

# ProtoSketchAR: Prototyping in Augmented Reality via Sketchings

Simone Arriu, Gianmarco Cherchi<sup>LD</sup> and Lucio Davide Spano<sup>LD</sup>

University of Cagliari, Department of Mathematics and Computer Science, Italy

---

## Abstract

*Prototyping is a widely used technique in the early stages of system design, and it is an essential part of a new product development process. During this phase, designers identify the main functionalities, concepts and contents of the system without creating a fully functional system. This paper aims to discuss the development of ProtoSketchAR, a tool enabling Augmented Reality (AR) prototyping by sketching. The application has different interaction modes, depending on the performed functionality. Basically, it is possible to create 2D/3D sketches to be placed in the real environment and to manipulate them. These functionalities allow the creation of virtual elements that can be used to prototype screens of AR applications. The application is web-based so that it can be run on any device with a compatible AR browser, regardless of the operating system used.*

## CCS Concepts

• **Human-centered computing** → *Mixed / augmented reality*;

---

## 1. Introduction

Applications developed for Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR) exploit the senses of the user to obtain a more engaging and immersive experience. Each one of these technologies has its characteristic features: in Virtual Reality, the user is wholly involved in a simulated virtual world, while in Augmented Reality he/she can define an enhanced version of the real world that is obtained through the use of digital effects, sounds or other sensory stimuli provided by the technology. Finally, Mixed Reality combines both AR and VR elements, allowing the user to directly interact with digital objects. Typically, the development of AR, VR and MR applications involves developing platforms or specific graphics engines, such as A-Frame, Unity or Unreal Engine. It is immediately clear that such tools are reserved for developers with technical skills, experience in programming or in the creation of 3D content. In addition, developing complete applications using these tools is a time-consuming and costly task and requires a significant commitment from the developers. For this reason, it is crucial to find a way to define an initial primordial version of the project (prototype) without necessarily have the ability to use one of the development platforms mentioned above. The prototyping phase is essential to understand the objectives and the technical aspects to implement before actually starting with the development. There are several ways of setting up this phase, and the implementation choices depend on various aspects: budget, time and materials available, designer skills, specific requests, and so on. Generally, the prototypes can have different categories: they can be low or high fidelity, in such a way as to define the level of detail and

functionality present within the prototype, or they can be vertical or horizontal, task-oriented or based on scenario [RvHPR18].

In the case of an AR, VR or MR application, it is recommended to use a type of immersive prototyping, which allows total immersion by the user in order to "deceive" his/her senses and perceive the immersive reality as real, while allowing natural intuitive interactions. This approach typically creates a high fidelity prototype and requires more effort than a simple paper-based sketch. Although the existence of prototyping techniques more suitable for specific application contexts, this does not prevent using an approach that involves the combination of several techniques. For example, paper-based prototyping is not entirely suitable for creating AR, VR or MR prototypes, but it can be used as a preliminary technique for immersive prototyping [RvHPR18].

In this paper, we introduce *ProtoSketchAR*, a web-based application that provide a tool capable of creating low-fidelity AR prototypes. The approach used to create the prototype is sketch-based, as the user will use the device to draw his/her own sketches and position them in the real environment. Considering that the tool allows the creation of low fidelity prototypes, the user does not need to have sketching, programming or 3D modeling skills. In the following, we will discuss different aspects. The various existing prototyping techniques to create prototypes in Augmented, Virtual and Mixed Reality applications will be analyzed, showing their strengths and peculiarities. Then, we will focus on the ProtoSketchAR application development, analyzing the various screens and features of which it is composed. Finally, we will present an evaluation of the application, tested with various users, reporting feedback and aspects to be improved.

## 2. State of the art

The prototyping process is a relevant part of the design process of interactions, devices or products in general. Different digital tools for prototyping 2D applications exist, but the speed and the flexibility offered by paper-based prototypes gives them a clear advantage in this field, especially in the initial phases of a project. However, when it comes to AR, VR or 3D applications in general, paper-based approaches lose their advantage [NNYR18] since they are difficult to render on such a medium. Therefore, the design community misses the equivalent of pencil and paper for prototyping such environments, and the quest for new approaches is still open. The current solutions in the literature provide support in different use cases.

