AUTHORING TOOLS FOR AUGMENTED REALITY SCENARIO BASED TRAINING EXPERIENCES

by

ANDRÉS N. VARGAS GONZÁLEZ
B.S. Escuela Superior Politécnica del Litoral, 2010
M.S. Computer Science, University of Central Florida, 2014

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2022

Major Professor: Joseph J. LaViola Jr.

ABSTRACT

Augmented Reality's (AR) scope and capabilities have grown considerably in the last few years. AR applications can be run across devices such as phones, wearables, and head-mounted displays (HMDs). The increasing research and commercial efforts in HMDs capabilities allow end users to map a 3D environment and interact with virtual objects that can respond to the physical aspects of the scene. Within this context, AR is an ideal format for in-situ training scenarios. However, building such AR scenarios requires proficiency in game engine development environments and programming expertise. These difficulties can make it challenging for domain experts to create training content in AR.

To combat this problem, this thesis presents strategies and guidelines for building authoring tools to generate scenario-based training experiences in AR. The authoring tools were built leveraging concepts from the 3D user interfaces and interaction techniques literature. We found from early research in the field and our experimentation that scenario and object behavior authoring are substantial aspects needed to create a training experience by an author. This work also presents a technique to author object component behaviors with high usability scores, followed by an analysis of the different aspects of authoring object component behaviors across AR, VR, and Desktop. User studies were run to evaluate authoring strategies, and the results provide insights into future directions for building AR/VR immersive authoring tools. Finally, we discuss how this knowledge can influence the development, guidelines, and strategies in the direction of a more compelling set of tools to author augmented reality SBT experiences.

A mi mami María Dolores López Villacrés

ACKNOWLEDGMENTS

There are many people that have contributed directly or indirectly to this dissertation. I will do my best to not forget anyone.

I would like to thank my faculty advisor Dr. Joseph LaViola, his support and academic guidance have been invaluable to me during this journey. Thanks to the members of my committee: Dr. Charles Hughes, Dr. Gregory Welch, Dr. Joseph Kider and Dr. Mark Billinghurst for their guidance and helpful feedback to improve my work. I am particularly grateful to Dr. Billinghurst for his earlier collaboration with this work.

Special thanks to the past and present members of the Interactive Computing Experiences Research Cluster, for their support, motivation and help, especially Brian Williamson, Dr. Seng Lee Koh, Dr. Kate Kapalo, Dr. Eugene Taranta, Dr. Sina Masnadi, Dr. Corey Pittman, Dr. Kevin Pfeil and Dr. Ravikiran Kattoju. So many great discussions, and anecdotes. I learned so much from each of you. I would like to extend my thanks to Brian Williamson, for all his collaboration, help and great conversations about so many random topics. I would like to especially thank Dr. Koh for his friendship and help.

I am very grateful to Dr. Lori Walters and Dr. Robert Michlowitz for all the help to capture 3D scans of the lab and objects from the environment. I would also like to thank Dr. Robert Sottilare for his contributions to this work.

Infinite thanks to my grandmother who as a kid told me that the best inheritance she could give me was education. Indeed she is always right. To my dear mom Beatriz and aunts Alejandrina and Paulina in Ecuador, thanks as well for the motivation in your own unique way. I am grateful for taking care of me when I was a kid. Thanks to my dear wife Claudi to keep me going and for being there days and nights. Thanks to Leon my son for all the smiles and extra motivation. Also, I am truly grateful to Ilona Heinze for giving us a hand at the very end when I needed time to work.

I want to thank Dan Aton from 8KSolutions for his support, trust and understanding to help me balance work and university workload. Thanks to Tim Kanellitsas and Andy Jevsevar for the time at KairosXR, I learnt so much from you. Thanks to Eugene Rodriguez and his wife for the cafecito and warm talks.

I would like to conclude by thanking the following people: Rachel Shann, Liuliu Wu, Denisse Cayetano, Walter Wehner, Ankit Sharma, Behnaz Nojavanasghari, Syed Alam Abbas, Mykola Maslych, Jose-Valentin Sera-Josef, Jacob Belga, Ryan Ghamandi and all the great people I met during my time at UCF, Gustavo Cali, Dr. Carmen Vaca, Dr. Katherine Chiluiza, Dr. Enrique Pelaez, Dr. Xavier Ochoa and all my friends in ESPOL for the good times when back home.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

For more than three decades, instructional design (ID) has leveraged Virtual Reality (VR) environments for training perception and action tasks [154]. But, VR environments are, after all, not real; real-world cues, egocentric relationships, and tangible interactivity in a training environment are crucial for enhancing efficacy as users practice tasks as they would in a typical training scenario [6, 50, 61].

For instance, training a novice pilot in a cockpit might require pushing buttons, touching screens or maneuvering with the flight stick, etc., capabilities that are difficult to simulate in VR. On the other hand, augmented reality (AR) can fill this gap with regard to the capability of providing, reinforcing, and manipulating tangible cues and actual controls inside a real cockpit. In alternative examples, nursing students can practice a variety of medical procedures on the same mannequin using AR projection technology [122], or technicians can practice maintenance tasks through AR head-worn displays (HWDs) with the help of a subject-matter expert who could be remotely located [39], or sports players can practice game concepts as in [147].

From a learning perspective, intelligent tutoring systems provide a more tailored learning experience than traditional classrooms or e-learning. In the motherboard assembly domain, Westerfield et al. in [151] show that an intelligent approach improves test scores by 25% and task performance by 30% in comparison to a non-adaptive strategy. Design considerations for combining AR with ITS systems are provided in [48]. The complexity of an ITS is a significant limitation on a possible tool that can combine both approaches.

This PhD thesis explores strategies for scenario and objects' behavior authoring to build AR training experiences. There is a wide variety of ways that AR can be used to enhance training, and these can be classified according to their spatial mapping and physical space requirements. Figure 1.1

depicts examples of AR applications classified along possible spatial mapping requirements in the x-axis and physical space needed in the y-axis. Vital sign monitoring in [83, 124] are examples of AR applications not requiring any spatial mapping and small physical space as content is displayed in front of the person's view based on information analysed and processed from wearable sensors. SoccerTutor by Vargas et al. in [147] shows an AR training application requiring a big open field to render graphics and a low understanding of the environment as AR content is over-imposed on the field. Processar [20], and Motherboard Assembly [151] require higher levels of spatial mapping with a physical interaction space defined by the bounds of the object. This dissertation focus on scenarios that fit inside a room, e.g. kitchen, office, etc. Scalar [116] is an example of augmented content associated with elements inside a room-size environment (see shaded region in figure below).

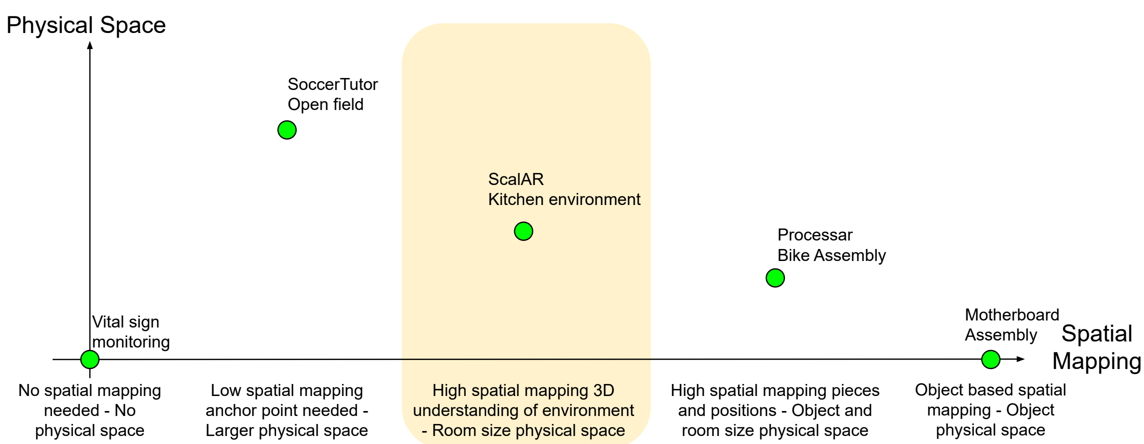


Figure 1.1: Space continuum, light shade highlights area of incidence for this work.

To provide a better user experience for the use cases just given, some examples of research on input devices that provide unique means of interaction are, e.g. smartphones [62] taking advantage of the touch capability, gestures for instance, to make annotations [18, 108], using the affordances of the environment [47] to give input, sketch to generate AR content [41] and even custom interaction devices such as in [52].

One of the important elements for AR Training are authoring tools that make it easy to create training simulations. To build a compelling experience, it is essential to consider an asset's fidelity to be as close as possible to the interaction in the real world, and information visualization needs to be conveniently placed on the space [87] or screen. These concerns inspire us to work towards creating better immersive interfaces for authoring scenarios and the objects' behaviors associated with the training experience. Therefore, we base our work on the concepts defined for scenario authoring and objects' behaviors from robotics and virtual environments to create authoring of such concepts in immersive AR/VR environments. Even when other related issues could be discussed, we concentrate on proposing immersive 3D UI for authoring. These interfaces are built upon design principles that allow us to compare our work to the current well-established desktop interfaces.

This dissertation concentrates on the following topics:

1. Authoring scenario-based training simulations within a room-size context
2. Authoring objects' behaviors and their components in real and virtual objects

These topics are essential because training scenarios efficiently convey a specific message or idea to a learner or person of interest. For instance, in [14] Buttussi et al. found that increased knowledge and self-efficacy are retained for safety training procedures regardless of the display fidelity. AR provides the capability to enhance a particular location with over-imposed graphics while at the same time taking advantage of the cues provided by the environment as can be seen in [147]. To achieve these outcomes, tools to author scenarios and objects' behaviors are required to be easier to use for domain experts.

1.1 Motivation

The work presented in this document is motivated by the constructivist principles of situated learning [60]. The initial work in [147], served as an exploration of an in-situ learning experience in AR (see Figure 1.2). It showed the learning benefits presented in an immersive environment instead of a traditional Desktop interface. Extending this concept to situated authoring, content creators can benefit from the affordances given by the real physical environment to create more compelling content, as demonstrated by Lane et al. [17]. The fundamental authoring aspects of such training experience involve scenarios and objects' behaviors.

Figure 1.2: From [147], the figure shows a scenario for an offside rule with players as virtual objects following behaviors conveying a concept.

Scenario authoring gives content creators the ability to create training experiences. Within the context of a room-size physical space defined in Figure 1.1, many virtual and real objects can be placed in the environment and interacted with by a trainee in this case. Defining the flow of interaction is a challenging problem that we model leveraging the work by Achour in [1]. We focus on "atomic actions" which model the interactions between a virtual actor and a virtual object. The Desktop tool by Norton et al. [107] to author scenario-based learning experiences is also a motivation to explore the concepts they propose in an in-situ environment.

Finally, these actions/interactions between a virtual actor and a virtual object trigger in the object a specific behavior, motivated by the work by Kallmann in [64]. They categorize object property as a class involving descriptions, parts and actions in this dissertation's scope. Specifically for parts, we also look at the work in robotics for affordance templates to define object components' behaviors to robots [43, 56]. Motivated by this work, we implement these concepts in our AR/VR authoring tools. To fulfill the promise of situated authoring affording a real benefit, we motivate our work on improving the usability of immersive approaches by comparing choices made in such interfaces against non-situated authoring choices.

1.2 Research Statement

This dissertation leverages all the above considerations and presents studies that tackle different aspects of authoring AR scenario-based training simulations. Based on our motivation to analyze and improve the usability of an immersive authoring tool for scenario and objects' behaviors authoring, AR/VR systems were implemented taking into consideration best practices from building 3D User Interfaces [74, 114], guidelines from designing mixed reality applications [96], and traditional user interfaces [2, 131] for desktop. First, in chapter 3, through a user study, we evaluate a holistic approach for authoring scenario-based training simulations grounded on the concepts of scenario authoring and exploring different interaction techniques. This holistic approach is built in AR and Desktop and introduces extrinsic characteristics or descriptions of objects as information that can augment virtual objects. We learned that compared to the Desktop, participants found it harder to keep track of the authored progress on the list of tasks within the immersive environment. This is due to the scene's visual crowding of different virtual elements. Other recommendations are given to reduce workload and improve usability when authoring a scenario in AR. Then, to understand the significance of virtual and real objects for authoring training simulations, an authoring technique is proposed to add or enable geometric constraints intrinsic to objects authored. The methodology is presented and evaluated through an exploratory study in chapter 4. The method proposed was well received by participants, with steps easy to remember and reproduce. Chapter 5 evaluates this authoring technique across AR, VR and Desktop and sheds light on users' preferences and challenges in each condition with recommendations to improve usability in immersive approaches. Chapter 6 concludes this dissertation.

Recent commercial efforts like Snap Lens Studio or Zapar provide non-immersive easy-to-use interfaces for rapid prototyping of AR/VR experiences [8, 156]. These tools were created for different purposes and could be adapted to be used for building AR training experiences, but, with some higher level of complexity. For other tools like [3, 63, 97], the learning curve for non-programmers can be high, which becomes a barrier for domain experts whose knowledge can be used for generating a training experience. Using immersive interfaces to build AR experiences and more suitable ways to generate content is not as well-defined as in a 2D graphical user interface. 3D user interfaces and interaction techniques have been extensively explored [74] in the context of VR, and less extent in AR [4]; however, given the current state of the art devices, we seek to provide guidelines and recommendations for building authoring tools.

Authoring tools for AR experiences are not new and have been explored in different contexts, such as game-level authoring [6, 104] and immersive tangible interfaces [6, 77, 81, 115]. Authoring applications for AR have been developed using different interaction devices ranging from desktop [84, 85, 127], mobile [81, 121, 150] and immersive AR [35, 81, 104]. Advantages of situated authoring have been found in [17] by placing authors in the learners' environment so better pedagogical content can be generated. Due to this, it is essential to evaluate the usability of immersive authoring tools to ease the content generation process. This dissertation seeks to define a set of design guidelines, user interfaces and interaction techniques to facilitate the creation of AR scenario-based training experiences, allowing non-technical users to create AR SBT scenarios. Studierstube by Schmalstieg [125] is probably one of the first frameworks that explored AR in the living room and collaborative space. Avantguarde by Sandor [123] is the first work to build authoring tools for ubiquitous AR. Furthermore, Güven in [38] presents the concept of situated media and hypermedia together with authoring techniques for such concepts in desktop and tablets interfaces.

1.2.1 Thesis Statement

By providing 3D spatial user interfaces in the context of an Augmented Reality scenario-based training experience, users situated in an immersive environment can author virtual/real objects' behaviors with outcomes comparable to a Desktop authoring tool.

1.3 Contributions

The following are the contributions of this dissertation:

- An evaluation of a visual programming interface that follows scenario authoring guidelines from Achour in [1] under two different interfaces AR and the Desktop. The work by Norton et al. [107] serves as a reference, with the addition of the graph-based authoring paradigm and a 3D room-size scenario.
- A simple methodology to enable affordances in object components, adapting instructions

provided to robots to understand how to interact with real objects in a real environment. The approach, named AffordIt!, involved the definition of a region of interest, the segmentation of the region as a component and an affordance template definition to constraint the component movement or rotation to specific values.

- Extending to the AR/VR domain the work by Kallmann in [64] for object interaction in Real-Time Virtual Environments with a specific focus on object components/parts. The AffordIt! methodology is adopted and an evaluation across AR, VR and Desktop is performed to analyze areas for improvements for immersive interfaces.
- A list of guidelines and strategies to improve the usability of immersive authoring tools. From experimentation and gathering of data in the form of interviews and questionnaires. We condensed elements from the interfaces and decisions that could be improved in future approaches while building immersive authoring tools to author scenario-based training simulations.
- Demo applications developed for the different topics explored in this dissertation.

1.4 Thesis Outline

Chapter Two will discuss related work to authoring tools in general, AR content generation tools and Object Behaviors. Chapter 3 presents a study comparing custom-built desktop and AR-based authoring tools. This study examines the usability and efficiency of participants while authoring an SBT experience. Chapter 4 shows AffordIt!, a technique to author intrinsic virtual object behaviors. Finally, Chapter 5 discusses an evaluation of AffordIt! across different conditions, AR, VR and Desktop. Chapter 6 ends the dissertation with a discussion and conclusion.

CHAPTER 2: LITERATURE REVIEW

This dissertation is grounded on research in the following areas: Scenario-based training, authoring tools, AR content generation tools, Geometric Content Creation & Manipulation and real and virtual objects behaviors.

2.1 Scenario Based Training (SBT)

SBT also referred to as Scenario-Based Learning (SBL), is grounded in the constructivist principles of Situated Learning [60]. This is the idea that transferable knowledge is optimally acquired and understood when it takes place *within* the context and domain of its application (i.e. under situated cognition [13]). Augmented Reality represents an ideal environment for SBT; learners are exposed to *near-world* simulations [112] of real-world situations, blending in virtual objects that simulate sensory inputs to enhance spatial cognition and experience of the physical environment [19,51,120]. Thus, *situated authoring* could also benefit from similar affordances of AR [17], extending this theory. Lane et al. found that by placing authors in the learners' environment that is similar to the actual environment [17], novice authors were able to model pedagogically effective content. Based on the studies of situational [17] and scenario authoring [1], a graphical approach that additionally serves as visual feedback is expanded further in Chapter 3.

Scenario-based training approaches have improved communication in the context of disaster management as is shown in the work by Haferkamp et al. [40] with a 2D computer-based simulator. SBTs can increase in complexity based on scenario preparation as in the study by Paige et al. in [110] with very realistic setups. They showed that high fidelity positively impacts self-efficacy for effective teamwork performance. On the other hand, Dahlstrom et al. in [22] argue that lower-fidelity simulations can still provide competence development and significant results when the scenario is appropriately designed. In our case, AR brings the advantage of using real-world cues coupled with high-fidelity assets as virtual elements. An analysis of lower-fidelity assets is out of the scope of this work but an interesting topic for future work exploration.

2.2 Authoring Tools

2.2.1 Augmented Reality

Nebeling et al. [102] categorize AR-based authoring tools on the spectrum of level of fidelity vs skills and resources required. Our work aims for high fidelity and low skills and resources required.

In some cases, AR authoring approaches rely on tracking fiducials to render a virtual object in the viewer's perspective. Tiles is an early effort by Kato et al. [115] to provide a collaborative scenario where the combination of virtual objects is used to interact with the scene to invoke an outcome. Similarly, in [81] Lee et al. introduce the concept of immersive authoring. Virtual objects in a scene are modified by different UI components associated with tangible fiducial markers. Generally, it was found that the immersive strategy was preferred over non-immersive approaches. In line with this work, Rajaram et al. in [118] explore using a regular sheet of paper to enable instructors to create AR educational experiences. The type of assets used is descriptive in nature, either to augment information given on the paper or to visualize a concept. Interestingly, participants found more practical authoring on a handheld AR device and visualizing it on a head-worn display as the more natural experience.

In-situ authoring tools are situated in the context of procedural instructions. In the work by Gonzalez et al. leveraged by this thesis [35] a visual authoring tool is presented to add behaviors and interactions to virtual objects in a scene. The behaviors authored in this tool are descriptive (text or audio). In the work by Ng et al. in [104] a set of guidelines is presented for procedurally authored games in AR. The type of non-character behaviors used is non-deterministic to the object's nature rather augmentative for instance, changing visibility using a bounding box as a trigger mechanism. In the industrial context, AR is used to create procedural training experiences by demonstration as in [20, 75]. Both approaches were constrained to recording video, audio and movement of virtual assets. While well received and preferred over the PC they did not explore behaviors related to the geometric properties of the objects. In the same context, the work by Izquierdo-Domenech et al. [54] and Chidambaram et al. [20] use Computer Vision to classify objects from the environment. Izquierdo-Domenech et al. [54] is a more automatic approach to add semantic information to objects in the scene. In this dissertation we focus on the evaluation of a human-in-the-loop approach where geometrical constraints are defined by the subject matter expert using an authoring tool, as seen in AffordIt!, [93].

