, including touch interaction for direct manipulation of objects on the WSD, a distinct device with a dissociated view (e.g., tablet), and a physical keyboard as efficient means to enter text.

Regarding **the design of the WSD itself**, it is better to reduce as much as possible the bezels between the display tiles. It is also important to ensure the visibility of the cursor at all times, and to facilitate long-range interactions, e.g., through pointing facilitation techniques [50].

Flat and circular displays have their pros and cons. Circular displays reduce the amplitude and the amount of movement required, text is easier to read from everywhere, but there is always a part of the screen at the back that can be invisible or forgotten, and there is less space to add furniture to make the interaction more comfortable. Flat displays, i.e., the most common form of WSDs, require much physical navigation. The user must move a lot to see the content on the other side of the screen and to get an overview. But there is more space around the display, which makes it easier to adapt it for a better user experience, for instance, by adding furniture.

To improve the user experience regarding the **space around the WSD**, we propose to provide: a height-adjustable mobile and tilting table to park interaction devices, and chairs for users to rest.

D. Limitations

Concerning the user studies, our findings come with several limitations. First, the task was limited to the reproduction of an existing UI prototype for a WSD environment, and no creative part was involved. In addition, the user studies involved only three participants who were all unfamiliar with the design tools used. The advantage here, however, was that they had no prior habits, e.g., using specific shortcuts, and were not frustrated by not being able to work as quickly as an expert would on a familiar software.

In the second user study, a clear learning effect was observed between the first (played on WSD-IA) and last sessions (played on WSD-VW). Indeed, participants were not experts at using Miro, despite having used it for other purposes than interface design. They discovered some features and faster ways of performing some actions during the sessions. This makes it difficult to compare the completion levels between them.

Finally, it must be mentioned that we encountered (unexplained) bugs with the display of the context menu in Miro, and that bezels would sometimes occlude menu options. Also, the mock-up covered mainly the left part of the display. We expect more issues to arise when users need to create objects on the right side of the display using touch (particularly for flat WSD such as WSD-VW) or the keyboard interaction methods.

VIII. Conclusion and Future Work

In this paper, we presented two user studies and a comparative analysis investigating the suitability of existing UI design tools to design at 1:1 scale on WSDs using three different interaction methods: touch, a keyboard with a touchpad, and a tablet. Our main takeaways are that (i) prototyping at 1:1 scale and being able to see live the final rendering in real time is appreciated, (ii) tablet-based interaction proved to be the most comfortable, (iii) each interaction method has its benefits and drawbacks; using a mix of modalities is promising, (iv) the spatial positioning of interaction elements on the screen, as well as the design of the physical environment, is of utmost importance to reduce physical fatigue and support efficient work, (v) among the nineteen desktop-optimized design tools analyzed, none satisfies all the criteria, but Miro is the most suitable.

So, to answer our research questions, a desktop optimized tool cannot be used as-is to design at 1:1 scale in a WSD environment. To improve the design of such a tool, we propose 12 design recommendations indicating how UI elements should be placed, and which features need to be provided, see Section VII-C. Regarding interaction modalities, the best approach for this type of design currently seems to be the synchronized tablet, but the other modalities also provide some unique benefits. A mix of interaction methods seems promising and is yet to be tested.

However, these studies involved few users and are focussed on the reproduction of a UI design. To validate the proposed guidelines, further experiments are needed.

More specifically, future work needs to further explore how 1:1 scale design can be supported by focussing on creating a new design rather than reproducing an existing one. As design is often a collaborative task, future research also needs to investigate how to manage collaboration during design sessions at 1:1 scale on WSDs. Additional questions are how modalities can be mixed and support the transitioning between multiple interaction methods in the same session. The studies from this paper enabled us to better understand the requirements for technical systems and the related problems and opportunities when using them to design for WSDs. In a longer perspective, this work is done as part of a design science methodology [70], ultimately aiming to create a design tool that is suitable for WSDs. The results presented in this paper are part of the first step, the relevance cycle, and will enable us to identify the user requirements.