One of the available techniques for prototyping AR and VR applications is storyboarding, which is successfully employed for movies and videos, basically representing scenes in a comics-like format. For instance, McCurley [McC] applied such a technique in VR, using a starting template that puts the user at the centre of a circular terrain, annotated with the different interest areas in the field of view in a head-mounted display (HMD). This allows placing the content and considering which objects are included according to the different areas supporting the storytelling. Such an approach does not provide an immersive preview of the experience. We can find two categories of solutions for this problem in the literature.

The first type tries to keep the paper drawing approach, focusing on deriving an immersive experience from the sketch. Nebeling et al. [NM19] created a prototyping tool for AR/VR called *360proto*, which requires the designer to draw sketches on a special template sheet, containing thin guiding lines for drawing the content on a sphere texture (i.e., as in equirectangular photos). This requires modifying the drawing technique, but it allows to try the experience by simply taking a picture of the drawing and projecting it on the HMD. Interactive actions are usually supported through Wizard of Oz techniques. Further development of such approach is available is XRDirector [NLC\*20], which includes a complete role-based authoring system for creating immersive prototypes. The system assigns different roles to the people involved in the prototype creation: *actors* for animating 3D characters, *lights* controlling the scene lighting, *camera* for simulating the user's point of view changes, and *viewers* that watch to live results. Even if it requires at least three members in the team, the combination of the roles and the organization in managing the different aspects provides a great expressiveness in the scenes that a team can prototype.

The second solution relies on changing the drawing support from paper to a digital surface. Such approaches are particularly well-suited for AR or MR applications since they allow sketching directly on the visualization of the mobile camera output. The most straightforward approach is using markers for positioning contents, which are simply flat 2D drawings [PTB\*01] changing their orientation according to the marker's rotation. Xin et al. [XSS08] proposed using a set of pre-defined frames for creating 3D sketching in an ordered sequence of drawings. Such an approach complicates the drawing work since it is difficult to fit the different parts working on fixed perspectives. Other approaches apply computer vision techniques for creating 3D models from paper sketches [dSvDV09], but the automation often leads to undesired

results. More complex solutions combine 2D drawing with six degrees of freedom tracking devices for creating sketches directly in a 3D visualization. For instance, Arora et al. [AHKG\*18] created *SymbiosisSketch*, a hybrid sketching system that combines mid-air 3D sketching on Microsoft HoloLens with a 2D drawing application running on a tablet. While the opportunity of observing the sketches while immersed in the 3D environment and the coordination with a simpler 2D drawing device is valuable, creating the sketches through mid-air gestures is neither straightforward nor comfortable for the user. In addition, devices such as the HoloLens are not aimed at the consumer market, so most experiences should be designed for mobile-based AR.

Finally, an intermediate solution between static sketches and immersive prototypes is creating videos representing the experience. The designer records a video in a possible location where the experience might take place, then s/he overlays some graphical elements in post-processing that representing the virtual elements in the experience. This requires expertise in video editing. Pronto [GJSW19] tries to solve this problem by supporting sketching and drawing animations on the recorded video, supporting the immediate visualization of the result.

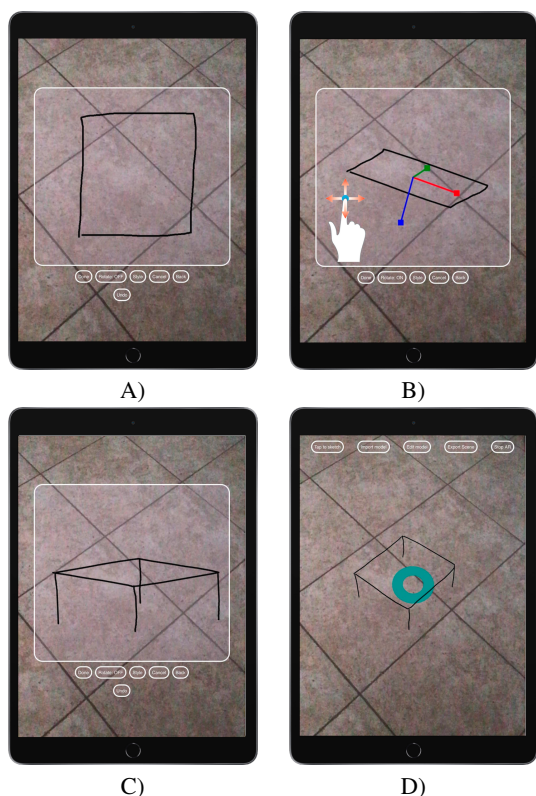
This paper tries to combine the advantages of sketching on a phone or tablet screen, similar to what is usually done on paper. At the same time, we try to support a more straightforward way for creating 3D content and positioning it in a real environment. We focus on mobile-based AR experiences, trying to combine the control of the drawing perspective and 2D drawings to create the sketch while providing the application context.