Virtual objects are key components when authoring instructions in AR. Jasche et al. in [59] compared two types of AR object visualizations Concrete (CAR) and Abstract (AAR). CAR involves using complete CAD-style meshes, while AAR relates to using wire meshes and 3D arrows. The result shows that concrete visualizations induce fewer errors from participants, especially with complex tasks and have a clear advantage over AAR visualizations. Both approaches are improved when coupled with videos. Our work adopts the concrete visualization for the demonstration of behaviors authored. Intrinsic object properties like stiffness and motion resistance are captured in RealityBrush [72], a novel authoring system that creates virtual replicas of real objects with measured kinetic properties. This work enables a virtual asset to afford two types of actions, poke and push, depending on the force applied. In a similar context in GesturAR [148], everyday objects are scanned, and behaviors are authored by mapping a freehand interaction to an action. Similar to our work, a hinge-joint type of interaction is proposed. However, object component segmentation is not explored.

2.2.2 Desktop

Early research in desktop-based authoring tools focused on localizing and mapping fiducial tags to 3D graphic content [70]. We adhere to Kato et al. [71] design principle of object affordances matching physical constraints of the object. This work has mainly focused on user interactions needed when defining AR tracking fiducials [84, 85, 127], such as for attaching actions and behaviors to virtual content. MacIntyre et al. [85] presented many novel features for exploring AR content inside a MacroMedia environment [1], but only while off-line in a desktop setting. The work of Spini et al. proposes an authoring web tool for asset placement and visualization of quasi-photorealistic scenes in VR [136]. Web 3D is close to our work on the Desktop end but our application explores this further, adding behaviors to elements placed in the scene as a sequence of actions generated by the user. Game engines such as Unity3D, Unreal or Amazon Sumerian are common desktop tools developers use to create AR SBTs. To ease AR training scenarios development, commercial companies like NGrain with Producer Pro [63], ScopeAR with WorkLink [3], or Microsoft with Dynamics 365 Guides [97] offer a Desktop application or Unity plugin that allows creators with little or no coding knowledge to build training experiences to be deployed on AR powered devices. Our Desktop interfaces follow design principles based on scenario authoring guidelines specific to the scenario evaluated and different from the assembly training context of the commercial tools. Assembly training might need a more detailed mapping of the object space (different to the room space) in which interactions happen (see Figure 1.1).

In the educational context, Zhang et al. in [160] propose an authoring tool for experimental education, a traditional graphical user interface to enable educators to author AR attributes of virtual experimental equipment. Defined behaviors can be picked from a database and attached to the virtual objects represented by an AR marker at runtime. Another alternative toolkit is ARSpot [117]

[1] `https://en.wikipedia.org/wiki/Macromedia`

which is built on top of Scratch [88, 119] a popular toolkit for kids to learn programming concepts and visualize their code within an integrated development environment. The solutions discussed utilize fiducials to represent virtual objects in the scene, which we adopted for our AR condition.

2.2.3 Virtual Reality

Virtual Reality provides an ideal environment to simulate real-world conditions when creating identical replicas of the environment. Ipsita et al. in [53] propose VRFromX a VR authoring tool that allows users to create interactive virtual experiences from a 3D point cloud scan. A region of interest (ROI) is first extracted and then queried to a database of virtual objects. Then affordances can be enabled in the resulting object. They test the usability of the system in a welding training scenario. In line with this work, Masnadi et al. in AffordIt! [93] focus on behaviors authored in objects' components. Similarly, an ROI is defined as identifying a component of an object to then enable geometric affordances on it. A preliminary evaluation showed high usability on the techniques proposed.

ScalAR in [116] provide a holistic approach to scene authoring by tackling the problem of authoring a scenario independently from a room layout configuration. Content information from a scene is synthesized using a semantic understanding technique. Virtual replicas of the identified objects are used in a VR environment to author semantic level associations of AR content in each scene. The behaviors authored in this approach are descriptive or augmentative rather than specific to the objects intrinsic properties. In the context of Internet of Things (IoT) Ivy by Ens et al. [25] use VR for authoring intelligent environments. A node-link visual programming interface allows users to author IoT programs and visualize sensor data. Ivy's object space is different than objects with geometrical components explored in this work.

Designing an authoring tool for virtual environments using constructivist principles is explored by

Winterbottom & Blake in [153]. Constructivism is grounded on the theory that knowledge is built by the interaction of a user with the environment. Our scope is based on actor-object interaction rather than actor-avatar. Authoring at run time was first explored by Steed et al. [137] using data flow diagrams while immersed to define a virtual object behavior. Their authoring was constrained to interactions with input devices. Inspired by this work, our approach allows for in-situ tools to visualize the outcome of the authored behavior.

For virtual object alignment and manipulation we look to Hayatpur et al. [46] which presents three techniques invoking either a plan, ray or point and using hand gestures to apply movement constraints. Virtual Reality is also used as a means to enhance the authoring experience when combined with a Desktop approach [49]. Holm et al. explore the advantages of a combined approach, but our work evaluates the interfaces independently. However, a similar study combining both interfaces could be performed in the future.

2.3 AR Content Generation Tools

Augmented Reality content generation tools are classified as standalone and AR-plugins which can be distributed, platform-specific, or platform independent [99]. Our work is categorized as standalone and distributed as platform-specific for AR and platform-independent for desktops (web interface). For content authoring, sketch is one of the mediums utilized to build AR scenes. For example, Sketchaser uses a visual language to generate virtual content from hand sketches [41]. Multi-touch interaction has also been used to apply transformations to virtual objects in the real world [62]. Other content generation tools have focused on extracting 3D models from cartoon drawings and allowing users to interact with them through a multi-touch interface [27]. Other research used familiar controls like smartphones coupled with AR HWDs [155]. The phone works as an input tool to select, place and manipulate virtual objects in the user's physical space, helping them use a familiar control to generate content. While prior research focuses more on asset creation and object placement, our work explores higher level scene generation with added behaviors under different interface conditions.

In the context of augmenting content on real world objects, research shows for instance that maintenance tasks [26, 47, 161] have increased performance of users by taking advantage of the affordances from the domain environment. Context is an aspect that the authors mention is important since it defines the interaction based on what is being visualized. It is achieved by using image recognition techniques [26] or simply by placing a QR code in the field of view [161].

AR training applications are becoming more relevant and authoring commercial tools are available from different companies. These tools enable domain experts to build AR training scenarios often using a traditional user interface. In line with work done by Lee et al. [81], an up to date comparison of equivalent systems could bring insights into further work required to improve immersive authoring of AR based training experiences. An easy to understand conceptual model is introduced in the following chapter. The model follows the simple case of using atomic actions (an action happens from an agent to another agent). This model can grow in complexity if the whole grammar is implemented [1].

2.4 Geometric Content Creation and Manipulation

Deering presented HoloSketch a novel VR sketching tool at the time to create geometric content, manipulate it and add simple animations [24]. Our work is different from HoloSketch in the interaction techniques, mesh segmentation and use context. However, different features from HoloSketch can be adapted to AffordIt!. For mesh manipulation we have found Sketch based applications to be the predominant research in this domain. SKETCH by Zeleznik et al. [159] is an early example of creating 3D objects from 2D sketches. In SKETCH constrained transformations are applied to objects, a concept that we utilize in AffordIt!. In Shao et al. [130] a sketch based application is presented that applies behaviors to concept sketches based on a region of interest selection followed an animation added to an individual part. This is similar to our approach, except that their interface is entirely 2D interactions upon a 2D object while AffordIt! explores 3D interactions and seamless visualizations with a 3D object. Commercial companies have also begun to provide a variety of tools [16, 133] that easily create 3D geometric content. AffordIt! is complimentary to these tools by providing an extension of capabilities in applying intrinsic behavior to an object.

Our interaction techniques derive from the research in object authoring by Hayatpur et al. [46] which presents three techniques for object alignment and manipulation in VR. These techniques invoke a plane, ray, or point and use hand gestures to apply movement constraints to a virtual object. Their research presents a rich set of interaction possibilities, however the idea of changing an object geometry to tie behaviors to its component parts is not studied. We address this by proposing two techniques to generate intrinsic object behaviors at run-time. First, a user is allowed to define each object component behavior from the interaction in a VR environment. Second, we apply authoring behaviors similar to [90, 91] except that we transition from a 2D sketch based interface to a 3D interaction paradigm.

Authoring constraints has been explored in the context of objects associations based on geometries as in [134, 145]. For instance, a book if placed on the top of a desk is associated with the table with one face facing the desk. In the work by Oh et al. [109], authoring objects are constrained to movements in a plane, when a collision is detected. While this is similar to our movement constraint behaviors, it is a Desktop based solution rather than authoring from within the VR environment. The theory of affordances by Gibson is divided into the concepts of attached and detached object affordances [32, 33]. Attached objects cannot be removed from their parent object unless they become a detached one and usually have constraints in their movements. While there is successful work in robotics to apply affordance theory to provide guidelines for object manipulation [43, 57, 89], the application on 3D objects authoring is limited.

2.5 Real and Virtual Objects' Behaviors

Text, audio, videos, animations, color and visibility are examples of some characteristics that can be authored and associated to virtual objects in a scene with specific examples given in the literature review above. A general model for interactions between virtual human agents and objects is proposed by Kallmann et al. in [67]. This work includes the definition of the virtual object space according to four different classes of interaction-features: intrinsic object properties, interaction information, object behaviors and expected agent behaviors. A graphical user interface and examples are described in [66]. Furthermore, in Kallmann et al. in [68] the object properties are divided in descriptions, parts and actions. A taxonomy for interactive object behaviors is proposed and demonstrated with a GUI to visually author behaviors as graphical state machines. Yet again, an extension of this model in [65] defines more complex behaviors such as multiple actors interacting with one object and the the actor manipulation of virtual objects introducing the term ”smart objects”. The extent of the work by Kallman et el. is demonstrated in a non-immersive environment, our work does evaluate immersive approaches and adopts the parts and actions intrinsic properties for our scope.

Geometric properties are discussed in the work by Thalmann [141], the problem being how to model virtual objects interaction with others. They explore as well how to adapt the grasping definitions from robotics in a VR simulation. An additional property is added to the object model to define how an object can be grasped. Lee et al. in [80] introduce the concept of programming by demonstration. An object behavior is composed of events, context and action components which they call ACE. This behavior is encoded as a Backus-Naur Form (BNF) notation that is then attached to a virtual object to specify interactivity in the virtual world. Authoring is performed in a desktop interface and visualized in VR whereas the interface we used allows a user to author and visualize behaviors from within the VR environment. An additional conceptual model called VR-WISE is presented in Pellens et al. [111] which establishes simple behaviors that when nested could produce more complex ones. Simple behaviors are categorized as primitives and are defined as move, turn and roll in their taxonomy, but is not evaluated in a user study.

Smart Objects in the context of Internet of Things are real physical artifacts, enhanced with sensors and connected in a network that allows communication with humans and other artifacts as a part of the Internet of Things (IoT) paradigm by McEwen and Cassimally [95]. From an HCI perspective humans interacting with such objects face a usability challenge. Work by Matassa et al. [94] emphasize the problem of smart objects being unable to immediately communicate to people what they can afford to do. Baber et al in [5] proposes a conceptual framework to exploit the affordance concept through an understanding of how humans engage with objects. The forms of engagement proposed are environmental, perceptual, morphological, motor, cultural and cognitive.

2.6 Summary

The authoring of scenarios and objects behaviors from an in-situ approach and how it compares to a traditional desktop experience is important to address in order to take advantage of the better pedagogical outcomes situated authoring offers [17]. We, therefore, leverage different concepts and ideas to develop a holistic authoring tool in AR and Desktop for building SBT simulations and then AffordIt! to analyze authoring object component behaviors. We evaluate its effectiveness across AR, VR and Desktop. Different guidelines and recommendations are presented to better help system engineers build authoring tools for AR based training simulations.

CHAPTER 3: COMPARING DESKTOP AND AR AUTHORING TOOLS A HOLISTIC APPROACH

This chapter is based on work published in: Gonzalez, A. V., Koh, S., Kapalo, K., Sottilare, R., Garrity, P., Billinghurst, M., & LaViola, J. (2019, October). A comparison of desktop and augmented reality scenario based training authoring tools. In 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) (pp. 339-350). IEEE.

3.1 Introduction

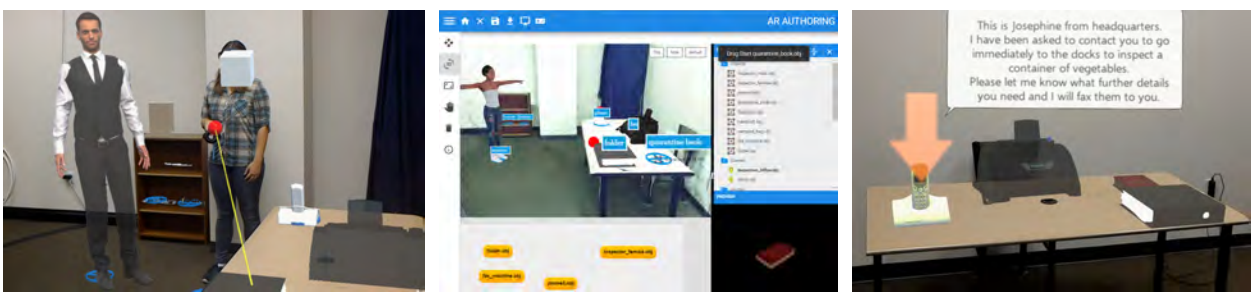


Figure 3.1: On the left, the AR authoring environment. The center figure presents the GUI of our desktop authoring tool. On the right, the playback application running an authored scenario.

In this chapter, an AR authoring interface (see Figure 3.1 left) is built to allow an author to create a virtual object and position it in the environment. Also, to define actions following a visual authoring approach. Similar to the work by Ens et al. [25] and Ng et al. [104] we use floating 3D panels and visual authoring elements to convey an authoring workflow for a scene. The Desktop interface is built following traditional 3D editing tools approaches (see Figure 3.1 center).

Technological advancements have allowed learners of all ages to interact with devices such as laptops, mobile phones, and even Virtual Reality (VR). Three-dimensional learning has been shown to provide better gains than traditional observation. For instance, in work by James et al. [55] participants actively interacting with a 3D model could better retain object shape structure and recognize faster the artifact on a recognition task. Similarly, anatomy learning can be improved by allowing students to directly manipulate virtual anatomical structures [58], leading to successfully generated observed structures on a post-test. As this type of learning becomes more prevalent, users are using commercial off-the-shelf Augmented Reality (AR) and VR technology, such as the HoloLens and HTC Vive, to create content. These technologies make it possible for learners to immerse themselves in training environments that might otherwise incur expensive costs or require significant time commitments and resources. For example, nursing students can practice a variety of medical procedures on the same mannequin using AR projection technology [122], or mechanics can practice maintenance tasks with the help of a remote expert and AR head worn displays (HWDs) [39].

In light of this, content generation for learning [157, 158] becomes more important than ever since designers must consider not only user needs, but also the platforms on which learners consume content. Sometimes these platforms can be implemented directly in the learning environment [28, 147], enhancing efficacy as users practice tasks as they would in a typical training scenario. For this reason, AR provides an ideal format for scenario-based training (SBT) since it involves using real-world cues and spatial relationships based upon the user's position in the environment [10,107,126]. These specific cues and affordances are given by default in an AR scenario compared to VR where the perception of affordances and experience of presence is dependent on the VR application meeting some requirements [36].

Content generation often requires extensive knowledge of programming and is not intuitive for novice users. For example, commercial content tools (e.g. Unity and Unreal) have a high learning curve that needs to be overcome to become adept at authoring course content. This increases the workload of the instructor, who may not be skilled in programming or have knowledge of content creation tools. In addition, the instructor cannot readily visualize how the student will interact with the tool. To combat this, situated authoring has been explored in VR/AR contexts [25, 104], where the instructor can author from within an AR/VR environment. However, a comparison of such systems with a traditional user interface has not been done. It is unknown how authors perceive or perform in an immersive environment compared to a more traditional system.

This work aims to provide insights into the differences between authoring SBT scenarios on Desktop and Augmented Reality interfaces. Two applications are described and a novel between-subjects study is carried out. Participant performance was measured by task completion time and the number of completed tasks. Perceived usability was gathered and analyzed as qualitative data.

3.2 System Design

Two systems were developed based on the same conceptual model and having the same functionality, however, they differ in the interface and interaction techniques which were motivated through literature research described in the next section as well as own experimentation in iterative building approaches. An additional application was created to visualize the AR content generated by the users in the study. Our authoring systems have three major functions: (1) place objects in the scenario, (2) attach attributes to objects and (3) define actions between them. **Attributes** are multimedia assets such as texts, audio files and questions. Objects in the scenario can have associated attributes that represent specific behaviors on the scene. For instance, a virtual phone placed in the scenario can have an associated sound for ringing, a voice mail sound, a text message or whether it can be picked up at runtime. An **action** represents an interaction between two objects, which defines the interaction mode and attribute that will be triggered, on the second object. Since the study's goal is to do a fair comparison, users are not required to input information. Instead, they are given files as texts or audios representing this data. Text input is a complex problem and is not part of the scope of this work.

3.2.1 Design Goals and Considerations

The end goal of this design is to allow participants to construct a scenario-based training (SBT) experience without any coding involved. AR instruction has previously been demonstrated as a valuable method for procedurally providing guidance [147, 151]. Our systems were inspired by the efforts exploring authoring tools made by Norton et al. with a desktop interface [107], Ens et al. with a VR system [25], and Ng et al. with an AR tool [104].

The conceptual model follows the nodes and links paradigm, defining a node as a virtual object in the scene with **attributes**, such as texts, sounds, questions, or whether an object can be picked up at runtime. A link characterizes an **action** between two objects that triggers a specific attribute (see Figure 3.2). This model represents a specific case from [1], defined as an atomic action. Visual programming can ease the coding learning curve as shown by tools such as Scratch [119] and Alice [21]. This representation has been used widely in commercial tools such as the Unreal Engine [30] and Amazon Sumerian [128]. Recently Unity3D announced plans to natively support visual programming [140].

The following design guidelines for this study are based on scenario authoring literature from a pedagogical and content creation perspective.

- **Authoring from learner's perspective:** An AR authoring tool automatically gives the content creator an idea of how the learners will visualize the experience. Initial exploration is made in [17], where learners and authors use the same tool.
- **Use of atomic actions:** Our study is based on the ability of participants to generate a course of actions that a learner can follow. To achieve this, actions are modelled as a basis of the

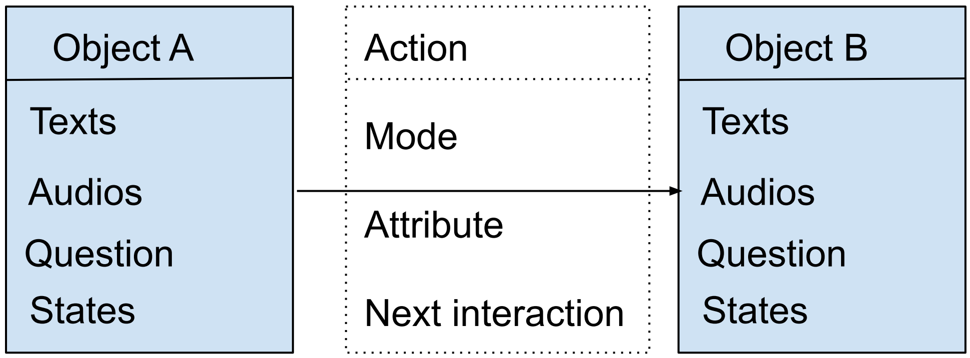


Figure 3.2: The most basic processing unit of the system. An object, A, interacts with another object, B. The mode defines the event that triggers an attribute e.g. collision or tap, which then enables an executable action.

interaction between objects. However, more complex models presented by Achour et al. can be explored in future work given the current architecture [1].

- **Authoring is determined as a sequential ordered set of actions:** Users authoring the tutoring experience can define the order in which actions execute; if the order is not specified, then the steps will be executed in the order they were created.

- **Architecture to support future work:** Given the modular architecture and the graph-based model, the system is scalable to support more complex conceptual models such as the one proposed by Achour et al. [1]. However, with complexity, challenges arise in the interaction and placement of information in the space. Additionally, the system can leverage Intelligent Tutoring System (ITS) capabilities from frameworks such as GIFT [135]. This was not used in this study but could be helpful for future work.

3.2.2 System Applications

The following describes the procedures involved in implementing the two conditions of the study. The third application allows participants to visualize the content authored by them.