IX. Acknowledgments

We thank all user study participants.

References

[1] L. Schwartz, V. Maquil, and M. Ghoniem, “Designing for and on wall-sized displays: a preliminary study with FIGMA,” in *CENTRIC 2024: The Seventeenth International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services*. IARIA, 2024, pp. 41–44.

[2] C. Parker, M. Tomitsch, N. Davies, N. Valkanova, and J. Kay, “Foundations for designing public interactive displays that provide value to users,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.

[3] M. Finke, A. Tang, R. Leung, and M. Blackstock, “Lessons learned: Game design for large public displays,” in *Proceedings of the 3rd International Conference on DIMEA Digital Interactive Media in Entertainment and Arts*, ser. DIMEA ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 26–33. [Online]. Available: https://doi.org/10.1145/1413634.1413644

[4] A. Prouzeau, A. Bezerianos, and O. Chapuis, “Towards road traffic management with forecasting on wall displays,” in *Proceedings of the ACM ISS International Conference on Interactive Surfaces and Spaces*. ACM, 2016, pp. 119–128.

[5] W. Buxton, G. Fitzmaurice, R. Balakrishnan, and G. Kurtenbach, “Large displays in automotive design,” *IEEE Computer Graphics and Applications*, vol. 20, no. 4, pp. 68–75, 2000.

[6] C. Liu, O. Chapuis, M. Beaudouin-Lafon, and E. Lecolinet, “Shared interaction on a wall-sized display in a data manipulation task,” in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 2075–2086. [Online]. Available: https://doi.org/10.1145/2858036.2858039

[7] D. Wigdor, H. Jiang, C. Forlines, M. Borkin, and C. Shen, “WeSpace: The design development and deployment of a walk-up and share multi-surface visual collaboration system,” in *Proceedings of the 2009 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 1237–1246. [Online]. Available: https://doi.org/10.1145/1518701.1518886

[8] F. Rajabiyazdi, J. Walny, C. Mah, J. Brosz, and S. Carpendale, “Understanding researchers’ use of a large, high-resolution display across disciplines,” in *Proceedings of the 2015 International Conference ITS on Interactive Tabletops &; Surfaces*. New York, NY, USA: Association for Computing Machinery, 2015, p. 107–116. [Online]. Available: https://doi.org/10.1145/2817721.2817735

[9] E. Pietriga, F. Del Campo, A. Ibsen, R. Primet, C. Appert *et al.*, “Exploratory visualization of astronomical data on ultra-high-resolution wall displays,” in *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913. SPIE, 2016, pp. 344–358.

[10] J. Simonsen, H. Karasti, and M. Hertzum, “Infrastructuring and participatory design: Exploring infrastructural inversion as analytic, empirical and generative,” *Computer Supported Cooperative Work (CSCW)*, vol. 29, no. 1, pp. 115–151, 2020.

[11] J. Rambourg, H. Gaspard-Boulinc, S. Conversy, and M. Garbey, “Welcome onboard: An interactive large surface designed for teamwork and flexibility in surgical flow management,” in *Proceedings of the 2018 ACM International Conference ISS on Interactive Surfaces and Spaces*, ser. ISS ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 5–17. [Online]. Available: https://doi.org/10.1145/3279778.3279804

[12] M. M. Thomas, T. Kannampallil, J. Abraham, and G. E. Marai, “Echo: A large display interactive visualization of ICU data for effective care handoffs,” in *2017 IEEE Workshop VAHC on Visual Analytics in Healthcare*, Phoenix, USA, 2017, pp. 47–54.

[13] V. Doshi, S. Tuteja, K. Bharadwaj, D. Tantillo, T. Marrinan *et al.*, “Stickyschedule: An interactive multi-user application for conference scheduling on large-scale shared displays,” in *Proceedings of the 6th ACM PerDis International Symposium on Pervasive Displays*, ser. PerDis ’17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3078810.3078817

[14] J. Wall and M. Bertoni, “The decision arena: A model-centric interactive workspace for product-service system design,” in *Proceedings of NordDesign 2020*. Design Society, 2020, pp. 1–10.