## 3. The ProtoSketchAR interface

The goal of ProtoSketchAR is to provide a rapid, low-fidelity prototyping tool for AR applications. It combines both finger or stylus sketches with stock 3D models to create disposable yet interactive AR experience representations. Low-fidelity prototypes do not require a high detail in the sketch representations (both for 3D objects and UI elements), but they should be both quick to execute and understandably represent the object or concept. Therefore, we support contextual editing of the sketch, using standard touchscreens and cameras in mobile devices. The application starts from a *home* screen, containing the visualization of the AR scene as framed through the camera. The top-level menu allows to easily access the prototyping features: 1) drawing sketches, 2) importing 3D models from a repository, 3) editing existing sketches or models, 4) exporting the scene in a *.GLTF* format.

The sketching tool provides two different modes: *classic* and *pinned*, both supporting the drawing using a stylus or the finger, but managing the projection from the 2D surface of the mobile device to the 3D world through different interactions.

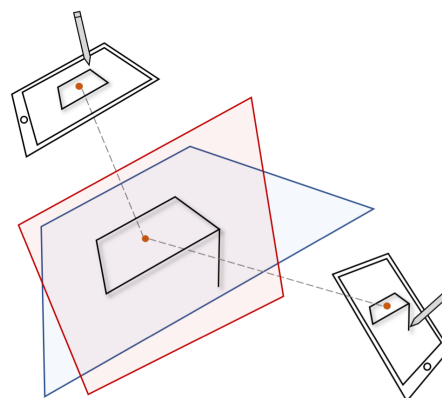
We display the *classic* mode in Figure 1. In this mode, the user draws sketches in the 2D canvas (see Figure 1-A). When a part of the sketch is finished, the rotation tool (Figure 1-B) allows to rotate all the points and lines in the sketch through the trackball metaphor. In such a way, the user can control the 3D perspective on the object, reaching a more convenient orientation for drawing specific



**Figure 1:** Sample sketching in the classic mode for drawing a table: A) drawing the table surface, B) rotating it, C) drawing the feet and finally D) place it onto the floor.

parts. Figure 1 shows a sample of this situation. The user is drawing a table and selects two different perspectives: one for drawing the surface and one for the feet. While the user rotates the object, the tool shows the axes to keep the original object's orientation. By disabling the rotation tool, the user can continue in drawing the sketch, adding elements from a different perspective (Figure 1-C). Once the sketch is complete, the user can position it in the surrounding environment, using the 3D cursor interface in Figure 1.

In the *pinned* mode, the user creates the sketches exploiting the sensors in the mobile device for changing the orientation of the drawing surface in the real world. The interaction proceeds as follows: the user “pins” the projection in the real world of the screen canvas centre. Fixing such a point, the user can move around and rotate the device for controlling the orientation of the screen canvas projection, which will be consistent with the device. Figure 2 shows the interaction sequence: once the cursor is fixed in a point (the orange point), this will become the barycenter for the user, who will move around it to draw the sketch. Firstly, the user draws the top of the table from above (blue plane), and then, by moving the device to a different position (red plane), the user draws the table feet. While drawing, the user receives two types of feedback. The first is shown on the drawing canvas. The second instead is the sketch positioned in the real environment (i.e., on the pinned plane), showing the resulting visualization *in situ*. To avoid cluttering in the drawing in-



**Figure 2:** Sample sketching in the pinned mode for drawing a table: the user interacts with the environment by pinning the centre of the scene (orange point) and then rotating the device to sketch the table from different slants (first the blue plane and then the red one).