AR Application

The visual assets and controls for the AR application follow the guidelines for designing mixed reality applications [96]. These visuals on the scene are collections of Interactable Objects. Floating panels arrange controls such as buttons that help the authoring process. An iterative design process led to the development of the AR condition.

First Iteration -

First prototype leverages the Hololens capabilities and uses HoloToolkit for Unity3D to generate a spatial user interface that allows for object placement, attribute selection, and actions definitions. Requests are sent to the Web server to save the content generation state. Means of interaction involve gazing, tapping on a clicker, and speech recognition (see Figure 3.3). Transform manipulation widgets can be invoked with voice commands: ”translate”, or ”rotate”. Initially, a spatial mapping of the environment is executed and can be stopped by saying ”stop”. A floating widget can be invoked by tapping on an object or interaction between objects.

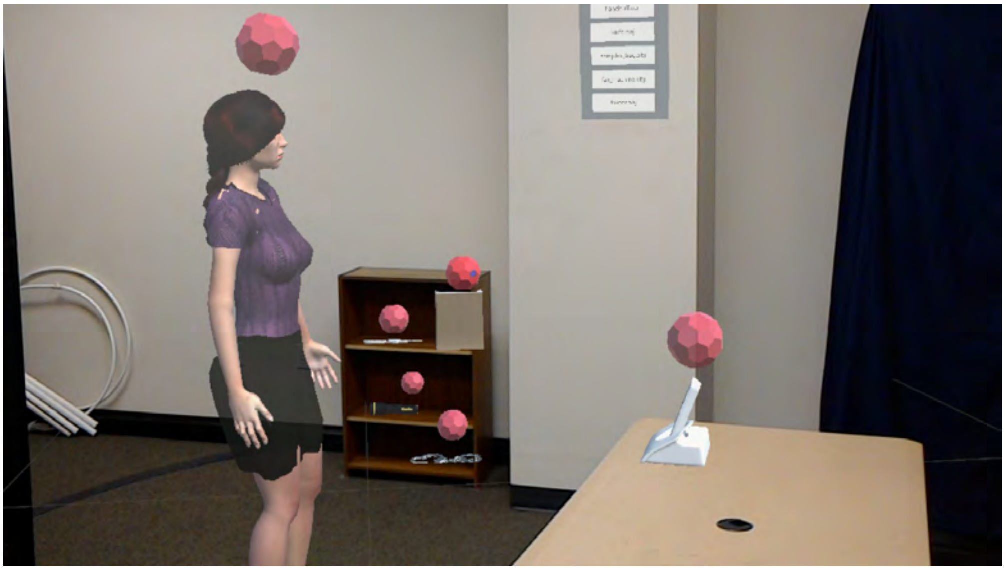


Figure 3.3: First AR prototype iteration with default HoloLens interaction techniques.

Adding Object Attributes. After a tap is performed on the phone and ”Add Text” is selected from the floating window, a file explorer is visualized with text elements that can be appended to the phone, see Figure 3.4. Text is added as an attribute by tapping on the corresponding icon which will remain selected. An attribute can be unselected by tapping again on the same icon.

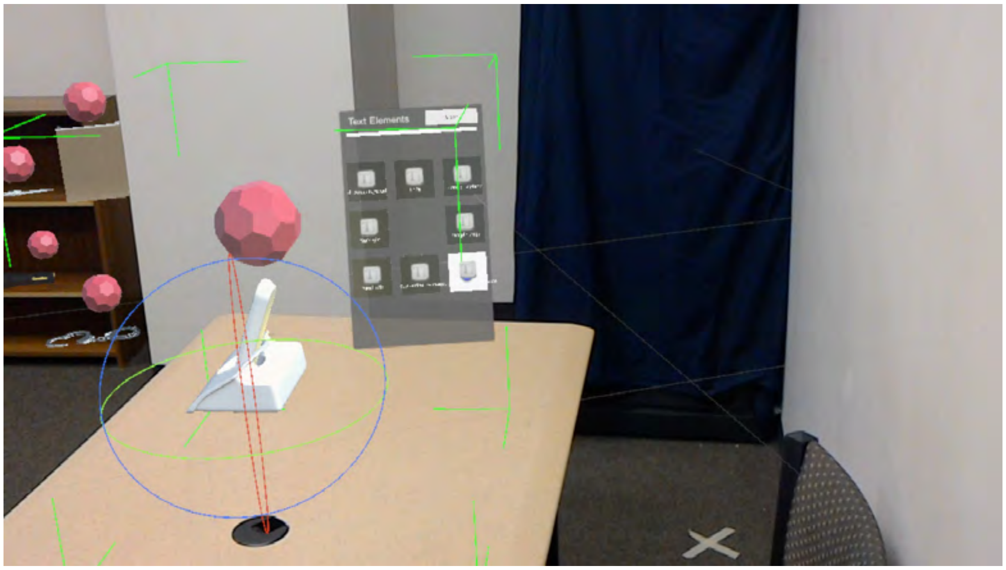


Figure 3.4: First AR prototype adding text to the virtual phone in the scene.

Action Between Two Objects. Every object added to the scene has an orb element on top. This sphere helps to establish a link between objects. The author taps on the node that represents the object, which then initiates the action, and a line will be rendered following the gaze pointer to another orb representing an object with attributes for execution. A label can be assigned using speech, by tapping on the white orb and saying a phrase. Another tap on the white orb will save the label. When tapping on the white sphere a floating widget will request to define the interaction mode, the attribute will trigger and the next action occurs. Figure 3.5 shows an ”inspector reading a voice mail,” the voice message is already an attribute of the phone and is present on the parameters of the connection.

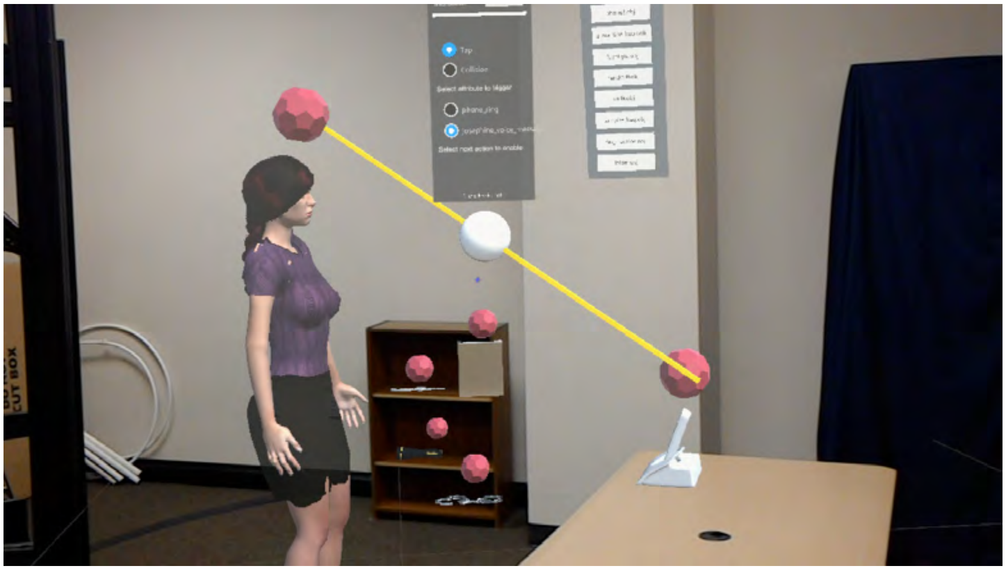


Figure 3.5: First AR prototype adding an action between the user and the phone to trigger a specific attribute.

In line with results reported by Ng et al., such as frustration and physical load [104], these techniques were not well received. In addition, floating panels were anchored to objects which produced excessive participant movement.

Second Iteration -

The first iteration provided us with insights towards improving input devices for a better user experience. The use of an external pointer instead of gaze reduced the head movement required for the interaction. A phone shown in Figure 3.6 was added as an input device, providing a rotation vector, which was used as a pointing device and a touchscreen to extend the interface for UI selection. A custom pointer mapped the phone rotation vector as a ray in the virtual world. The starting position of the ray is 40 centimeters down from the head and ends 5 meters in the ray direction. A cursor is positioned at the first hit point of the ray with any element on the scene.

Two touch-capable areas are defined for a physical button and a joystick fixed to the screen. Tap functionality is replaced by the tactile virtual button on the screen. Hand recognition is introduced as well to create actions between objects in the scene. A LeapMotion was chosen for accurate hand tracking. It is placed on a 3D printed structure fixed on top of the HoloLens (see Figure 3.6) and USB powered from a laptop carried on a backpack.

Transformations on objects are invoked once again upon voice commands, however, rotation and translation are executed on the XZ plane, using the joystick attached to the phone screen. Floating windows are positioned 1.5 meters from the user view in order to fit them in the HoloLens screen field of view. These UI panels follow the gaze of a participant. Content generated through the authoring is stored on a web server.

Figure 3.6: Second prototype: HoloLens coupled with a leap motion for hand tracking.

Adding Object Attributes. Pointing to an object and pressing the physical button invokes a floating window that shows possible operations to perform. The options are also shown on the phone screen and can be selected by touch. Figure 3.7, depicts the possible elements that can be added to the selected object. Users can select objects by pointing the mobile device and button pressing or by touch input directly on the phone screen. Attributes are unselected in the same way.

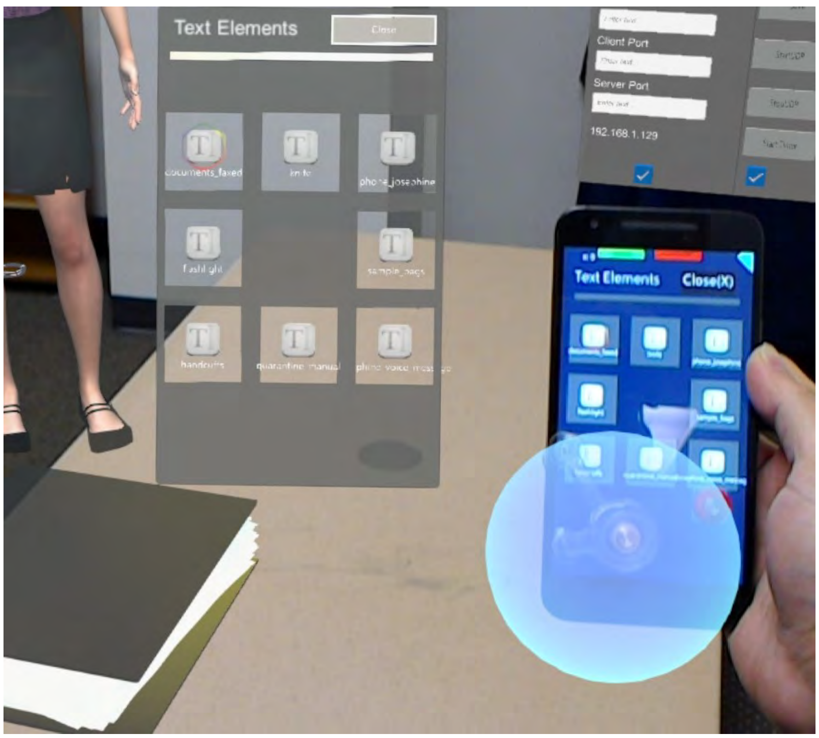


Figure 3.7: Second prototype: view from HoloLens showing both ways to interact with the menu.

Action Between Two Objects. An action is defined as a link generated by the hand collision with a virtual element followed by another hand contact with a different object in the scene. Hand is represented by a blue transparent sphere in the virtual world (see Figure 3.8). Parameters of the action are set by selecting the white orb allowing the user to place an identifier for the action by voice and then invoking a panel to set the interaction mode (e.g., tap or collision). This will cause the attribute to trigger and the next action to follow. An alternative interface is proposed as well, allowing participants to set action parameters from the phone. Participants freely selected which input method they preferred.

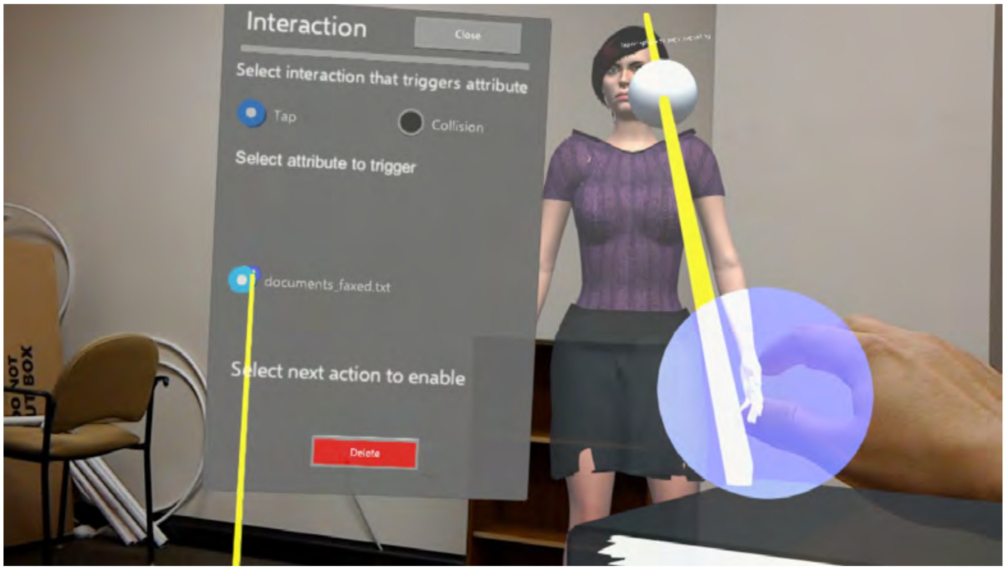


Figure 3.8: Second prototype: An action is established between the inspector (avatar) and a folder. The parameters of the action can be seen on the Interaction panel.

The second iteration explored different input modalities and the anchoring of 3DUI widgets to the user view following the best practices for maximum comfort [96]. Gaze input was replaced by using a phone as a pointing device; (see Figure 3.7). The phone screen was an extension of the 3DUI widgets and hand tracking from a LeapMotion positioned on top of the HoloLens was used to create actions as seen in Figure 3.6. The hand collision with a virtual object would start a link, and the subsequent hand collision with another virtual object defined the action. The pointer input was well received, but the phone UI caused a break in immersion. Hand interaction also induced a higher physical load in addition to tracking issues when not in the leap motion field of view.

Third Iteration -

The final iteration (see Figure 3.1 left) used feedback from the previous designs to generate a more friendly spatial 3D user interface. In line with a study from Poupyrev et al. [114] and 3D interaction techniques by LaViola et al. [74], a virtual pointer was chosen as the main interaction technique since objects in the scene are big and remote selection was preferred. The phone was replaced by the HTC Vive controller (see Figure 3.9). According to Niehorster et al. [106] "the Vive can be used for experiments in which the risk of losing tracking is small because the participant only moves in a small area", as in our experiment setup. Use of the Vive controller is also recommended when "a few degrees of offset in pitch and yaw measurements don't matter" and when all of the tracker measurements are not used, as in our case by just using the controller.

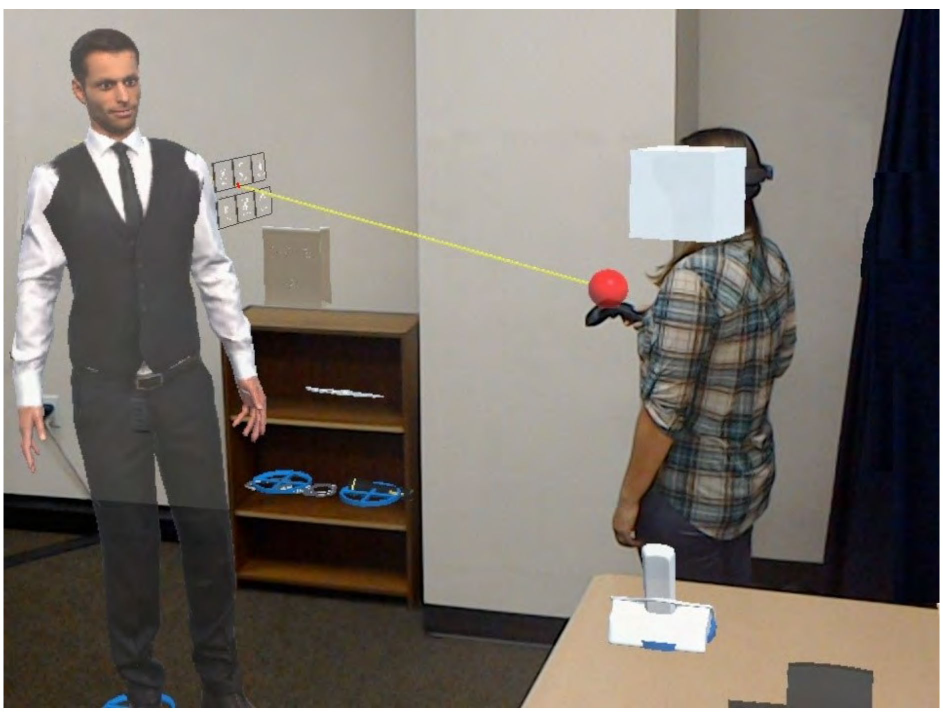


Figure 3.9: Third prototype: A user is interacting with the menu to author attributes of the virtual avatar using an HTC Vive controller.

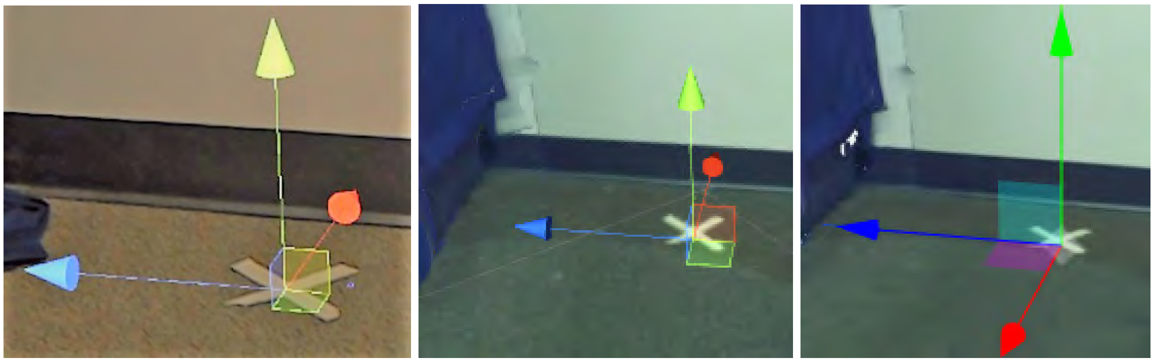


Figure 3.10: Left, picture taken with HoloLens indicating the coordinates system reference in AR. Center, the coordinate system in Vive space, Right, similarly the coordinate system in WebGL space.

The HTC Vive and Microsoft HoloLens coordinate systems were manually synchronized so that the controller transform is aligned with the AR counterpart. For the alignment, a point is set in the real world as an anchor point for HoloLens. The point is physically located on the floor between the table and the blue rack, as can be seen in Figure 3.10 and an ’X’ taped to the floor. In the Vive space an empty gameObject is added to the virtual scan on top of the ’X’ matching position and rotation. This process is repeated for the WebGL application. The three reference points in the three different spaces serve as an origin to which all elements are transformed to (see Figure 3.10). An additional transformation was applied to elements authored in Desktop when visualized in the AR playback (mirroring in the YZ plane).

A custom virtual pointer maps the controller 6 DOF as a ray in the AR world. The ray starting position is placed at the controller tracking sensors and ends 5 meters in the ray direction. A cursor is positioned at the first hit point of the ray with any element on the scene. The trigger and trackpad[1] buttons from the controller were used to interact with the 3D UI. The content generated is committed as a web request and saved in the Database.

[1]https://www.vive.com/us/support/vive/category_howto/about-thecontrollers.html

Desktop Application

The desktop authoring tool (see Figure 3.11 center) provides a traditional graphical user interface similar to a basic 3D editing tool. The backend and frontend are built on top of the Google Web Toolkit (GWT) [42] and Javascript libraries such as "Three.js" [15]. The frontend is developed following user interface design principles such as: task-related grouping, graphic layouts, metaphors, direct manipulation, and form filling [2, 131]. The interface layout comprises a file explorer for assets, a preview asset area, a 3D viewport for visualizing 3D content, a vertical bar for object transformations and a 2D canvas for graph manipulation. The interaction is performed by mouse and keyboard. The generated scenario is equivalent to the one produced by the AR counterpart. We chose to build the system as a web application due to the distribution flexibility and the rise of standardizing an immersive web [86].