[15] S. Kubicki, A. Guerriero, L. Schwartz, E. Daher, and B. Idris, “Assessment of synchronous interactive devices for BIM project coordination: Prospective ergonomics approach,” *Automation in Construction*, vol. 101, pp. 160–178, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580517304284

[16] L. Lischke, L. Janietz, A. Beham, H. Bohnacker, U. Schendzielorz *et al.*, “Challenges in designing interfaces for large displays: The practitioners’ point of view,” in *Proceedings of the 11th NordiCHI Nordic Conference on Human-Computer Interaction*. ACM, 2020, pp. 1–6.

[17] I. Belkacem, C. Tominski, N. Médoc, S. Knudsen, R. Dachselt *et al.*, “Interactive visualization on large high-resolution displays: A survey,” in *Computer Graphics Forum*. Wiley Online Library, 2022, p. e15001.

[18] L. Chen, H.-N. Liang, J. Wang, Y. Qu, and Y. Yue, “On the use of large interactive displays to support collaborative engagement and visual exploratory tasks,” *Sensors*, vol. 21, no. 24, p. 8403, 2021.

[19] C. Ardito, P. Buono, M. F. Costabile, and G. Desolda, “Interaction with large displays: A survey,” *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, pp. 1–38, 2015.

[20] L. Schwartz, V. Maquil, and M. Ghoniem, “The challenges of designing for large interactive displays,” in *IHM23 Adjunct*. ACM, 2023, pp. 1–6.

[21] M. Sinaei Hamed, P. Kwan, M. Klich, J. Aurisano, and F. Rajabiyazdi, “The elephant in the room: Expert experiences designing, developing and evaluating data visualizations on large displays,” *Proceedings of the*

ACM on Human-Computer Interaction, vol. 8, no. ISS, pp. 301–329, 2024.

[22] Figma. FIGMA design. Accessed: 2025.05.06. [Online]. Available: https://www.figma.com/design/

[23] Miro. Welcome to the innovation workspace. Accessed: 2025.05.06. [Online]. Available: https://miro.com/

[24] C. Snyder, *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann, 2003.

[25] B. Bailey, J. Biehl, D. Cook, and H. Metcalf, "Adapting paper prototyping for designing user interfaces for multiple display environments," *Personal and Ubiquitous Computing*, vol. 12, pp. 269–277, 2008.

[26] I. Avellino, C. Fleury, W. E. Mackay, and M. Beaudouin-Lafon, "Camray: Camera arrays support remote collaboration on wall-sized displays," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 6718–6729.

[27] D. Fallman, M. Kruzeniski, and M. Andersson, "Designing for a collaborative industrial environment: the case of the ABB powerwall," in *Proceedings of the 2005 conference on DUX Designing for User eXperience*, 2005, pp. 41–es.

[28] L. Lischke, S. Mayer, K. Wolf, N. Henze, H. Reiterer *et al.*, "Screen arrangements and interaction areas for large display work places," in *Proceedings of the 5th ACM PerDis International Symposium on Pervasive Displays*, 2016, pp. 228–234.

[29] M. Chegini, S. Lin, D. J. Lehmann, K. Andrews, and T. Schreck, "Interaction concepts for collaborative visual analysis of scatterplots on large vertically-mounted high-resolution multi-touch displays," in *Forum Media Technology*, 2017, pp. 90–96.

[30] R. Langner, U. Kister, and R. Dachselt, "Multiple coordinated views at large displays for multiple users: Empirical findings on user behavior, movements, and distances," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 608–618, 2018.

[31] A. Bezerianos, P. Dragicevic, and R. Balakrishnan, "Mnemonic rendering: an image-based approach for exposing hidden changes in dynamic displays," in *Proceedings of the 19th ACM symposium UIST on User Interface Software and Technology*, 2006, pp. 159–168.