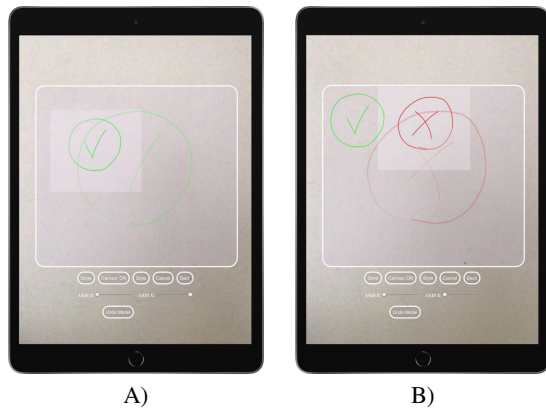
terface, the feedback on the drawing canvas is temporary, allowing the user to continue drawing having some reference points: it fades away in 3 seconds, leaving room for other sketches and the environment observation. Figure 3 shows a sample of the pinned mode feedback. First, the user draws a green check inside the bigger rectangle having a white border. As visible in Figure 3-A, the green check inside the canvas is fading, and this allows to better observe the resulting sketch in the real environment position, represented by the smaller check. After that, in Figure 3-B, the user moves the reference point for the pinned plane on the right, and s/he has drawn a red X. The old feedback about the check on the canvas is completely fade away, while the sketch in the environment remains. In addition to manually creating his/her own 2D / 3D sketches, the user can quickly insert simple 3D models into the environment using the *import* mode. As in the Sketching mode, after choosing the position in the scene using the cursor, the user can import one of the 3D models already present in ProtoSketchAR (see Figure 4-A) or a new *.GLTF* format model via URL. Once the model has been chosen, he/she clicks on the *Place* button to position and display it in the surrounding environment, as shown in figure 4-B.

Once the sketch and the models are placed in the scene, it is possible to select one of them and modify it in *edit* mode. In ProtoSketchAR we developed the classical manipulation operations as *Scale*, *Rotate* and *Translate* (as well as the removal of the object through the *Delete*). Finally, it is possible to save the scene with the sketches and models in it, in *.GLTF* format, with *export* mode. Each model is saved with its position, rotation and size in the real world as set up while using the app.

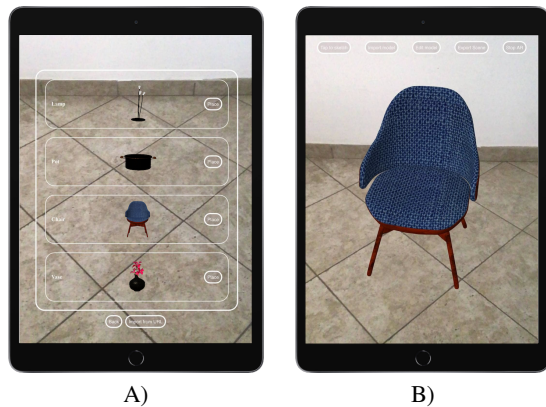
We developed ProtoSketchAR using A-Frame [afr], which provides both support for managing the AR scene graph and plane detection.

#### 4. Evaluation

In order to provide an evaluation of the ProtoSketchAR overall usability, we planned and executed a small user study. The goal was



**Figure 3:** Feedback in the pinned mode. The strokes in the drawing canvas fade away over time (part A), while the sketch in the pinned plane remains visible (part B).

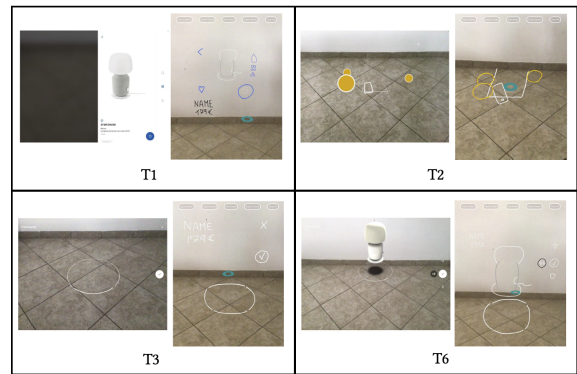


**Figure 4:** Adding a 3D model into the scene with the import mode: A) choosing a model from the list of models available in ProtoSketchAR and B) placing it into the scene.

to collect quantitative and qualitative information about the positive and the negative aspects of the proposed AR prototyping interface.