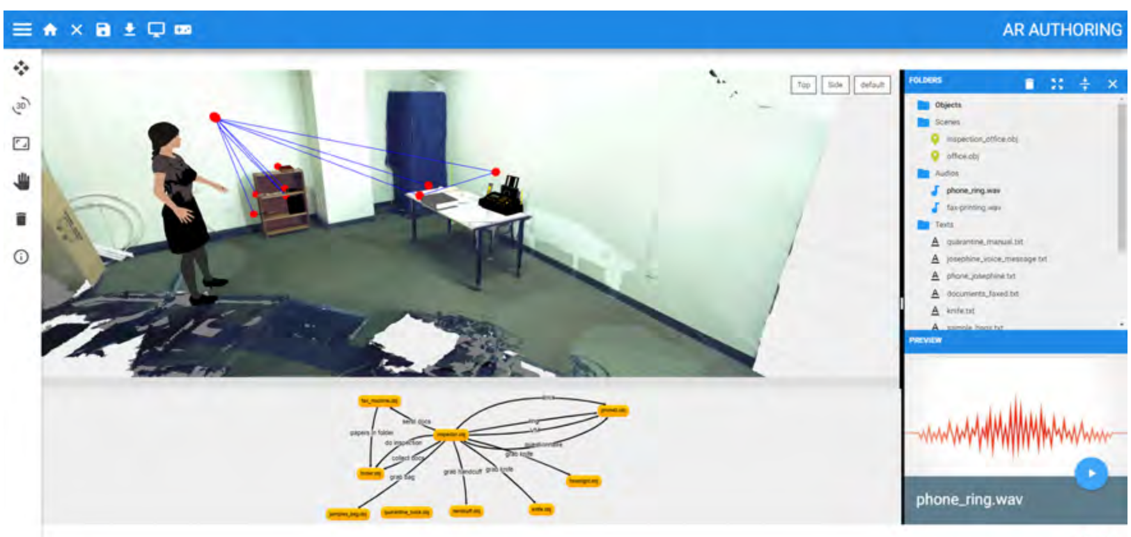


Figure 3.11: An example of the desktop system displaying a scenario authored with actions between user and virtual objects in the scene.

Playback Application

Once the scenario authoring is complete, a user can run the training course and visualize the elements placed with the respective attributes and actions attached. The playback is independent of the condition a course is authored with. The application reads a training instance from the server and displays the information accordingly to the data created. As shown in Figure 3.12, the phone displays a question with 5 answers for the user to pick. The orange arrow on top of an asset represents the next object the trainee should interact with. As depicted in Figure 3.12, the orange arrow can provide a scaffold so the trainee knows he/she must trigger another action using the fax to continue through the scenario.

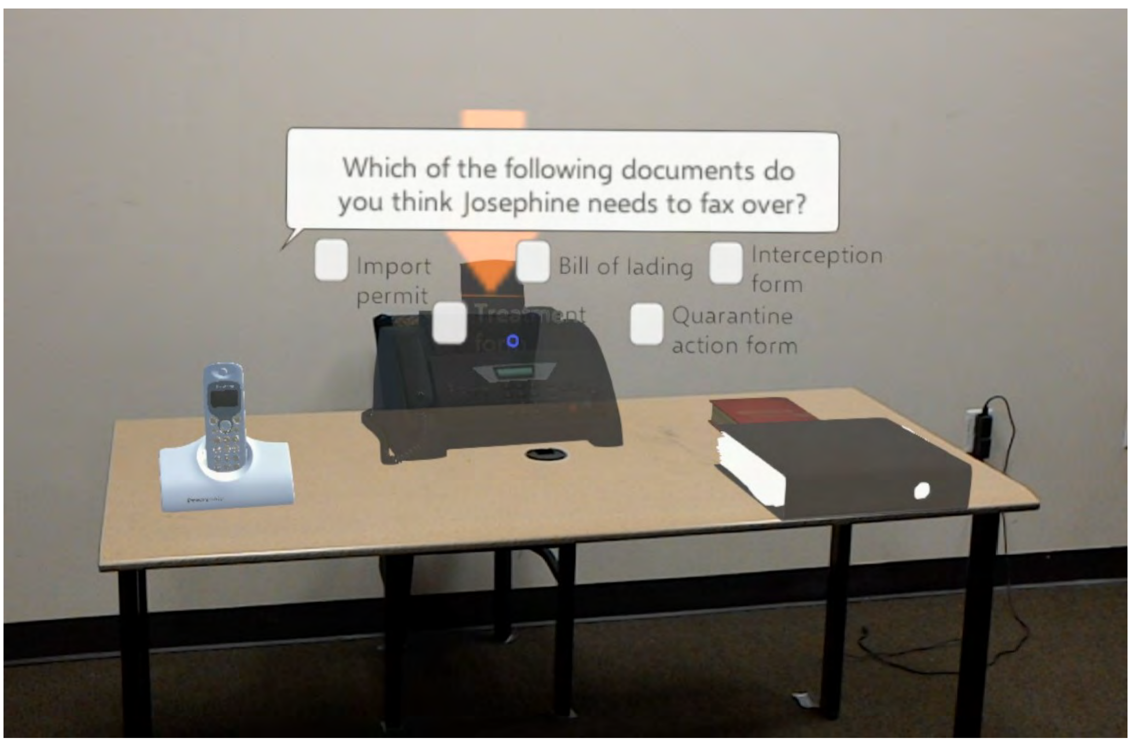


Figure 3.12: An example of the playback system displaying a question with 5 possible answers.

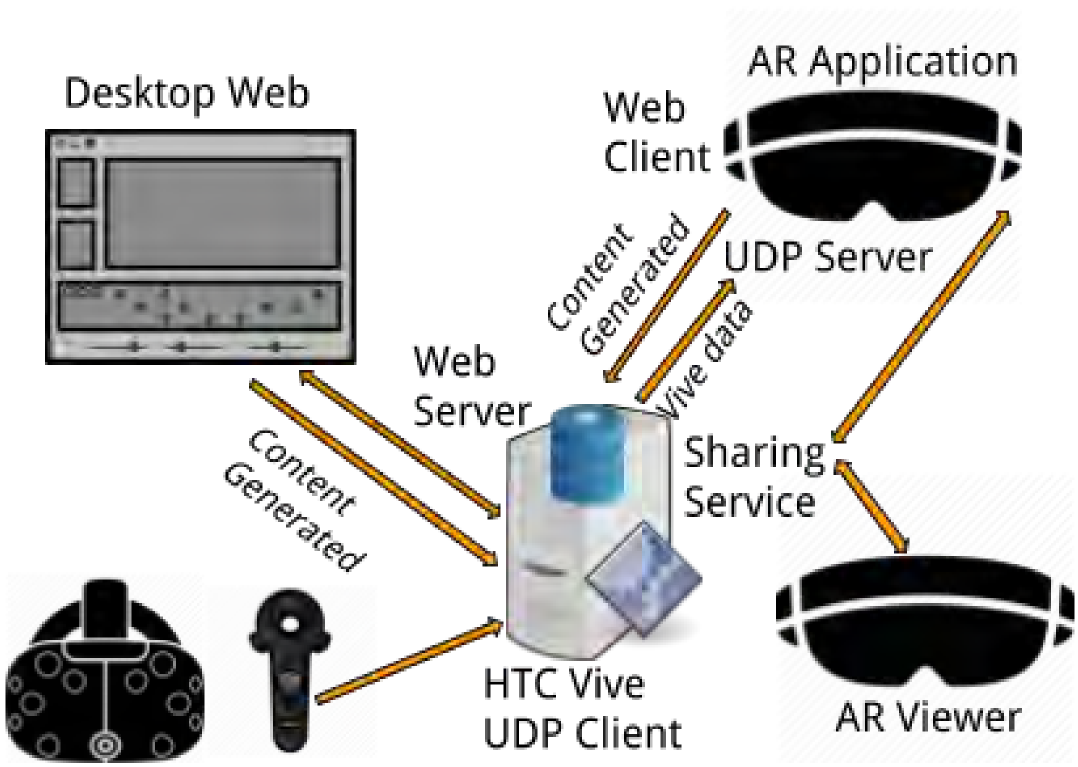


Figure 3.13: System Architecture: Orange arrows represent information flow between the systems.

Figure 3.13 describes the system architecture for both conditions. Applications follow a client-server approach where a PC is used as a web server to host the Desktop authoring tool and a Database to store the content generated state for both conditions. Both systems share the same database hosted on a local web server. The PC also runs a "Sharing Service" which is used to allow an additional AR headset to stay in sync seamlessly in real time. A HoloLens hosts the AR authoring tool, which sends the content generated via an HTTP Web Client to be saved on the Database through the Web Server. This application also runs a UDP Server to receive the HTC Vive controller 6DoF information and button states from a third application running a UDP Client.

The world coordinates from the Desktop, AR and HTC Vive applications are manually aligned. Space alignment is not a contribution of this work, and it was manually set using the controllers and physical elements of the scene in such a way that the three spaces share the same reference transform (see Figure 3.10).

3.4 User Study

An exploratory user study was conducted to find user preferences on usability and perception on the two interfaces presented. Quantitative and qualitative metrics were gathered, the first comprise missed tasks, misplaced objects and time required, followed by the second with post-participation surveys. A between-subjects design was used with half of the participants using the AR authoring on a PC and the rest using the Microsoft HoloLens AR authoring application. Both groups were trained on the same tasks and were assigned the same problem. Given this, the following hypotheses are proposed:

- **H1,** Desktop participants will take a shorter time to complete the study than those using the AR interface due to physical load differences and familiarity with Desktop interfaces.
- **H2,** Participants will find the AR authoring tool as enjoyable and usable as a traditional Desktop environment.

3.4.1 Use Case

Achour et al. [1] note that a scenario can be a story, use case descriptions, or a script. Based on this definition, our evaluation is constructed around the following scenario:

Create a training experience for dealing with a quarantine problem. The experience aims for the user to learn procedures to follow before a quarantine inspection is made. The office setup is provided with a phone, a fax machine, a quarantine manual book, a folder, and items for the shelf such as handcuffs, a flashlight, a knife, and a sample bag. An inspector who is the user should be placed in a starting position on the scene. A quarantine manual book is placed on the scene with important information about the documentation required. A phone rings and the inspector is required to take the call after the phone rings. An assistant Josephine requests an inspection leaving an audio message which is translated to text. The inspector is required to answer a questionnaire with the required documents needed for the inspection. After answering the inspector is required to phone Josephine for the documents. Josephine sends the documents by fax and is visible in the scene in a folder. A folder is placed on the scene for the documents faxed, the folder can be collected. This folder contains documentation for doing an inspection. This scenario has been adopted from the work in [107].

In the role of a creator, the user is given a set of tools that allow him/her to generate the training scenario experience. The scene replicates a real-life situation. The following tasks are required to be completed:

1. An inspector who is the user of your generated experience should be placed on a starting position in the scene marked with a label.

2. A virtual quarantine manual book is to be placed in the scene with important information about the documentation (**quarantine_manual.txt**) required to carry out the task.

3. A virtual phone is setup to ring (**phone_ring.wav**) at a user interaction.

4. The person calling is your assistant Josephine who requests an inspection (**josephine_voice_message.txt**).

5. Josephine asks a question to the inspector about which documents need to be faxed (**documents_josephine_fax.txt** contains the question with the right answer).

6. The inspector makes a call to Josephine through the fax machine (**phone_josephine.txt**).

7. The documents (**documents_faxed.txt**) are received by fax and placed on a folder on the desk.

8. An additional interaction with the folder will display a question (**do_inspection.txt**).

9. Two more items are placed on the bookshelf (flashlight and handcuffs) for the inspector to pick and assign descriptions (**flashlight.txt** and **handcuffs.txt**).

3.4.2 Scenario Selection & Preparation

Our study is based on previous research conducted to evaluate SBT. This experience is based on the "Quarantine Procedures" training introduced by Norton et al. [107]. The criteria for selection is as follows: 1) The scenario could be replicated in a traditional office space, 2) Elements of the scene can have attributes, 3) The experience demonstrates all user interface capabilities, 4) The scenario has real-world validity (it is not an unrealistic or impossible scenario) and it is reproducible.

The scenario problem narrative comprises virtual assets that can be placed on the real-world visible furniture, see Figure 5.9. First, the experimenter reads a narrative regarding the purpose and general nature of the scenario. Next, using the system, the participant is trained and asked to author such a scenario (see Use Case). According to the chosen scenario, a physical location was prepared with the following furniture: a desk, a shelf and a chair. The space dimensions were (4 x 3) / 2 meters in a triangular shape. The room did not have mirrors or glass due to the scanning device limitations, and constant lightning created an optimal environment for use with the Microsoft HoloLens device.

To ensure a fair comparison between both systems, a 3D scan of the room was acquired from a FARO ultra-portable Focus Laser Scanner at the highest resolution as seen in Figure 3.14. The generated pointcloud was further processed to produce a final mesh. An origin point was defined in both the real and virtual representation of the space. This point serves as an anchor location for registration when the completed scenario was demoed with the HoloLens. Before using the FARO, different iterations with depth cameras were made for 3D reconstruction, however, poor levels of realism were achieved. The scanned mesh was used in the Desktop condition.

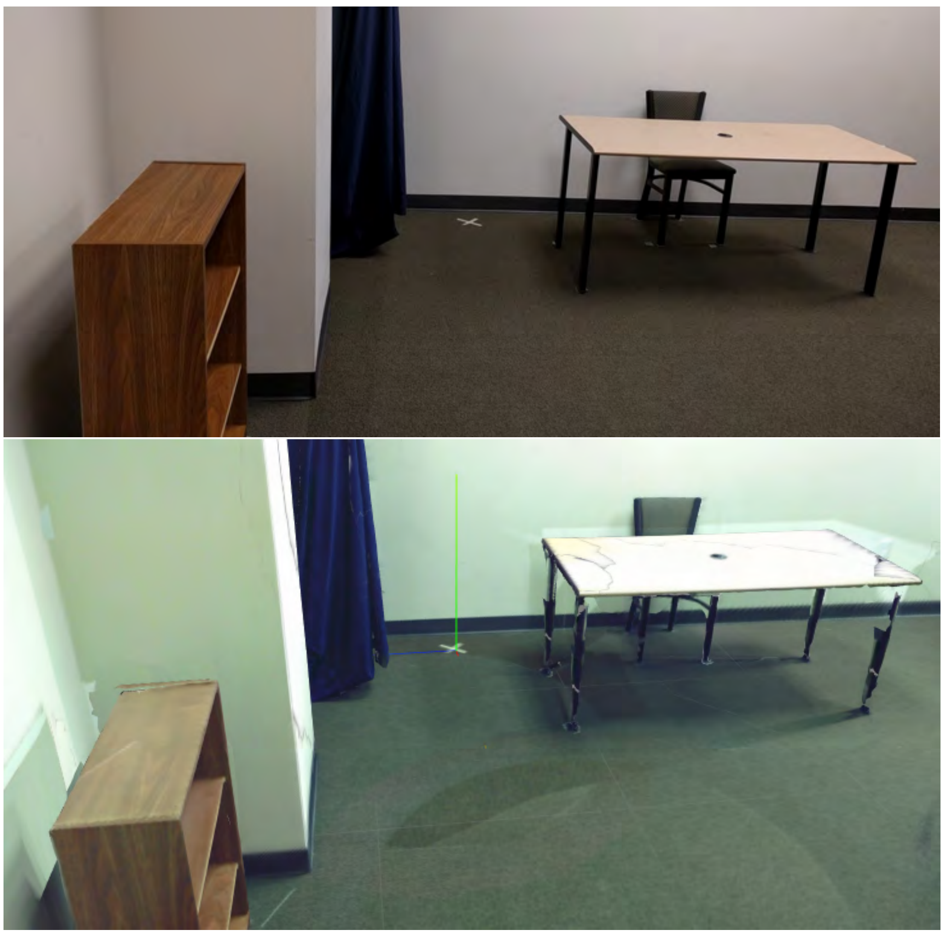


Figure 3.14: Top, a picture taken from the side of the room. Bottom, a screenshot taken from the Desktop authoring tool scene from approximately same position.

3.4.3 Tasks

In order to complete the tasks given in the use case, participants are required to place objects in the scene, add attributes to the items placed and create actions between these objects.

Placing Objects in the Scene

For the AR condition, objects can be selected from a floating panel displaying a list of virtual buttons, each button represents an object instance. Using the HTC Vive controller the user points at an element from the list and by pressing the controller trigger button, an instance of the virtual object is created that follows the controller pointing ray end. An additional trigger press fixes the object position in the scene, e.g. in Figure 3.1 left, an object is positioned on top of the table.

A user in the Desktop interface can select 3D models from the "Objects" folder list on the right side panel. An instance is created by left clicking and dragging a file to the 3D scene, after a click release the object is placed. Figure 3.1b, shows a virtual element being added to the scene. The objects can be transformed by using translation and rotation tools from the left vertical bar.

Adding Object Attributes

Selection in AR is invoked by pointing to the object and pressing the trigger button. The attribute panel is visualized showing operations that can be performed, see Figure 3.15. For instance, if "Add Text" is selected, users can then select attributes to add on a new floating panel by pointing and pressing trigger button on the file of choice. A colorful overlay will be displayed on the option selected which can be removed by pointing and pressing the trigger button once again. During selection, by using voice command "translation" or "rotation", objects can be transformed in the

XZ plane using the trackpad on the Vive controller.

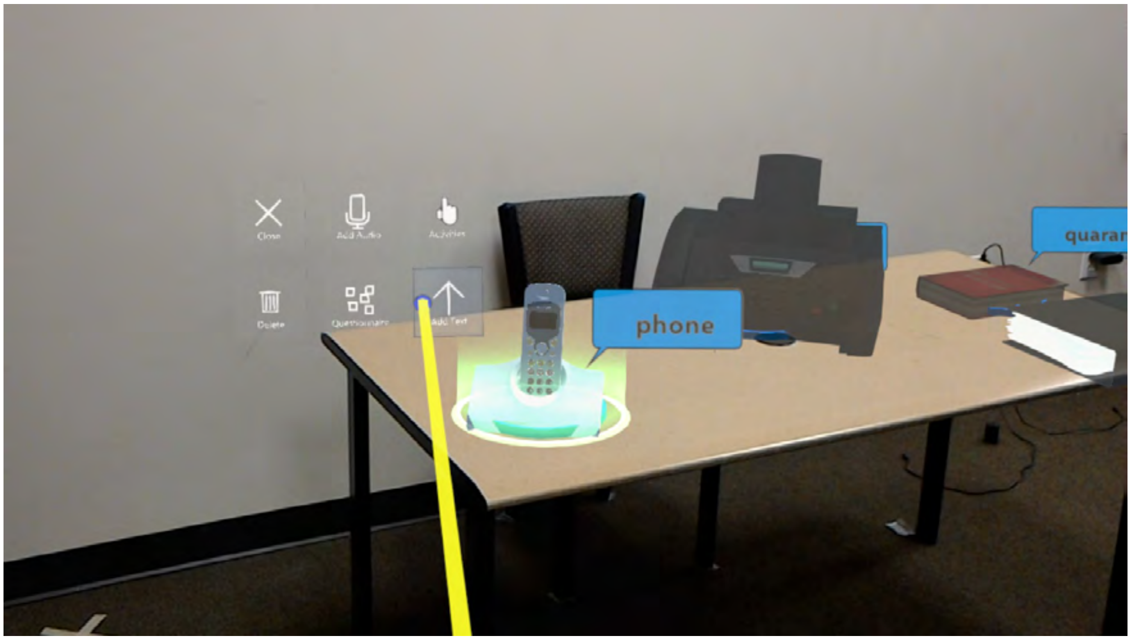


Figure 3.15: AR condition floating panel shows the six possible operations that can be performed on the object selected.

Selection is invoked on the Desktop by left clicking on the virtual object or the corresponding node from the graph area. Upon selection, an Attributes panel is visualized. Elements from the asset area can be dragged & dropped to the corresponding attribute category in the panel, as can be seen in Figure 3.16.