[32] L. Vandenabeele, H. Afkari, J. Hermen, L. Deladiennée, C. Moll *et al.*, "DeBORAh: A web-based cross-device orchestration layer," in *Proceedings of the 2022 International Conference AVI on Advanced Visual Interfaces*, 2022, pp. 1–3.

[33] E. Pietriga, S. Huot, M. Nancel, and R. Primet, "Rapid development of user interfaces on cluster-driven wall displays with jbricks," in *Proceedings of the ACM EICS Symposium on Engineering Interactive Computing Systems*, 2011, pp. 185–190.

[34] H.-J. Kim, J.-W. Kim, and T.-J. Nam, "Ministudio: Designers' tool for prototyping ubicomp space with interactive miniature," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 213–224.

[35] H.-J. Kim, C. M. Kim, and T.-J. Nam, "Sketchstudio: Experience prototyping with 2.5-dimensional animated design scenarios," in *Proceedings of the 2018 DIS Designing Interactive Systems Conference*. ACM, 2018, pp. 831–843.

[36] I. S. MacKenzie and R. W. Soukoreff, "Text entry for mobile computing: Models and methods,theory and practice," *Human–Computer Interaction*, vol. 17, no. 2-3, pp. 147–198, 2002. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/07370024.2002.9667313

[37] U. Hinrichs, M. Hancock, S. Carpendale, and C. Collins, "Examination of text-entry methods for tabletop displays," in *Second Annual IEEE International Workshop TABLETOP on Horizontal Interactive Human-Computer Systems*, 2007, pp. 105–112.

[38] T. J. Dube and A. S. Arif, "Text entry in virtual reality: A comprehensive review of the literature," in *Conference HCII on Human-Computer Interaction. Recognition and Interaction Technologies*, M. Kurosu, Ed. Cham: Springer International Publishing, 2019, pp. 419–437.

[39] M. R. Jakobsen and K. Hornbæk, "Negotiating for space? collaborative work using a wall display with mouse and touch input," in *Proceedings of the CHI Conference on Human Factors in Computing Systems*, ser. CHI '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 2050–2061. [Online]. Available: https://doi.org/10.1145/2858036.2858158

[40] F. Heidrich, M. Ziefle, C. Röcker, and J. Borchers, "Interacting with smart walls: a multi-dimensional analysis of input technologies for augmented environments," in *Proceedings of the 2nd AH Augmented Human International Conference*, ser. AH '11. New York, NY, USA: Association for Computing Machinery, 2011. [Online]. Available: https://doi.org/10.1145/1959826.1959827

[41] M. R. Jakobsen, Y. Jansen, S. Boring, and K. Hornbæk, "Should I stay or should I go? selecting between touch and mid-air gestures for large-display interaction," in *Conference INTERACT on Human-Computer Interaction*, J. Abascal, S. Barbosa, M. Fetter, T. Gross, P. Palanque, and M. Winckler, Eds. Cham: Springer International Publishing, 2015, pp. 455–473.

[42] A. Prouzeau, A. Bezerianos, and O. Chapuis, "Evaluating multi-user selection for exploring graph topology on wall-displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 8, pp. 1936–1951, 2017.

[43] V. Maquil, D. Anastasiou, H. Afkari, A. Coppens, J. Hermen *et al.*, "Establishing awareness through pointing gestures during collaborative decision-making in a wall-display environment," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, 2023, pp. 1–7.

[44] L. Schwartz, M. Ghoniem, V. Maquil, A. Coppens, and J. Hermen, "Designing at 1:1 scale on wall-sized displays using existing ui design tools - annex 1," 2025.

[45] ——, "Designing at 1:1 scale on wall-sized displays using existing ui design tools - annex 2," 2025.

[46] Google. The browser built to be yours. Accessed: 2025.05.06. [Online]. Available: https://www.google.com/intl/en/chrome/

[47] A. C. Figliolia, F. E. Sandnes, and F. O. Medola, "Experiences using three app prototyping tools with different levels of fidelity from a product design student's perspective," in *International Conference ICITL on Innovative Technologies and Learning*. Springer, 2020, pp. 557–566.