**Procedure.** We provided our participants with a case study to prototype: the *Ikea Place* [ike], a well-known mobile AR application that allows positioning virtual pieces of furniture in the user's home through AR. To avoid asking them to design the application, but only to transpose an idea they have clear into the AR prototype, we selected the screens supporting the main application tasks, and we asked them to reproduce them in low-fidelity through ProtoSketchAR. The tasks we identified are the following:

- T1: *Selection of the piece of furniture.* It is a simple view showing a picture of the object, some information such as its name and dimensions and a button for entering in the AR mode.
- T2: *Environment scanning.* The app shows an environment scanning interface for identifying a surface where to place the selected object. Some feedback elements (circles and squares) guide the user in moving the camera for reconstructing the environment.



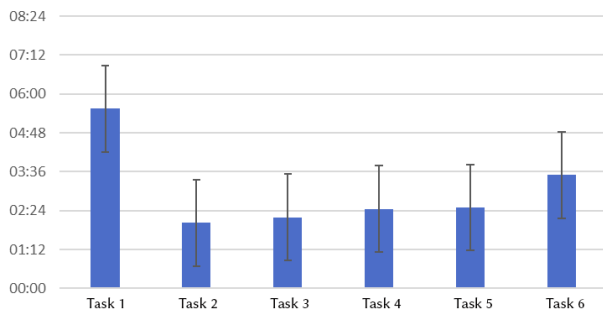
**Figure 5:** Four sample sketches drawn by the user study participants for T1, T2, T3 and T6. The left part contains the original Ikea app screen for each sample, and the right part contains the user's sketch.

- T3: *Placement feedback (positive).* The app displays a cursor for placing the object in the environment, showing positive feedback (the object fits the surface).
- T4: *Placement feedback (negative).* Same as T3, but with negative feedback (the object does not fit the surface).
- T5: *3D object placed.* The app displays a 3D object in the real environment.
- T6: *3D object menu.* The app shows some buttons for moving the 3D model, adding it to the favourite items, and removing it from the environment.

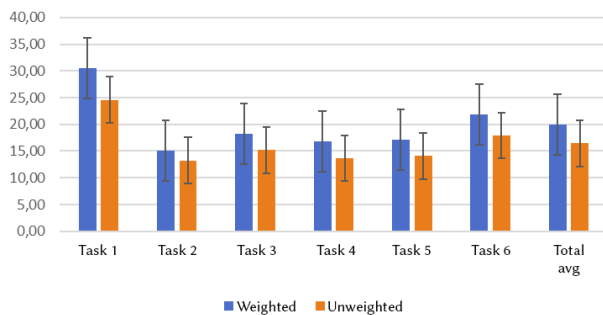
For each task, we registered the completion time and the number of errors. After each task, we asked the participants to fill the NASA TLX [HS88] questionnaire for evaluating the task load. After completing the entire test, we asked them to fill a SUS [B\*96] questionnaire and a set of rating scales about specific aspects of the prototyping interface. Finally, we collected their qualitative feedback in a debriefing session.

**Participants.** Ten people participated in the study. Their age ranged between 20 and 31 years old, having low or medium familiarity with AR applications. Before starting the test, we informed them about the study goals, the procedures and the information we were about to collect.

**Results.** All participants correctly completed all the tasks. Figure 5 shows sample sketches drawn by participants for T1, T2, T3 and T6. Each sample contains the original Ikea application screen on the left and the sketch created through ProtoSketchAR on the right. The images demonstrate that the hand-drawn sketches are enough for rapidly creating a feeling of the overall interaction, even if at a low-fidelity level. Figure 6 shows the average time spent for each task. Task 1 took more time than the others, while tasks from 2 to 6 required between 2 and 3.5 minutes. Such a difference is mainly caused by the information contained in the first screen of the Ikea app, mainly contained in standard 2D widgets, which required more time than sketching the feedback elements in AR. In addition, the participants used the first task also for familiarizing themselves with the interface.