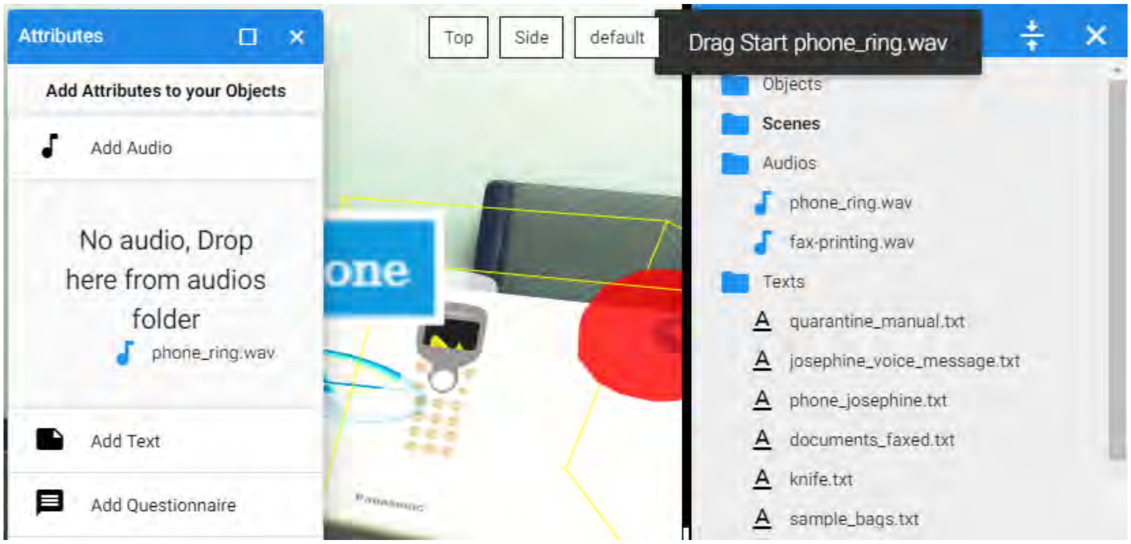


Figure 3.16: Desktop condition a sound is selected to be dragged and dropped on the Attributes panel belonging to the object selected.

Action Between Two Objects

For the AR condition, an action is defined as a link generated by pointing to an object and holding the trigger button, which produces a line that follows the pointer. The action line will be completed by pointing to a different object and pressing the trigger button. An example can be seen in Figure 3.17. Parameters of the action are set by selecting a white orb located on the middle of the line. Once selected, the user can place an identifier for the action using voice (voice is transcribed and set as an identifier). Finally, a floating panel is displayed to set the properties of the action.

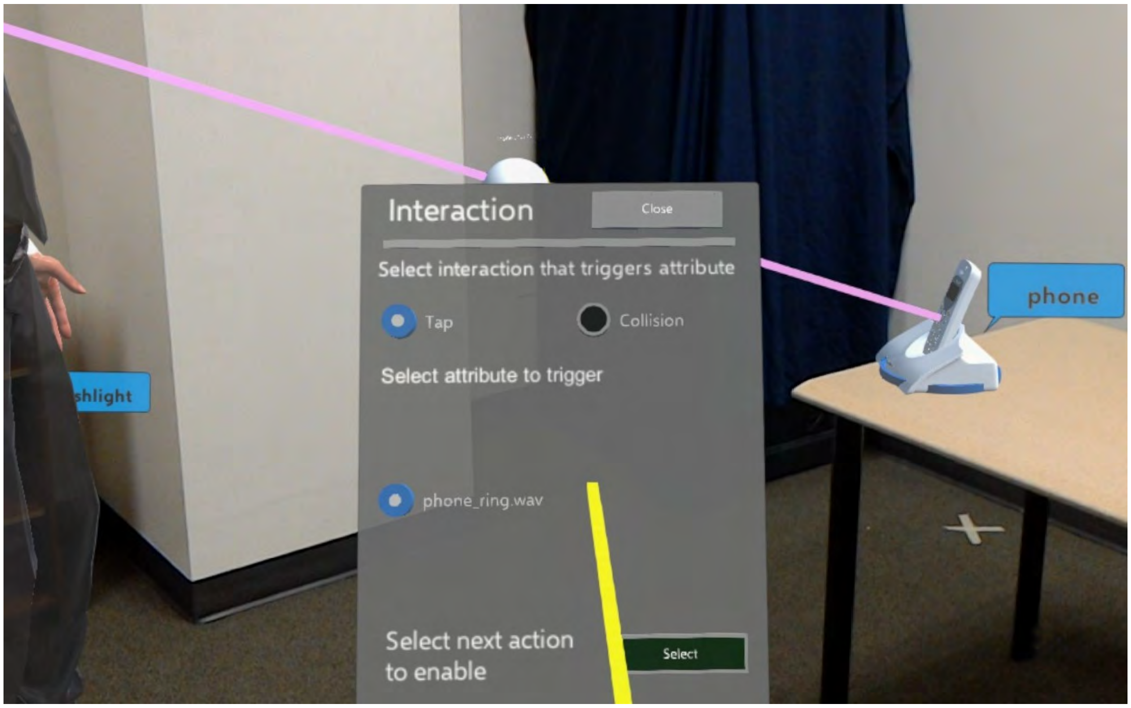


Figure 3.17: Action floating panel invoked after selecting the white orb on the connection line between two assets.

In the Desktop interface an object in the graph area is represented by an orange box. An action is created by a click on the box followed by a drag to another object, releasing the click will create a connection. An identifier is placed in the middle point of the arrow. A click on the arrow line displays an Actions panel which can be seen in Figure 3.18, which represents a relationship between two objects. Parameters of the action can be set on this panel.

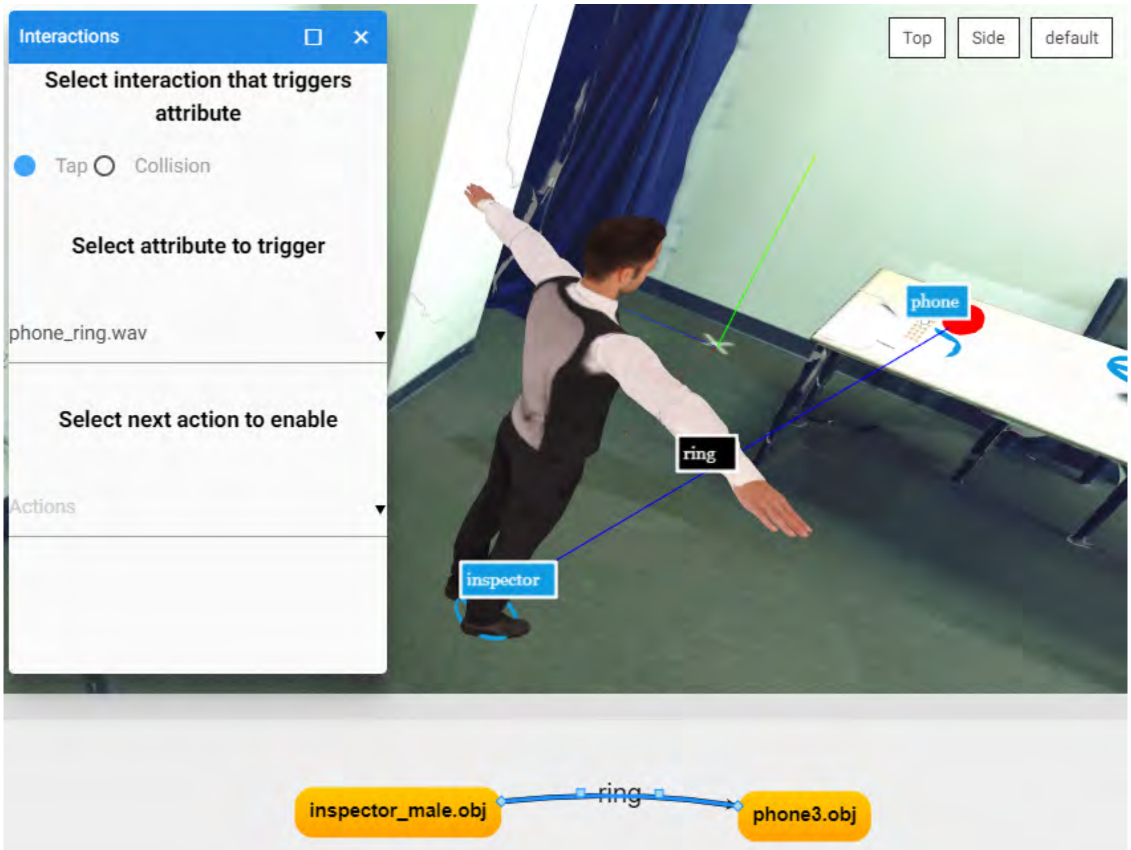


Figure 3.18: Action Panel invoked by clicking the connection between two objects (orange boxes). A label is used to identify the action.

3.4.4 Participants and Apparatus

Twenty eight people (16 male, 12 female) aged 18 to 39 ($\mu = 20.64, \sigma = 4.72$) were randomly distributed into two groups. Participants were recruited from a university population from a variety of engineering majors. A Likert scale from 1 to 7 with 1 representing "little experience" and 7 "very experienced" was used to measure the following in a pre-questionnaire: user experience with modeling toolkits & game engines ($\mu = 2.39, \sigma = 1.59$), participants experience in AR

($\mu = 2.19, \sigma = 1.52$) and experience with VR ($\mu = 2.30, \sigma = 1.38$). The experiment was either conducted on a PC (Core i7-6800K CPU, Nvidia GTX 1080 graphics card, 16 GB RAM) via a 55-inch flat-panel TV display, or on the Hololens. Another HoloLens was used for following up the user actions on the AR interface.

3.4.5 Study Design

Our experiment follows a between-subjects design with 28 participants randomly divided into two experimental groups. A pre-questionnaire to gather prior knowledge and a post-questionnaire (see Table 5.1) for user experience and perception were prepared. An additional System Usability Scale (SUS) [12] and NASA TLX [45] questionnaires were prepared. Each group was assigned to one condition.

3.4.6 Study Procedure

The study was designed to be completed in approximately 60 minutes for both conditions. Each group followed the same protocol. Initially, participants were asked to fill out two questionnaires about demographics and previous experience. Next, the problem was introduced for about 5 minutes, followed by a training session of 15 minutes on the corresponding tool randomly assigned to the participant. The training showed an example of a singular action task built on the interface by the proctor followed by a similar task performed by the user. After, participants were asked to solve the problem with the application provided and their execution was timed. Once the authoring was completed, they were shown the result on the HoloLens using the playback application. Then participants filled out a post-questionnaire (see Table 3.1) using a Likert scale from 1 (Very Little or Totally Disagree) to 7 (A lot or Totally Agree), a SUS questionnaire about user experience and perception of the usability of the tool and a NASA TLX questionnaire. Participants also had the option to write any feedback regarding the system or experience. Finally, the counterpart interface from the other condition was introduced to the participant in a 10 minute time frame with a similar training as in the beginning. Participants were asked about their perceived preference on which interface they would prefer to use and why.

Table 3.1: Post Questionnaire. Participants answered these questions on a 7 point Likert scale (1 = Very Little or Totally Disagree, 7 = A lot or Totally Agree).

#	**Question**
Q1	How much effort did you put into the authoring of the scenario?
Q2	I felt that the system was mentally demanding to use
Q3	I felt hurried or rushed using the system
Q4	The system was effective
Q5	I enjoyed using the authoring interface
Q6	The interface was challenging to use
Q7	The objects and assets in the scenario seemed realistic
Q8	I felt like I was building a scenario based learning experience
Q9	Please rate your level of frustration and stress when using the system
Q10	How physically demanding was the task?
Q11	How successful were you in accomplishing what you were asked to do?

3.5 Results

Quantitative data, such as task completion and time, were analyzed. The time distribution in both studies is shown in Figure 3.19. All participants were able to complete the authoring scenario and were evaluated on task completion. Qualitative data gathered with surveys (Table 5.1) was analyzed using a Mann-Whitney U test. The goal of this analysis was to demonstrate any differences between the Desktop and AR conditions (Table 5.2). Results show no difference in usability aspects, task completion and time taken.

3.5.1 Time

Figure 3.19 shows the performance time distribution in both conditions. A Shapiro-Wilk test on Desktop times shows the data is not normally distributed, therefore a Mann-Whitney U test was used and revealed no significant differences between AR (Md=19.6, n=14) and Desktop (Md=18.1, n=14), $U = 76.0, p = 0.31$.

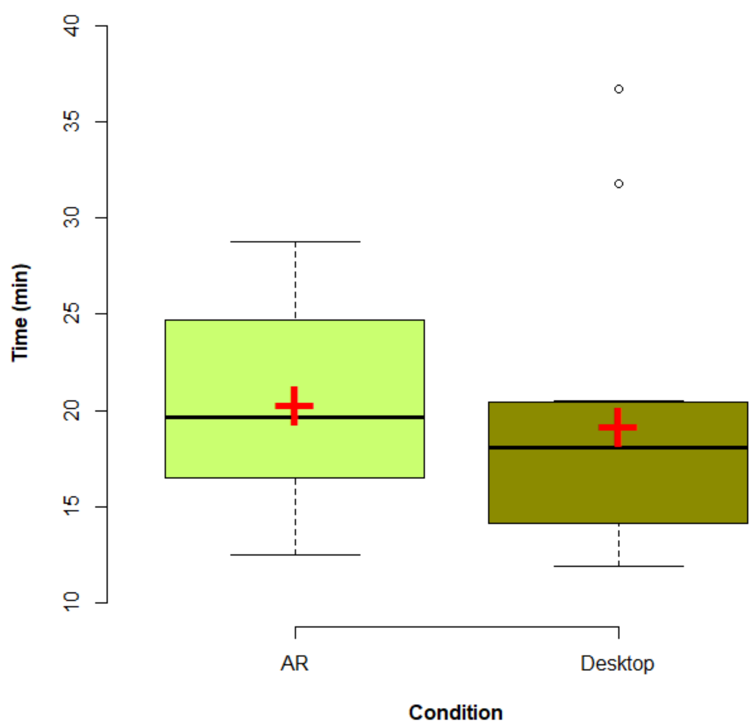


Figure 3.19: Box plot shows the mean "+" and median "-" times taken by participants under each condition.

To better understand how participants spent their time through the study, a Mann-Whitney U test found a significant difference in time taken between AR (Md=1.2, n=14) and Desktop (Md=0.6, n=14) to start the first task $U = 34.0, p<.0025$. Following the same test no significant difference between AR (Md=9.5, n=14) and Desktop (Md=7.0, n=14) was found on placing objects and adding attributes $U = 89.0, p = 0.68$ neither on creating actions with AR (Md=11.3, n=14) and Desktop (Md=8.9, n=14), $U = 68.0, p = 0.17$. By analyzing each condition, AR participants spent a longer time creating actions than placing objects and adding attributes $U = 51.0, p<.032$. For the Desktop condition, no significance was found on time taken among object placement, adding attributes and actions created $U = 63.0, p = 0.11$.

3.5.2 Task Completion

The different tasks from the problem were divided into three groups by misses on: object added to the scene, attributes added to the correct objects, and actions generated to trigger such attributes. A miss was defined as when the user missed to perform a task. The total number of misses per participant was counted and a ratio was calculated. A Mann-Whitney U Test revealed no significant difference between the misses ratios for AR (Md=0.12 n=14) and Desktop users (Md=0.06, n=14), $U = 63.0, p = 0.1$. No object was missed from the scenario among both groups. In addition, an independent analysis was performed on object placement; an object was considered incorrectly placed when it was 0.1 meters away from the placeholder position assigned. No significant difference was found in the number of objects incorrectly placed in the scenario ($t_{18} = -1.146, p = 0.267$). However, from observations, participants in the Desktop condition had some problems when translating objects due to the camera perspective. Figure 3.20 shows two examples of participants' movement across the room.

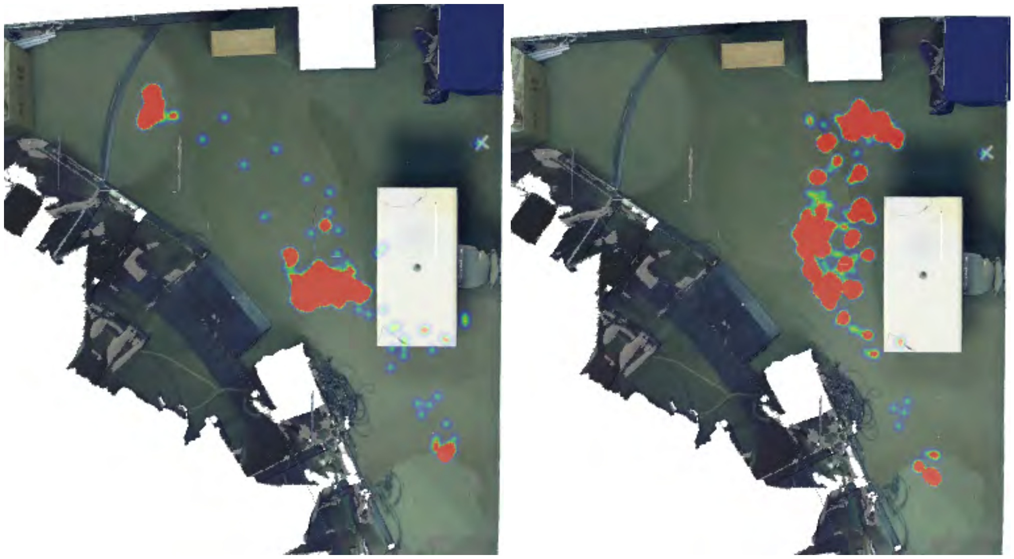


Figure 3.20: From HoloLens head tracking, heatmap of two participants' position while interacting with the objects during the experiment.

3.5.3 Usability and Perception

Table 5.2 shows the responses for each one of the questions from Table 5.1 presented to participants. A Mann-Whitney U test revealed significant differences in perceived system efficacy with lower scores for AR (Md=5.00 n=14) compared to Desktop (Md=6.00, n=14), $U = 49.5, p < .023$. Additionally, the Mann-Whitney U Test revealed a significant difference for perceived feeling like building an scenario based learning experience between the AR users (Md=5.00 n=14) and Desktop users (Md=6.00, n=14), $U = 55.0, p < .041$. No significant difference was found in effort (Q1), cognitive load (Q2), challenge (Q6) and frustration (Q9), showing that the participants perceived both interfaces to be equally usable despite hardware limitations and higher physical load. Consistent with findings in the time section above, participants did not feel hurried or rushed while using the system in either experimental condition (Q3). Finally, a Mann Whitney U test revealed no significant differences in the SUS scores between the AR (Md=50.00, n=14) and Desktop condition (Md=55.0, n=14), $U = 68.5, p = 0.18$. These SUS scores show relatively poor usability for both interfaces; there is room for improvement.

Table 3.2: Results from Table 5.1 on mean responses between Desktop and AR using a Likert from 1 to 7 on perception about each condition.

Q	**Desktop**		**AR**		**Z**	**p**
	Mean	Median	Mean	Median		
Q1	5.000	5.000	5.143	5.000	0.024	0.980
Q2	3.000	3.000	3.071	3.000	1.317	0.906
Q3	1.714	1.000	1.786	2.000	0.205	0.581
Q4	**6.000**	**6.000**	**5.000**	**5.000**	**-2.007**	**$<$.023**
Q5	6.357	7.000	5.571	5.500	-1.527	0.063
Q6	2.857	2.500	3.214	3.000	0.354	0.638
Q7	5.429	5.500	5.143	5.000	-0.034	0.486
Q8	**6.143**	**6.000**	**5.429**	**5.000**	**-1.746**	**$<$.041**
Q9	2.071	2.000	2.500	2.500	-0.853	0.197
Q10	**1.071**	**1.000**	**1.786**	**1.500**	**-2.262**	**$<$.012**
Q11	3.643	3.500	2.929	3.000	-0.432	0.333

3.5.4 Workload

Scores for each subscale of the NASA TLX were acquired using the unweighted (raw) score procedure. A raw TLX was chosen due to its shorter length and similar sensitivity to the full TLX [44]. Figure 4.13, presents NASA TLX workload ratings mean values and standard errors for each NASA TLX subscale. Each subscale is represented as follows: Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Own Performance (OP), Effort (EF), and Frustration Level (FL). A Mann-Whitney U test revealed significant differences in Physical Demand between AR (Md=25.51, n=14) and Desktop (Md=15.31, n=14), $U = 54.0, p < .013$. Also, a significant difference was found in perceived own performance with AR (Md=33.67, n=14) and Desktop (Md=52.04, n=14), $U = 56.0, p < .047$.