[48] Uxtools.co. UI design. Accessed: 2025.05.06. [Online]. Available: https://uxtools.co/survey/2023/ui-design/

[49] J. T. Hansberger, C. Peng, S. L. Mathis, V. Areyur Shanthakumar, S. C. Meacham *et al.*, "Dispelling the gorilla arm syndrome: the viability of prolonged gesture interactions," in *9th International Conference VAMR on Virtual, Augmented and Mixed Reality*. Springer, 2017, pp. 505–520.

[50] R. Balakrishnan, ""Beating" fitts' law: virtual enhancements for pointing facilitation," *International Journal of Human-Computer Studies*, vol. 61, no. 6, pp. 857–874, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S107158190400103X

[51] S. Phillora. (2024) 23 UX/UI design tools for better product design and UX. Accessed: 2025.05.06. [Online]. Available: https://maze.co/collections/ux-ui-design/tools/

[52] Adobe. (2024) Get started with ADOBE XD. Accessed: 2025.05.06. [Online]. Available: https://helpx.adobe.com/xd/get-started.html

[53] Axure. AXURE RP 11. Accessed: 2025.05.06. [Online]. Available: https://www.axure.com/

[54] Balsamiq. The effortless wireframing tool built for your big ideas. Accessed: 2025.05.06. [Online]. Available: https://balsamiq.com/

[55] Bubble. The full-stack, no-code app builder for everyone. Accessed: 2025.05.06. [Online]. Available: https://bubble.io/

[56] Canva. What will you design today? Accessed: 2025.01.30. [Online]. Available: https://www.canva.com

[57] Excalidraw. Excalidra. Accessed: 2025.05.06. [Online]. Available: https://excalidraw.com/

[58] Framer. The web builder for stunning sites. Accessed: 2025.05.06. [Online]. Available: https://www.framer.com

[59] Justinmind. Interaction design and prototyping tool for web and mobile apps. Accessed: 2025.05.06. [Online]. Available: https://www.justinmind.com

[60] MockFlow. Think. wireframe. brainstorm. with mockflow. Accessed: 2025.05.06. [Online]. Available: https://mockflow.com/

[61] MockPlus. Sign in to mockplus. Accessed: 2025.05.06. [Online]. Available: https://rp.mockplus.com/

[62] Penpot. Design and code beautiful products. together. Accessed: 2025.05.06. [Online]. Available: https://penpot.app/

[63] Proto.io. Prototyping for all. Accessed: 2025.05.06. [Online]. Available: https://proto.io/

[64] Protopie. #1 advanced prototyping tool for dynamic & multimodal interactions. Accessed: 2025.05.06. [Online]. Available: https://www.protopie.io/

[65] Sketch. Designers, welcome home. Accessed: 2025.05.06. [Online]. Available: https://www.sketch.com

[66] UXpin. Faster prototyping with AI component creation. Accessed: 2025.05.06. [Online]. Available: https://app.uxpin.com

[67] WebFlow. Your site should do more than look good. Accessed: 2025.05.06. [Online]. Available: https://webflow.com/

[68] C. Lallemand and G. Gronier, *Méthodes de design UX: 30 méthodes fondamentales pour concevoir et évaluer les systèmes interactifs*. Editions Eyrolles, 2015, original document in French, English title ”UX Design Methods: 30 Fundamental Methods for Designing and Evaluating Interactive Systems”.

[69] S. Voida, M. Tobiasz, J. Stromer, P. Isenberg, and S. Carpendale, “Getting practical with interactive tabletop displays: designing for dense data,” fat fingers,” diverse interactions, and face-to-face collaboration,” in *Proceedings of the ACM International ITS Conference on Interactive Tabletops and Surfaces*, 2009, pp. 109–116.

[70] A. R. Hevner, “A three cycle view of design science research,” *Scandinavian Journal of Information Systems*, vol. 19, no. 2, p. 4, 2007.