**Figure 6:** The average execution time for each task.



**Figure 7:** NASA-TLX scores per task: average values for each analyzed task. Error bars represent the standard deviation.

We assessed the perceived workload for each task using the NASA-TLX [HS88]. The questionnaire consists of 6 questions on a 1 to 100 scale (the lower, the better), assessing different dimensions influencing the workload. We can combine such dimensions assigning the same weight to each of them (the so-called raw-TLX), or by using weights obtained by asking each participant to rank their contribution to the overall workload by comparing them in pairs. We calculated the workload index both weighting and not weighting the factors, obtaining the results in Figure 7. The factor ranking was Mental Demand, Performance, Effort, Physical Demand, Temporal Demand and Frustration. Both indexes show the same overall trend: the perceived workload is low, and the participants considered the first task as the most demanding, while the other ones are comparable. For each task, the raw-TLX is consistently below the weighted value, so the dimensions that received higher values had less impact on the overall workload perception.

The post-test questionnaire highlights a good overall usability level. ProtoSketchAR obtained an average SUS [B\*96] score of 81.5 ( $s = 12.65$ ), positioning into the 87th percentile of the score distribution curve. In the custom Likert-scale questions (1 to 7), we assessed the effectiveness of the sketching modality ( $\bar{x} = 6.6$ ,  $s = .25$ ), the feedback provided ( $\bar{x} = 6.1$ ,  $s = .3$ ), the support for design iterations ( $\bar{x} = 6.2$ ,  $s = .23$ ), the effectiveness in prototyping AR applications ( $\bar{x} = 6.1$ ,  $s = .21$ ) and the ability to turn into a sketch the designer's ideas ( $\bar{x} = 6.3$ ,  $s = .19$ ). All the scales received high rankings, showing again a good acceptance of the prototyping tool. During the debriefing sessions, we also collected interesting

qualitative feedback that is important for the further development of the tool. First of all, the consistent higher time spent and workload in Task 1 highlights that the tool supports more easily the prototyping of virtual objects rather than UI or text elements. This suggests that we must consider including specific tools in ProtoSketchAR for writing text or including UI widgets and panels. The participants used both sketching modes during the tasks. The *classic* mode was preferred when they needed to position the sketch on a flat surface, and it supported faster iterations. The participants liked the *pinned* version for mid-air objects, taking advantage of the coordination between the movements and the screen orientation. They commented positively on the direct feedback on the 3D environment while drawing in the pinned mode. The screen size impacts the participant's strategy for creating the sketch: tablet users consistently maintained the entire sketch in their canvas, while smartphone users planned an iterative process splitting the creation of the sketch into parts. Such behaviour requires further investigation for identifying the size that triggers it and if specific interface elements or interactions are required for its support. Some participants asked for more paint options (e.g., colour-filling tools like the paint bucket). They are, of course, possible to implement but not included in the current version. Finally, it is not completely clear whether ProtoSketchAR finished scanning the environment or not from the current interface. Many participants started sketching too early, and this caused some problems in positioning them. The feedback on this must be improved.

In conclusion, such preliminary evaluation highlighted already an overall good usability and the effectiveness of the sketching approach, especially in modelling scenes including 3D objects. The two sketching modes seem to have specific use cases, so it may be useful to keep them both. However, the comments on the pinned mode feedback may suggest a poor design for the one provided by the classic mode. In contrast, the difference in efficiency suggests that we should simplify the pinned mode.

## 5. Conclusion and future work

In this paper, we present ProtoSketchAR, a web-based application providing a rapid prototyping tool in Augmented Reality through sketching techniques. The developed app can be classified as a low-fidelity prototyping tool, as the creation of sketches to place in the environment does not require high fidelity, but focuses more on the creation of rapid AR prototypes. ProtoSketchAR provides various well-implemented functionalities that compose an application ready to be improved and extended by new functionalities, allowing support and compatibility with other devices (AR/MR viewers) and allowing an even more immersive experience for the user. The evaluation shows already a good usability of the sketching interface. However, it also highlighted the need for in feedback for the classic mode and on the efficiency for the pinned mode. We plan to address this issue, together with other improvements in interface and functionalities, and validate the usability again with a new user test involving a larger audience, including people with a high familiarity with AR applications.