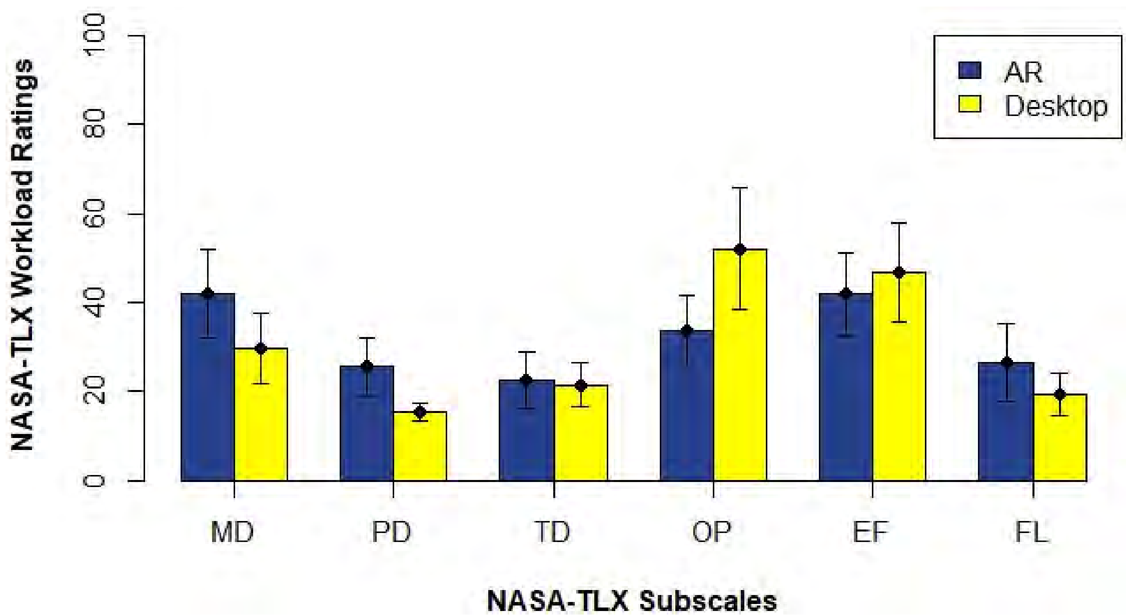


Figure 3.21: Plot shows the mean values and standard errors for NASA TLX workload ratings.

3.5.5 *Perceived Preference*

After completing the experimental task, participants were introduced to the interface they did not utilize for the experiment (either the Desktop or AR interface). We wanted to gather information about participant perception on different aspects of the interfaces (see Figure 3.22). Most users agreed that the AR interface can make the authoring experience look more enjoyable, even when they were not assigned to the AR experimental condition. Conversely, creating object interactions was not preferred on the AR condition, due to reduced visibility when the number of nodes connected increased. This is further expanded in the discussion section.

Table 3.3: Preference Questionnaire. Participants selected which interface (Desktop or AR) they would prefer on different aspects of the experience.

#	**Question**
Q1	Which interface type makes the authoring experience look more enjoyable?
Q2	Which interface type would make it easier to place virtual assets?
Q3	Which interface type would make it easier to create object interactions?
Q4	Which interface type would you choose if you were asked to create a scenario based learning experience?
Q5	Which interface type was looking more user-friendly/easier to use?

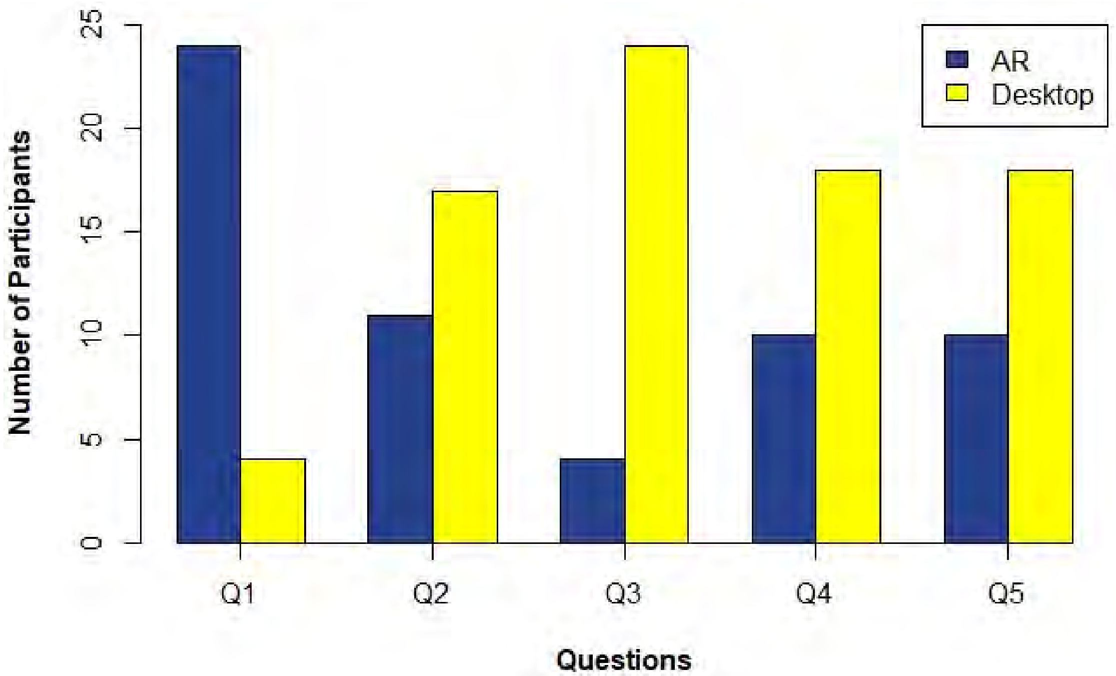


Figure 3.22: Participants preference for each one of Table 3.3 questions. AR in blue, Desktop in yellow.

3.6 Discussion

This experiment demonstrated that both interfaces were equally usable and yielded no significant difference in performance. Overall, users completed the task assigned in both conditions. Therefore, the attributes and actions model itself was easy to understand. Although both systems were meant to have the same functional features and outcomes, it is worth mentioning the differences found while building both SBT Authoring Tools (see Table 3.4). Below our findings are discussed and future directions for the design of AR situated authoring are given.

Table 3.4: Differences on preparation requirements and interfaces between a Desktop and AR SBT authoring interface.

	Desktop	**AR**
Screen Resolution	1920 x 1080	852 x 480
Scenario Navigation	Exocentric	Egocentric
Input	Keyboard, mouse	HTC Vive controller, voice
3D scene reconstruction	Yes	No
Spatial mapping	Yes	Yes
Remote Authoring	Yes	No
Immersion	No	Yes
Physical load	No	Yes

3.6.1 Global Progress and Visual Crowding

Despite no significant difference was found on task completion, in AR, participants had a limited field of view (35 degrees) of their authoring state with HoloLens, as opposed to the Desktop interface, where they had a global perspective.

> AR User 12: *"On the desktop you can see everything available at once, but in the AR I had to remember where things were."*

> Desktop User 10: *"The desktop UI is easier because I have everything in a compact screen"*

The scene course flow is still difficult to follow, and better analysis is required to find a way to visualize relationships in the space when the number of elements in the scene grows.

AR Users 3 and 9: *"Setting up multiple interactions between multiple objects can look messy."*, *"connections themselves were difficult to distinguish between when there were more than a few connected between the same pair of objects."*

This poses a challenge on how to best use space with virtual cues without limiting participant visibility of their current progress or interactions with other elements.

3.6.2 Authoring Time

Despite inexperience using the HoloLens, the need for navigation in the environment and controller adaptation, no significant difference was found in the task completion time between both groups. Contrary to our beliefs, hypothesis H1 cannot be confirmed. Furthermore, by analyzing individual interactions with the system (placing objects, adding attributes, creating interactions) time taken by participants could have been influenced more by their thinking process than by difficulties with the system. A significant difference in the time taken to start the first task could be due to the headset novelty effect. For the AR condition, allowing participants to generate actions by building a 3D graph of connections in AR resulted in participants spending a significantly longer time than in the rest of the interactions with the system. This is an aspect to improve in further iterations. For the Desktop, no significance was found between the individual system interactions; however, from observations, participants with no familiarity with the use of gizmos manipulators or the change in cameras perspectives had an added extra time while positioning objects in the right place, below some comments from participants at the end of the experiment.

Desktop Users 4 and 1: *"It was easier to put the assets in the reality one because you did not have to deal with the xyz thing."*, *"Have a button that highlights the item you are looking for so you can grab it if the item is behind a shelf or under table."*

3.6.3 Desktop vs AR Authoring Tools

Authoring a scenario in the Desktop interface required additional work to prepare the scene. First, due to the possible effects of visual realism on participants [79], a very realistic 3D reconstruction of the scenario was generated for the Desktop condition. Second, an anchor point was required to be set in the scene for registration with the WebGL scenario. Finally, coordinate systems can be also right or left handed which requires a transformation. Participants were not aware of this preparation as they were given the tools ready to use. Authoring in the AR interface required minimal preparation (just set an anchor point) as the scenario space can be mapped by the HoloLens. An additional difference involves the possibility of doing remote authoring, while AR presents an advantage if physically present at a scenario location, the Desktop interface and an eventual VR interface could enable remote authoring.

Input techniques for AR scenarios is highly dependable on the task nature. For this work a controller is a suitable interaction device for positioning and selection tasks [114]. No significant difference was found in object placement accuracy. However, a different scenario, for instance the authoring of an assembly task, may require different interaction techniques. Familiarity with traditional input devices for the Desktop interface is an advantage over AR/VR conditions.

3.6.4 Usability of the Authoring Tools

Results from Table 5.2 show no significant difference in enjoyment between the groups confirming hypothesis H2. Nonetheless, it was observed that the experience was less enjoyable for participants that felt the HoloLens was heavy to wear. Two users reported eye strain and two others headaches and took longer than 20 minutes to complete the task. For them the discomfort reduced the enjoyment.

AR Users 3 and 11: *"Eye strain was a bit of a problem after 20 minutes."*, *"If the AR were to maybe have a less heavy headset and better user interaction, I would definitely enjoy it more."*

From the user's comments the authoring experience was found to be more visually appealing in AR than the Desktop interface. The AR interface was well-received, despite well known limitations, such as limited field of view.

AR User 14: *"Personally I preferred to use the Augmented Reality because I feel more enjoy doing it and also it is more realistic when using Augmented Reality."*

Desktop User 3: *"the act of using ones body and looking around the objects as one does naturally was a very pleasant experience compared to the keyboard and mouse approach."*

Finally, participants expressed their feelings about the tool aspects that made their overall experience better. For the Desktop interface, people emphasized that using the modeling graph area to create actions and the drag&drop nature of the system was more favorable.

Desktop User 2: *"The arrows to connect interacting items make it easy to see what kind of interaction will happen, made my overall experience better."*

For the AR condition, participants greatly appreciated the use of the controller coupled with interactions in the AR scene:

AR User 9: *"The conjunction of Vive and HoloLens tech was a unique and enjoyable experience. The ability to see both real and alternate reality at once was quite satisfying."*

Another characteristic users enjoyed was the ability to visualize what they built with the playback feature:

> Desktop User 4: *"Seeing my creation come to life made my overall experience better"*

They also valued the situated interface (AR) as a mean to visualize the scene right away rather than imagining it while building it on the Desktop interface:

> Desktop User 14: *"AR was more user friendly because I can see the things working which I have imagine in desktop."*

> AR User 10: *"I feel that in regards to acting out the scenario, the augmented reality would be much more beneficial as you would actually be "in character" so to speak"*

The poor SUS scores (a score of less than 68 is considered below average usability) can be related to the low familiarity of participants with the types of tasks users performed in the experiment, such as performing camera placement or in some cases misconceptions from known interfaces such as trying to do object selections with double clicks (not used in the system). In AR the SUS could have been affected by some of the limitations described in Section 7.7. The focus was to build both systems equally capable and usable to ensure a fair comparison, and in this case, there was no significant difference between SUS values, but the usability of both systems can be improved as stated in section 7.6.

3.6.5 Similar Studies

Similar work presents advantages of AR over VR for selection and manipulation tasks. For instance, Krichenbauer et al. [73] found that VR participants required more time to complete a task than AR independently of the input device utilized. Even though our approach is holistic, similar results for task completion favor AR with less objects misplaced by participants than Desktop. Our authoring tool can also be further extended by analyzing how different 3D interaction techniques can enhance user experience using selection and manipulation. Work by Bellarbi et al. [7] show a study evaluating a novel technique for distant objects selection and manipulation versus the HOMER [9] approach. Authoring tools for AR can also help in assembly and maintenance tasks as in the work done by Gimeno et al. [34], while the context is different from scenario based training, results show high acceptance of the 3D authoring for such tasks. Finally, in line with results found in [104] situated authoring of AR scenarios was well received and enjoyable for participants, therefore future work should take into consideration the recommendation for building games provided by [104].

3.6.6 Recommendations for building AR SBTs authoring tools

While no significant differences were found in task completion and time, participants perceived the Desktop interface as more efficient, and the tool of choice if requested to author an AR SBT experience. In addition, physical load was reported as significantly higher in AR, which hinders augmented reality potential for now. This chapter recommends authoring AR SBT in Desktop and visualizing the results in AR. An additional study like the one made by Holm et al. is required to analyze how a combined approach might be more beneficial than building independent tools [49].

Further research needs to consider visualizations that allow participants to easily follow their work progress. While the 2D graph model was appreciated in the 2D context, it did not translate well to 3D as things got more complex. We recommend in future experiments to create task units allowing each individual to work in each unit at a time e.g. in our use case, 9 unit tasks can be identified. An additional component such as a list can help visualize the order in which tasks are executed.

This is also recommended for the Desktop interface.

In a formative study participants were given the option to use the touchscreen of their phone to fill in the forms. Users did not find it pleasant to switch between pointing & selecting in AR and selecting options with a touch screen. The Vive controller was well received in general despite sensitivity and latency reports. Unfortunately, at the time of the study there was no commercial AR headset with built-in controller as is the case now with Magic Leap AR headset [29]. This work recommends using laser pointing for selection and filling forms in room size scenarios.

Finally, participants understood well the use of floating panels to input information which is a familiar paradigm taken from the 2D counterpart. In contrast, link generation between objects could have been used more as a means of visualization than an actual requirement to interact with the system. It is recommended that operations on the virtual objects are simplified to the use of floating panels with 3D UI controls or more novel means of interaction. Visualizations can be then generated from those operations as the line relationship in our case.

3.6.7 Limitations

This work acknowledges limitations on hardware, input technique, device familiarity or novelty effect and specificity of the authoring scenario. Hardware limitations are given by the use of the HoloLens in the sense of a limited field of view (FOV), device weight and possible fatigue from use. The limited FOV might be one reason that there is no difference between AR and Desktop and people felt that the desktop was more usable. The motivation behind exploring AR in a real environment rather than on a simulated AR lead us to pick the latest commercial off the shelf see-through display available at the time. These limitations can be solved in the future with lighter HWDs with wider field of views. Our input device is limited by the possible latency generated from sending the information on a UDP network. Controller tracking information is sent and received at 20 frames per second. Current commercial devices provide built-in controller support for future studies.

There are limitations on the number of participants run in the study, for the web application distribution is easier however, for AR a more controlled environment is required. Users were more familiar in general with the traditional interface than the AR, and a novelty effect generated by the use of HoloLens could have influenced participants' decisions. To improve on this, a future study can use AR experts as subjects. Another limitation is the amount of training given to participants. Finally, the scenario is very specific and the results of this study can just generalize to room size scenario based authoring with constraints defined in Section 3. Future studies can evaluate other scenarios such as assembly tasks.

There are four aspects of this work that a real world setup would need to consider more; content curation, text input, 3D model pre-processing and a wider range of authoring scenarios. For Augmented Reality, these are still open problems out of the scope of this work. To lessen the influence of these problems in the experiment, participants were given the problem with the tasks to follow, in addition to text files and pre-processed models, scaling them to real sizes and centering their pivot points. The issue of curation was not considered part of this work. Another aspect is that our evaluation was focused on a unique use case rather than exploring a wider range of authoring scenarios.

3.7 Conclusions

Despite the potential of AR to facilitate authoring content for scenario-based learning, no compelling reason or motivation was found to recommend practitioners to move away from their Desktop tools. Functionally equivalent systems were developed using best practices for user interface design for both Desktop and Augmented Reality environments. However, it was found that authoring in AR afforded no real benefit in terms of performance time or perceived usability. To overcome these issues and deliver on the promise of AR, we believe researchers will have to develop entirely new and novel interaction techniques or focus on tasks that require unique visualizations beyond what is possible with a desktop interface.

Participants in AR particularly enjoyed using the controller to interact with elements of the scenario. These participants perceived the application more like a game than a productivity tool compared to the Desktop users.

This chapter evaluates participants' performance and usability of two interfaces to author AR scenario-based training experiences in a markerless setup. To compare the two interaction modalities, we developed a traditional GUI which produces the same outcome as the AR counterpart. Contrary to our expectations, H1 (participants taking a shorter time with the Desktop interface) cannot be confirmed. Despite the inexperience with using the HoloLens, navigation in the environment and controller adaptation, no significant difference was found in the completion time from both groups. The time taken for Desktop participants may not be as expected due to a higher learning curve with the Desktop interface compared to AR, where interaction may be easier to remember. Results from Table 5.2 show no significant difference in enjoyment between the groups confirming hypothesis H2 (participants finding AR as enjoyable as the Desktop interface). Nonetheless, it was observed that the experience was less enjoyable for participants that felt the HoloLens was heavy to wear. This work explored different setups and configurations for authoring AR SBT from a Desktop and AR interface. Also, it presents findings, challenges with proposed solutions and limitations to address for future iterations. While more work is needed, this work is a good starting point towards achieving usable and effective general purpose AR authoring environment tools.

CHAPTER 4: A TECHNIQUE TO AUTHOR VIRTUAL OBJECT COMPONENTS

This chapter is based on work published in: Masnadi, S., Vargas, A., Williamson, B., & LaViola, J. (2020). AffordIt!: A Tool for Authoring Object Component Behavior in Virtual Reality. Proceedings of Graphics Interface 2020.

4.1 Introduction

(a) (b) (c) (d)

Figure 4.1: These figures show a sequence of steps followed to add a rotation affordance to the door of a washer machine. (a) and (b) Cylinder shape selection wrapping the door. (c) and (d) component behavior authored and visualized.

Virtual object attributes such as text, audios or questions discussed in the previous chapter are a subset of behaviors to be authored. Objects possess more complex characteristics as they are composed of parts or components e.g. a drawer with doors. Authoring such interactions is studied in this chapter and a methodology was developed and evaluated with an exploratory usability study in a VR context.

As the prevalence of virtual reality increases for simulations and video games, there is an increasing desire for the development of virtual content that is based on real scenes and environments. A problem arises when a domain expert whose technical skills are based in realistic experiences necessary to a VR scene, but not asset creation (a situation described in Hughes et al. [50]) which are needed to build a virtual scene. To alleviate this problem, recent research has been focusing on frameworks to ease domain experts authoring process as seen in [25,35,104]. 3D scene reconstruction [100, 103, 139, 152] provides a suitable solution to the problem. Initially a 3D reconstructed environment will be composed of a continuous mesh which can be segmented via autonomous tools as shown in George et al. [31] and Shamir et al.'s survey [129] or human in the loop solutions as seen in [105, 146].

However, these tools fall short at identifying and applying affordances, the intrinsic properties, of the components of the object. For example, a storage cabinet may be segmented from a larger mesh, but the movements of the cabinet door remains absent. One solution is the use of a 3D modeler, such as Autodesk Maya [101] or Blender [11], but if the domain expert is unfamiliar with the software then a technical expert in asset creation is required. This solution carries a cost, however, as the domain expert's own intuition and understanding of an object's affordances could be lost in translation, either in relaying requirements to a third party or to software they are not experts of. As our solution we introduce AffordIt! an online tool that allows a 3D scene author to isolate key components of virtual content and assign affordances to it using their own intuitive understanding of the object.