## References

- [afr] A-frame - make webvr. <https://aframe.io>. Accessed: 2021-10-14. 3
- [AHKG\*18] ARORA R., HABIB KAZI R., GROSSMAN T., FITZMAURICE G., SINGH K.: Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), CHI '18, Association for Computing Machinery, p. 1–15. doi:10.1145/3173574.3173759. 2
- [B\*96] BROOKE J., ET AL.: Sus-a quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7. 4, 5
- [dSvDV09] DOS SANTOS G. J. D., VAN DIJK E. M. A. G., VYAS D. M.: Paper-based mixed reality sketch augmentation as a conceptual design support tool. In *Proceedings of the 2009 British Computer Society Conference on Human-Computer Interaction, BCS-HCI 2009, Cambridge, United Kingdom, 1-5 September 2009* (2009), ACM, pp. 447–452. doi:10.14236/ewic/HCI2009.56. 2
- [GJSW19] GASQUES D., JOHNSON J. G., SHARKEY T., WEIBEL N.: What you sketch is what you get: Quick and easy augmented reality prototyping with pintar. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), CHI EA '19, Association for Computing Machinery, p. 1–6. doi:10.1145/3290607.3312847. 2
- [HS88] HART S. G., STAVELAND L. E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Human Mental Workload*, Hancock P. A., Meshkati N., (Eds.), vol. 52 of *Advances in Psychology*. North-Holland, 1988, pp. 139–183. doi:https://doi.org/10.1016/S0166-4115(08)62386-9. 4, 5
- [ike] Ikea place app. <https://www.ikea.com/au/en/customer-service/mobile-apps/say-hej-to-ikea-place-pub1f8af050>. Accessed: 2021-09-15. 4
- [McC] MCCURLEY V.: Storyboarding in virtual reality. <https://virtualrealitypop.com/storyboarding-in-virtual-reality-67d3438a2fb1>. Accessed: 2021-09-15. 2
- [NLC\*20] NEBELING M., LEWIS K., CHANG Y.-C., ZHU L., CHUNG M., WANG P., NEBELING J.: *XRDirector: A Role-Based Collaborative Immersive Authoring System*. Association for Computing Machinery, 2020, p. 1–12. doi:10.1145/3313831.3376637. 2
- [NM19] NEBELING M., MADIER K.: 360proto: Making interactive virtual reality & augmented reality prototypes from paper. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), CHI '19, Association for Computing Machinery, p. 1–13. doi:10.1145/3290605.3300826. 2
- [NNYR18] NEBELING M., NEBELING J., YU A., RUMBLE R.: Protoar: Rapid physical-digital prototyping of mobile augmented reality applications. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), CHI '18, Association for Computing Machinery, p. 1–12. doi:10.1145/3173574.3173927. 2
- [PTB\*01] POUPYREV I., TAN D. S., BILLINGHURST M., KATO H., REGENBRECHT H., TETSUTANI N.: Tiles: A mixed reality authoring interface. In *Human-Computer Interaction INTERACT '01: IFIP TC13 International Conference on Human-Computer Interaction, Tokyo, Japan, July 9-13, 2001* (2001), Hirose M., (Ed.), IOS Press, pp. 334–341. 2
- [RvHPR18] RAZEK A. R. A., VAN HUSEN C., PALLOT M., RICHIR S.: A comparative study on conventional versus immersive service prototyping (vr, ar, mr). VRIC '18, Association for Computing Machinery. doi:10.1145/3234253.3234296. 1
- [XSS08] XIN M., SHARLIN E., SOUSA M. C.: Napkin sketch: Hand-held mixed reality 3d sketching. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology* (2008), VRST '08, Association for Computing Machinery, p. 223–226. doi:10.1145/1450579.1450627. 2