In this work we define a 3D reconstructed scene as being a recreation of a real world environment that contains one or more virtual representations of an object captured within that environment. The component of an object is then defined as a segmented portion of the mesh that is not removed, but rather used to assign intrinsic behaviors. The term affordance originates from psychology and it is defined as a possible action that can be performed over an object (or objects) by an agent in an environment according to Gibson et al. [32, 33]. The affordance concept is broad and since its introduction has informed multiple research efforts within the Human-Computer Interaction (HCI) field. Our focus on this work is not to dive deeper into the concept but to acknowledge its use in fields such as robotics [43, 56]. We are extending this concept from robotics to VR with a prototype system named AffordIt!. The system is implemented using an HTC Vive Pro Eye headset with tracked controllers within a kitchen virtual environment.

AffordIt! provides an intuitive method for a scene author to select a region of interest within a continuous mesh and apply affordances to it using procedures outlined in [91,130]. Rather than relying on a sketch-based interface, we looked to the work of Hayatpur et al. [46], in which users could invoke a plane, a ray or a point to constrain the movements of a virtual object. As such, our procedure has a user first selecting a region of interest using shape geometry followed by defining a specific movement constraint. After processing the operation on the mesh an animation demonstrates the behavior attached to it as shown in Figure 4.1. We evaluate this technique in an exploratory study where perceived usability and workload of the system is collected and analyzed. For the study we only use two mesh cutter geometries and two movement constraint definitions, though the concepts of AffordIt! could apply to other selection geometries or affordance definitions.

4.2 Implementation

AffordIt! system architecture can be depicted in Figure 4.2. Our technique works by first cutting a mesh using simple geometries and then applying intrinsic behavior to the segmented portion. Both steps require interactions with a user to define the region of interest and the behavior. The user's interactions can be performed independently of the mesh manipulation. For the exploratory study we focused on two mesh cutter shapes and two behaviors which are defined below.

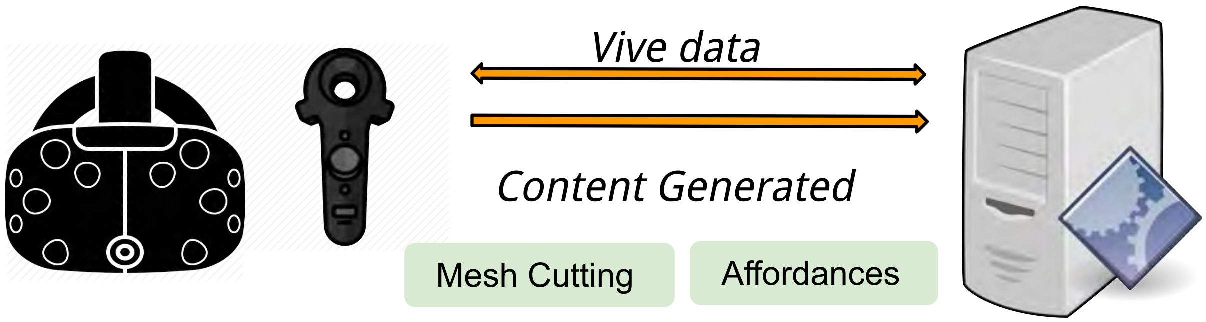


Figure 4.2: AffordIt! system architecture. Users interactions are stored in a local computer.

4.2.1 Mesh Cutting

For the cutting step a mesh cutter geometry is used to define the region of interest. When a cut is performed the original mesh is divided into two, one inside the mesh cutter and the other one outside. The algorithm clips the mesh using each face from the mesh cutter primitive using a brute force method as shown in Algorithm 1. The algorithm is derived from the Slicer implementation by CGAL [142] and is extended to be used on a more complex shape as the slicing tool rather than a simple plane. Triangles falling inside and outside the mesh cutter volume are segmented into two sets, while the ones that share vertices inside and outside the mesh volume are triangulated accordingly to fit in the appropriate set. The number of triangles on high polygon objects was reduced to optimize the cutting time to an order of magnitude of seconds.

In the exploratory study we focused on two mesh cutters, a cuboid and a cylinder, which will be created by the user using VR controllers. Mesh cutters can be extended to any shape, a possible approach is explained later in the chapter. The shape geometry editing is not part of the experiment but sheds some light on supporting different geometries for the mesh cutters.

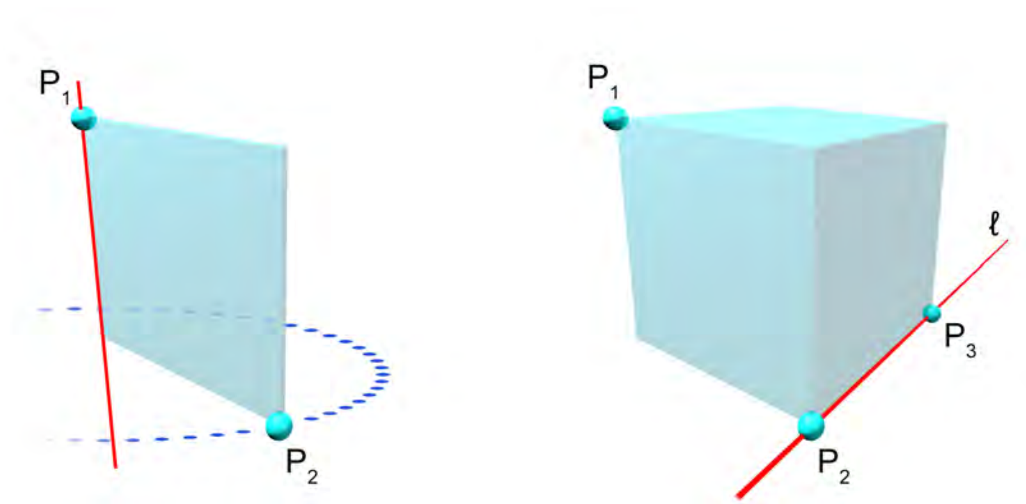
Algorithm 1 The mesh cutter algorithm

```
triangles ← getObjectTriangles()
selector ← getSelectorShape()
in, out ← List()
procedure MeshCutter(triangles, selector, in, out)
    for all triangle ∈ triangles do
        if selector.isFullInside(triangle) then
            in.Add(triangle)
        else if selector.isFullOutside(triangle) then
            out.Add(triangle)
        else
            CutTriangle(triangle, selector, in, out)
    return in, out
procedure CutTriangle(triangle, selector, in, out)
    vertsin, vertsout ← array[2]
    inCount, outCount ← 0
    for i = 0 to 3 do
        if selector.isInside(triangle.vertices[i]) then
            vertsin[inCount] ← triangle.vertices[i]
            inCount ← inCount + 1
        else
            vertsout[outCount] ← triangle.vertices[i]
            outCount ← outCount + 1
    tmpT ← Triangle()
    if inCount == 1 then
        tmpT ← Triangle(vertsin[0], vertsout[0], vertsout[1])
    else
        tmpT ← Triangle(vertsout[0], vertsin[0], vertsin[1])
    v1, v2, v3 ← tmpT.getVertices()
    ▷ /*getIntersectionPoint returns a point on line connecting the first two parameters where selector intersects the line*/
    Pt1 ← getIntersectionPoint(v1, v2, selector)
    Pt2 ← getIntersectionPoint(v1, v3, selector)
    ▷ /*Pt1 and Pt2 are the points where selector cut the edges of the triangles*/
    if inCount == 1 then
        in.Add(Triangle(vertsin[0], Pt1, Pt2))
        out.Add(Triangle(vertsout[0], Pt1, Pt2))
        out.Add(Triangle(vertsout[0], vertsout[1], Pt2))
    else
        out.Add(Triangle(vertsout[0], Pt1, Pt2))
        in.Add(Triangle(vertsin[0], Pt1, Pt2))
        in.Add(Triangle(vertsin[0], vertsin[1], Pt2))
    return in, out
```

Cuboid

Creation - To create a cuboid in the VR environment we implemented an interaction technique that uses three points. The first two points (P_1) and (P_2) fix the corners of an initial rectangle (R) in 3D with a normal ($\overrightarrow{n_R}$) parallel to the floor plane (Figure 4.3a). Moving P_2 around will adjust the dimensions of the rectangle as well as its rotation over the y-axis. The final point (P_3) will define the depth of the cuboid with R as its base. Let ℓ be the line that goes through P_2 and is parallel to $\overrightarrow{n_R}$, and P_v as the controller's location. We will have $P_3 = proj_{\ell}\overrightarrow{P_v}$ which is the projection of P_v on ℓ. After creating the cuboid, it can be manipulated further to adjust its transformation matrix (Figure 4.3b).

Manipulation - In this context, we define a widget as a small sphere which can be grasped by pressing a button on the controller and can be moved around in the 3D space while the button remains pressed. For rotation and translation, we place a widget in the center of the cuboid that while pressed passes the rotation and translation information from the controller to the shape. Three more Widgets are placed at the defining points, P_1, P_2 and P_3 can be dragged to adjust the scale of the cuboid. Moving any of these widgets will fix the diagonal vertex of the cuboid in space and will scale the shape according to the position of the widget (see Figure 4.5a).



(a) Creating cuboid face.

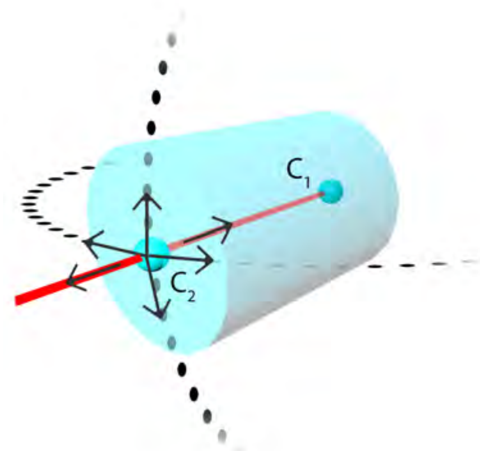
(b) Adjusting the depth using P_3.

Figure 4.3: Cuboid creation.

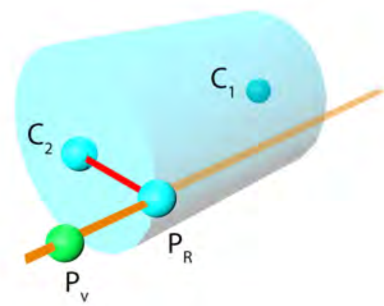
Cylinder

Creation - A cylinder is first created by defining a line with two points (C_1, C_2) which will be the orientation and height of the shape (Figure 4.4a). After fixing C_2, the controller's location will define P_R which is the closest point on the C_2 plane from the controller (Figure 4.4b). The radius of the cylinder is then calculated as $||C_2 - P_R||$. Similar to the cuboid, the transformation matrix of the cylinder can be altered after its creation.

Manipulation - A widget is placed in the center of the cylinder that maps the shape's rotation and translation to the rotation and translation of the controller. A second widget is placed at P_R which can be moved to alter the radius and height of the cylinder (see Figure 4.5b).

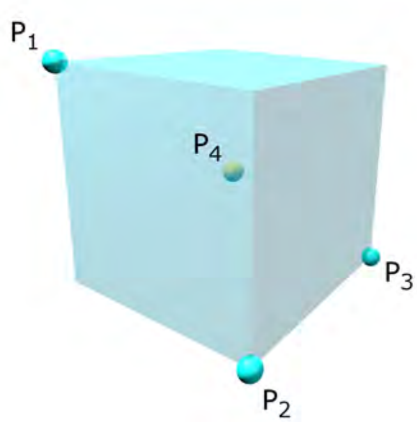


(a) Adjusting the orientation and height of the cylinder by moving C_2 in 3D space.

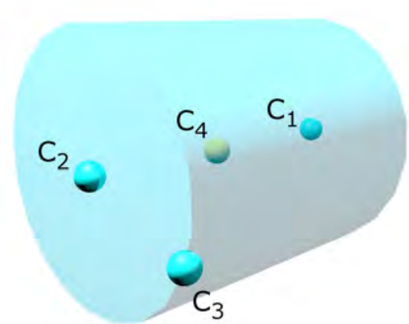


(b) Adjusting the radius using projection of P_v on C_2 plane.

Figure 4.4: Cylinder creation.



(a) P_4 allows cuboid translation and rotation. P_1, P_2 and P_3 scale the cuboid.



(b) C_4 allows cylinder translation and rotation. C_1, C_2 and C_3 allow scaling of the cylinder.

Figure 4.5: Manipulation widgets on primitives.

Extend to Any Geometric Shape

Editing Geometry Mode - The mesh cutter allows for any shaped geometry to be used for segmenting an object component. Participants begin with an initial primitive which can be modified by adding extra vertices to the mesh. As can be seen in Figure 4.6, on the left side of the image an extra vertex is placed in the cuboid edge by pressing and releasing the trigger button from the Vive controller on the desired position. In each vertex a widget is generated to manipulate the mesh morphology. These widgets can be dragged and the mesh changes according to the new widget position. Figure 4.6, shows how the newly created vertices on the right side of the image are translated upwards from the original position forming a semi-arc. While use of this feature was not a part of the study, it demonstrates how a starting primitive can be manipulated into a more complex shape.

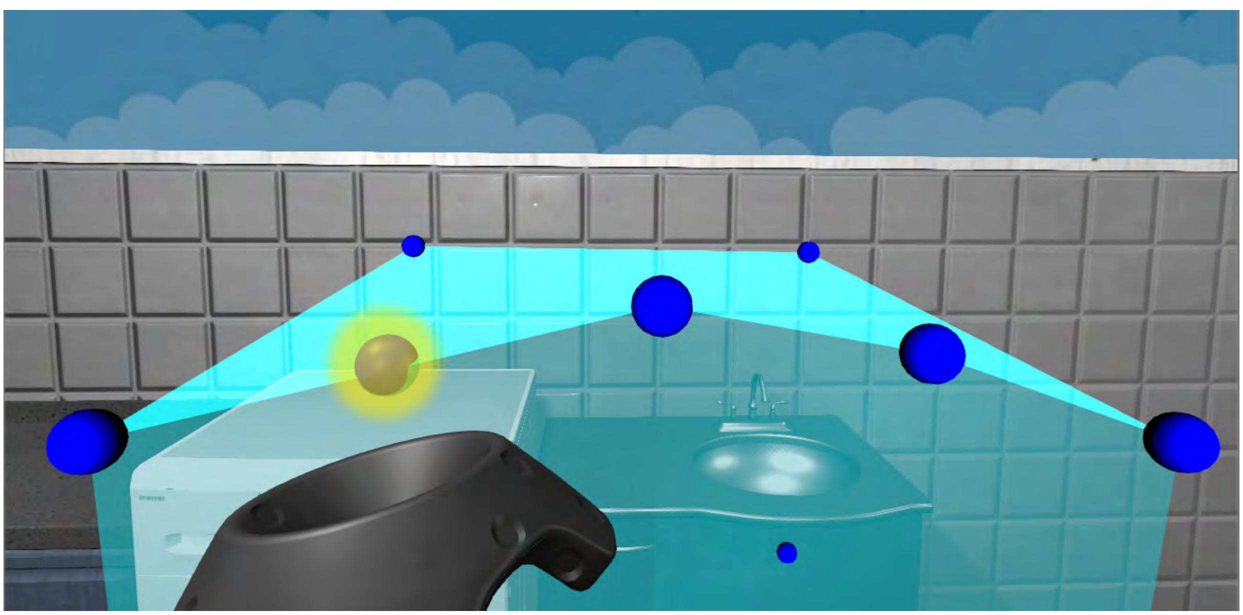


Figure 4.6: Highlighted in yellow a new vertex is created in editing geometry mode. Once positioned on an edge, affected faces are triangulated.

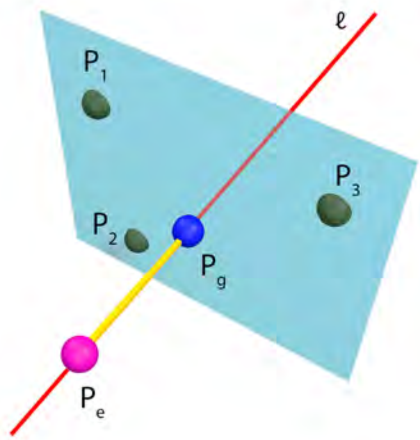
4.2.2 Interactions

An affordance we defined as part of the exploratory study was on constraining the movement of a component defined by the mesh cutter. An example of this is the key on a keyboard, which can only move in one direction (downward) and only for a fixed amount. Another example is a fridge door which can be rotated, but only around the axis of the hinge and within a specific range of angles.

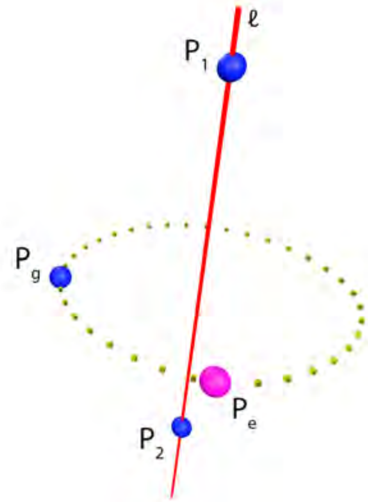
We created two tools to define the movement constraints for a component based on whether it is a perpendicular or rotational interaction. Additional interactions can be implemented but a more thorough analysis of the affordance concept is required as in the work by Baber et al in [5]. Such a study falls out of the scope of this work.

Perpendicular

A perpendicular interaction is the movement of an object in a straight line perpendicular to a plane (Figure 4.7a). This is first defined by creating three non-linear points which outline the plane perpendicular to the movement. For point placement, we cast a ray from the controller to the surface of the object. Next, a grasp point (P_g) is placed on the object. The system automatically defines ℓ as the orthogonal line from P_g to the plane. Finally, the user defines the interaction end point (P_e), where the grasp location will end up after the interaction. The projection point can be moved by moving the controller, but its location is calculated by the projection of the controller's location on ℓ. Once P_e is defined, the interaction is complete and the system will animate the object to demonstrate the newly defined behavior for the user.



(a) Perpendicular interaction.



(b) Rotation interaction.

Figure 4.7: Interactions.

Rotation

Rotation interactions are used for movements that are based on the rotation of an object around an axis (Figure 4.7b). To create this interaction the user will define the axis of rotation by placing two points (P_1 and P_2) creating a line that forms the axis of rotation (ℓ_a). Next, the grasp point (P_g) will be placed on the object to represent the location of effort (for example the door handle on a door). The final point is the end trajectory point (P_e), which shows the location that P_g will end up after rotating around ℓ_a. This is defined by calculating the complete rotation path of a circle starting at P_g and rotating around ℓ_a. Given the controller's location P_V, circle center P_c is calculated by $P_c = proj_{\ell_a}\overrightarrow{P_g}$. When the user presses the controller button, P_e is defined using:

$$
\begin{aligned}
\overrightarrow{v_\ell} &= proj_{\ell_a}\overrightarrow{P_v} - P_v \\
\overrightarrow{g_\ell} &= proj_{\ell_a}\overrightarrow{P_g} - P_g \\
P_e &= \overrightarrow{P_c} + ||\overrightarrow{g_\ell}||.\hat{v}_\ell
\end{aligned}
\tag{4.1}
$$

After fixing the location of P_e we will have a complete rotation interaction and the behavior can be animated as a demonstration to the user.

4.3 User Study

We performed an exploratory user study to understand the usability of AffordIt!. Post-participation surveys gathered qualitative information on usability, workload and perceived ease of use of the different aspects of the techniques. All participants used an HTC Vive Pro Eye for the study and started at the center position of a room with approximate dimensions of 4x4 meters. All virtual elements were conveniently placed so participants would not collide with real world elements during the study. We hypothesize that our tool will have high usability and low workload ratings.

4.3.1 Scenario and User Interface

The virtual scenario chosen for the experiment is a kitchen with different household appliances placed within the scene. We chose a kitchen environment so that any user can relate and have familiarity with the behavior of an appliance. Participants were allowed to interact with four objects in the scene: an oven, a washing machine, a storage cabinet and a coffee machine. Every combination of mesh cutter and affordance definition was performed on the objects. Figure 4.8, shows a side view of the physical area where the user study took place. The four virtual objects are super-imposed in the real room used for the study.

Figure 4.8: Side view of the 3D scanned area participants were allowed to walk. Virtual objects of interest for the study are positioned in the real world.

For the user interface, we used HTC Vive Controllers as the input device. The mesh cutters and the interactions to add affordances could be invoked from a menu (see Figure 4.9) attached to the left hand controller with the non-dominant hand. In the same controller, the track-pad button is used to show and hide the menu when pressed. For the controller on the dominant hand, a blue sphere is attached to the controller to be used as a custom pointer. The trigger button is equivalent to a "click" on a mouse and when pressed submits an action depending on the context. The gripper button when pressed executes an undo. The custom pointer is used to choose an option from the Menu as shown in Figure 4.9 by physically hovering the button and pressing the trigger. Once an option is selected the pointer is used to place the points required to perform the operations described in the previous section.

Figure 4.9: Menu with the different options to choose for participants.

4.3.2 Tasks

To complete the tasks participants were required to add behaviors to the objects in the scenario by invoking a mesh cutter tool (cuboid or cylinder) and define the behavior (perpendicular or rotation) of the segmented mesh.

Use a Mesh Cutter Tool to Define a Region of Interest

Participants were randomly assigned one object at a time. They decided which shape worked better to perform the object segmentation. After selecting the mesh cutter from the menu, participants approached the object and added the necessary points to create a cylinder or cube around the region of interest. If a mistake is done, the gripper button from the dominant hand controller would restart the procedure. After spawning the mesh cutter, users were allowed to transform the shape using widgets placed on the mesh geometry (see Section 3). Examples of cuboid and cylinder mesh cutters placed on objects are shown in Figure 4.1b and Figure 4.10, respectively.

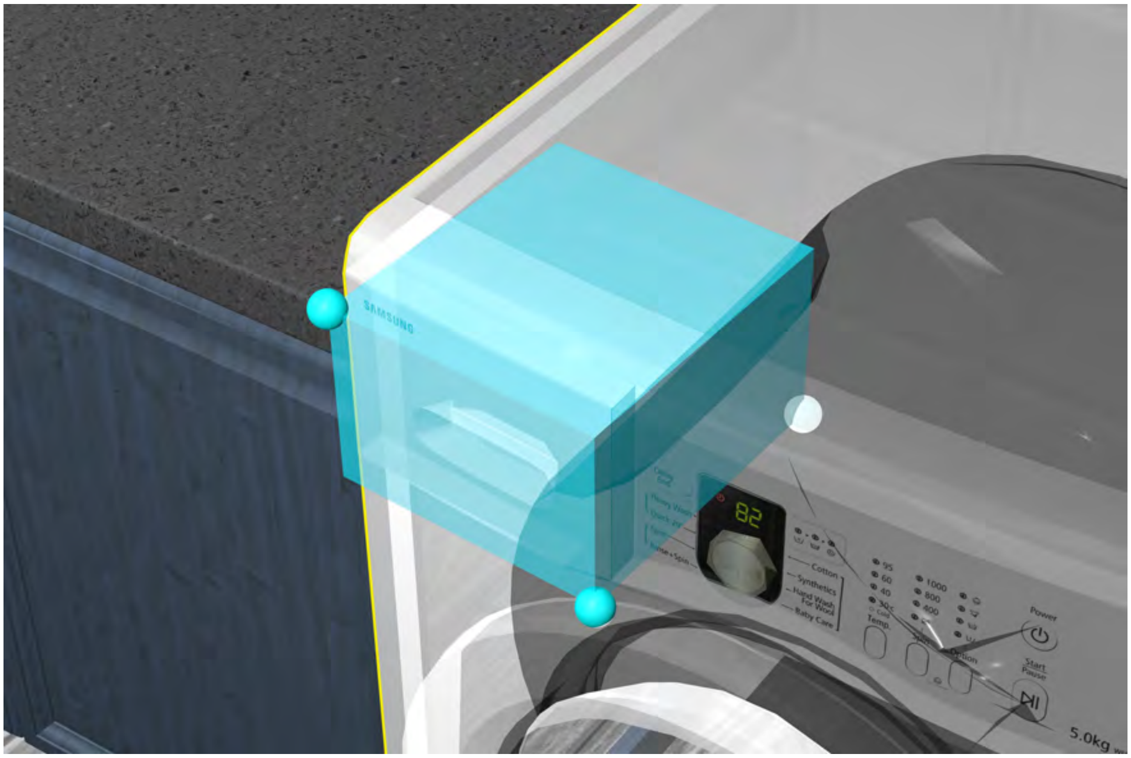


Figure 4.10: Cuboid mesh cutter placed on an object.

Add an Interaction to the segmented part

Next, users added an interaction to the selected region by placing points following the steps defined Section 3. For each step instructions are visualized as text in the menu to help participants remember which step they are performing. For the final point, widgets are spawned to visualize the object trajectory constrained to a path (See Figure 4.1c). For the perpendicular interaction, the path is linear and for the rotation it is circular. Users are allowed to undo one step at a time by pressing the gripper button. When the interaction is complete, the selected component will be separated from the original mesh and an animation shows the trajectory that the component is constrained to.

4.3.3 Participants and Apparatus

Sixteen people (10 male, 6 female) aged 18 to 29 ($\mu = 21.31, \sigma = 3.20$) engaged in the study. Participants were recruited from the University of Central Florida. Davis' Likert scale rating [23] from 1 to 7, (with 1 representing not experienced or not frequently and 7 representing very experienced or very frequent) was used to measure in a pre-questionnaire the following: VR experience ($\mu = 4.00, \sigma = 1.5$), user experience with modeling toolkits & game engines ($\mu = 2.88, \sigma = 1.27$) and how frequently they played video games ($\mu = 5.75, \sigma = 1.39$). To validate the usability of the proposed techniques a VR application was developed using an HTC Vive Pro Eye headset with a resolution of 1600x1400 per eye and a field of view of 110 degrees. Two controllers were used for bi-manual interaction. Headset and controllers were tracked by HTC lighthouses. The application was implemented in Unity3D game engine using C# and SteamVR. The experiment ran on a desktop computer with an Intel Processor Core i7-8700K CPU 3.70GHz, 32 Gb RAM and a Nvidia GTX 1080Ti graphics card.

4.3.4 Study Design and Procedure

Our exploratory study was designed to be completed in approximately 45 minutes. Study participants were asked to fill out demographics and pre-questionnaire forms. Next, the problem was explained for 2 minutes followed by a 5 minute video tutorial session, which allowed participants to familiarize themselves with the concepts and user interface. This was followed by a training session which was performed for an additional 5 minutes. The training session required participants to use the tools of AffordIt! following proctor instructions. An example object in the form of a modular sink with three drawers and two doors was used for training. For the experiment, participants were randomly assigned 4 different objects from the scene in Figure 4.11 to perform selection cuts in the objects' mesh and assign affordances to the component generated. After task

Table 4.1: Post Questionnaire. Participants answered these questions on a 7 point Likert scale (1 = Very Little or Totally Disagree, 7 = A lot or Totally Agree).

#	Question
Q1	How much did the WEIGHT of the headset affected you?
Q2	How ACCURATE the HTC-Vive controllers felt?
Q3	How much did the PHYSICAL BUTTONS on the HTC-Vive helped with the overall experience?
Q4	How much did the VIRTUAL BUTTONS on the left-hand MENU helped with the overall experience?
Q5	How easy was to perform a selection of a region of interest from an object using a CUBE shape?
Q6	How easy was to perform a selection of a region of interest from an object using a CYLINDER shape?
Q7	How easy was to perform a ROTATION affordance around a hinge?
Q8	How easy was to perform a PERPENDICULAR to a plane affordance?
Q9	I enjoyed using the system overall.
Q10	The objects and assets in the scenario seemed realistic.

completion a post-questionnaire (see Table 4.1) with a Likert Scale [23] from 1 (Very Little or Totally Disagree) to 7 (A lot or Totally Agree), was provided to the participant. In addition, a SUS [12] questionnaire for perceived usability of the tool and a NASA TLX questionnaire [45] for perceived workload was given to participants. Finally, participants were asked about their overall experience and any thoughts or suggestions they could have about the interface.

4.4 Results

All participants were able to complete every task. Surveys provided to participants gathered qualitative data (Table 4.1) which results are shown in Figure 4.12. The purpose of this analysis is to identify users' scores on each individual aspect of the system, how much workload was perceived, how usable were the techniques and observations that can bring insights on future directions.

Figure 4.11: User study virtual environment setup.

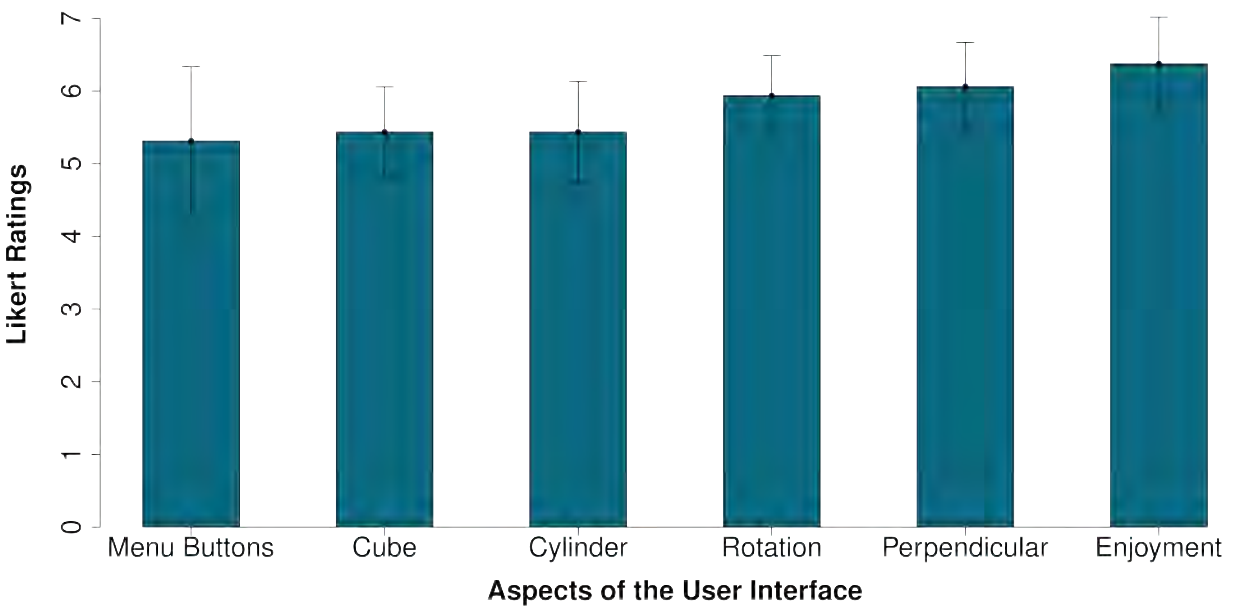


Figure 4.12: Plot shows the mean values and standard errors for each one of the aspects of the interface.

4.4.1 Usability and Perception

The user interface involved the use of menu buttons fixed to the left controller and placing points to define four different operations. These aspects of the interface (Q4, Q5, Q6, Q7, Q8, Q9) asked in Table 5.1 were rated by participants and results are shown in Figure 4.12. For overall usability, results from SUS scores ($\mu = 83.10, \sigma = 12.9$) show high usability for the user interface. Additionally, aspects of the hardware, such as the weight of the headset, causing issues had a low rating (Q1) ($\mu = 2.44, \sigma = 1.46$), accuracy had a high rating (Q2) ($\mu = 6.00, \sigma = 0.94$) and buttons from the controller ($\mu = 6.25, \sigma = 1.03$) were well received by participants. We conclude that these variables did not influence the correctness of the experiment. Finally, we saw a high rating for the perception of realism in the environment (Q10) ($\mu = 5.88, \sigma = 0.78$).

4.4.2 Workload

Figure 4.13, shows scores for each subscale of an unweighted (raw) NASA TLX. A raw TLX is preferred for this study since no difference has been found in sensitivity when compared to the full version [44]. The overall subjective workload score per participant is ($\mu = 37.35, \sigma = 12.22$), which shows a low workload perception. The six factors of the NASA TLX include: Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Own Performance (OP), Effort (EF), and Frustration Level (FL).

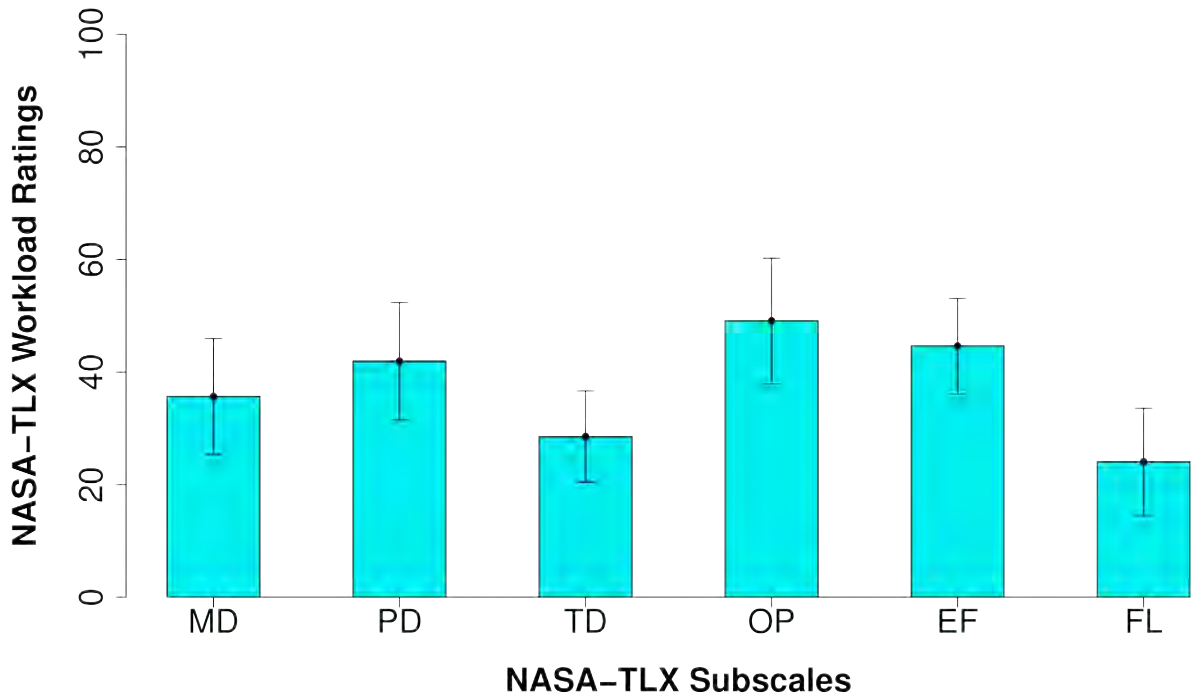


Figure 4.13: Plot shows the mean values and standard errors for NASA TLX workload ratings.

4.4.3 Implications

This chapter evaluates the usability of AffordIt! as a tool to create behaviors in objects' components. In line with work by Hayatpur et al. [46] and Shao et al. [130], the creation and manipulation of primitives resulted in an intuitive task for participants as shown in the results. An aspect not evaluated by this work nor explored in previous work is how to extend such primitives to adapt to specific shapes that could be found in a real world scenario. This work suggests to **create or generate primitives for the purpose of selecting and segmenting mesh components**. Evaluation of how such primitives can be adjusted to specific shapes is left for future work.

The interactions presented in this work were perceived as highly usable as results shown. However, based on work by Hayatpur et al. [46] and participants' comments in our study, it is suggested that **constrained movements should be authored in real-time.** This means, real-time visualization of the outcome while authoring the interaction.

Finally, the use of interactions can be extended to support more complex behaviors. In Deering, [24] animation editing is conceived through components called elemental-animation objects. Following this principle, this work suggests to **implement interactions that can be easily extendable by combining them or attaching them to one or more objects.**

4.5 Discussion and Observations

Our exploratory study was successful in offering us several points of feedback which are discussed below.

4.5.1 Usability and Workload Analysis

In our SUS and TLX analysis we found users to rate AffordIt! as having high usability and low perceived workload. This tells us that even this initial iteration has value in its use for affordance assignment to the components of an object. We were concerned that the virtual environment would be perceived as difficult, but the low workload rating from the TLX score assures us that users did not perceive themselves to be under a strenuous activity.

4.5.2 Post-Questionnaire Analysis

The Likert scale results from the post questionnaire provide us with additional feedback about how users felt toward the system. The low score for the headset weight (Q1) and the high score for the accuracy of the controllers (Q2) show us that the use of the HTC Vive Pro Eye did not have a negative impact on the user experience. Users found that they liked the virtual buttons (Q3) and the physical buttons (Q4). For the assigned tasks they found the creation of the cube and cylinders to be easy (Q5, Q6) and the assignment of the movement constraints to also be easy (Q7, Q8). Overall users enjoyed the system (Q9) and they found the objects and assets within the scenario to be realistic (Q10), suggesting high immersion within the scene.

4.5.3 Comment Observations

While all participants were able to create the shapes for selection and the interactions to define behaviors we found their suggestions intriguing and an avenue for opportunities for improvement.

Bring objects to the users rather than users to the objects

The study was conceived as an immersive authoring experience so the size of objects and the placement of objects within the environment replicate a real life scenario. A participant mentioned that they would prefer objects floating in the air to avoid bending to interact. We note that this is a valid point for a full VR authoring tool like in Hayatpur et al. [46].

> User 7: *"Sometimes I had to move my body a lot, like squating, to reach an object."*

Visual aid guidance on movement path while editing

Another intriguing set of comments was a user stating they had a good experience because of the thinking process involved while another participant did not like the outcome because of misplaced rotation points. We believe that more visual aid in the form of animations showing the movement path can help ease the thinking process of participants.

> User 2: *"I liked how the experiment made me think about how objects move."*

> User 7: *"I liked how accurate the movements were represented in VR. I disliked how sometimes the rotation points did not come out how I expected them to."*

Depth perception

Depth was perceived differently among participants with the use of transparency while authoring the object behavior affected user perception of depth in some cases. A possible solution is to allow toggle transparency depending on user needs. Also outlining the edges of the shape was suggested by a participant.

> User 9: *"Making the meshes transparent helps with setting the location of the cylinder/box, however it makes some interactions with the object such as adding hinges difficult."*

> User 14: *"I liked how easy affecting objects was. I'd suggest making the textures not so transparent or emphasizing the outlines of the cube and cylinder shapes."*

Possible applications

Participants also suggested a possible use-case of AffordIt! in the following areas: game design, building interior design, education, 3D modeling programs and animations.

User 3: *“useful for game design for the object interaction without coding ”*

User 7: *“It can be used for designing interiors or developing accurate gaming scenes with accurate animations. ”*

User 12: *“I think this can be useful for 3D modeling programs using VR, and for video game interactions.”*

User 13: *“creating a situation before actually building the real thing in irl (in real life)”*

4.6 Limitations

This study is exploratory in nature, and to the best of our knowledge, there is no tool available for comparison at the moment. A possible baseline condition could be 3D modelers on the desktop such as Maya or Blender but the number of features and complexity would not provide a fair comparison. This work acknowledges limitations on AffordIt!, which leaves room for future improvements. The study is designed as a human-in-the-loop approach, therefore inheriting intuition from the users is expected to accomplish the tasks. Ideally, an autonomous technique could be designed in which the object's geometry is analysed, a mesh cutter is designed, and an affordance is applied. However, we believe that the intuitive understanding of the user should be included within the process.

Some meshes contain no internal faces, exposing a hole once the affordance is applied. We could advance our mesh cutting algorithm to also extrapolate face and normal data to the newly exposed sections of the mesh.

Also, we can develop interactions similar to [133] such that a user can draw the region of interest, snapping points to the most likely portion of the object, rather than relying on pre-defined selection shapes. This could provide increased accuracy and remove human error. As one user commented:

> User 5: *"Snapping surfaces of the mesh cutter to parallel surfaces of the object of interest"*

Finally, in order to provide a direct comparison to 3D modeling software, as future work we would like to conduct a larger study that seeks out modeling software experts to compare AffordIt! with traditional modeling software tools on a desktop environment. Likewise, an additional baseline condition in a desktop environment following the same principles could be implemented for direct comparison with AffordIt!.

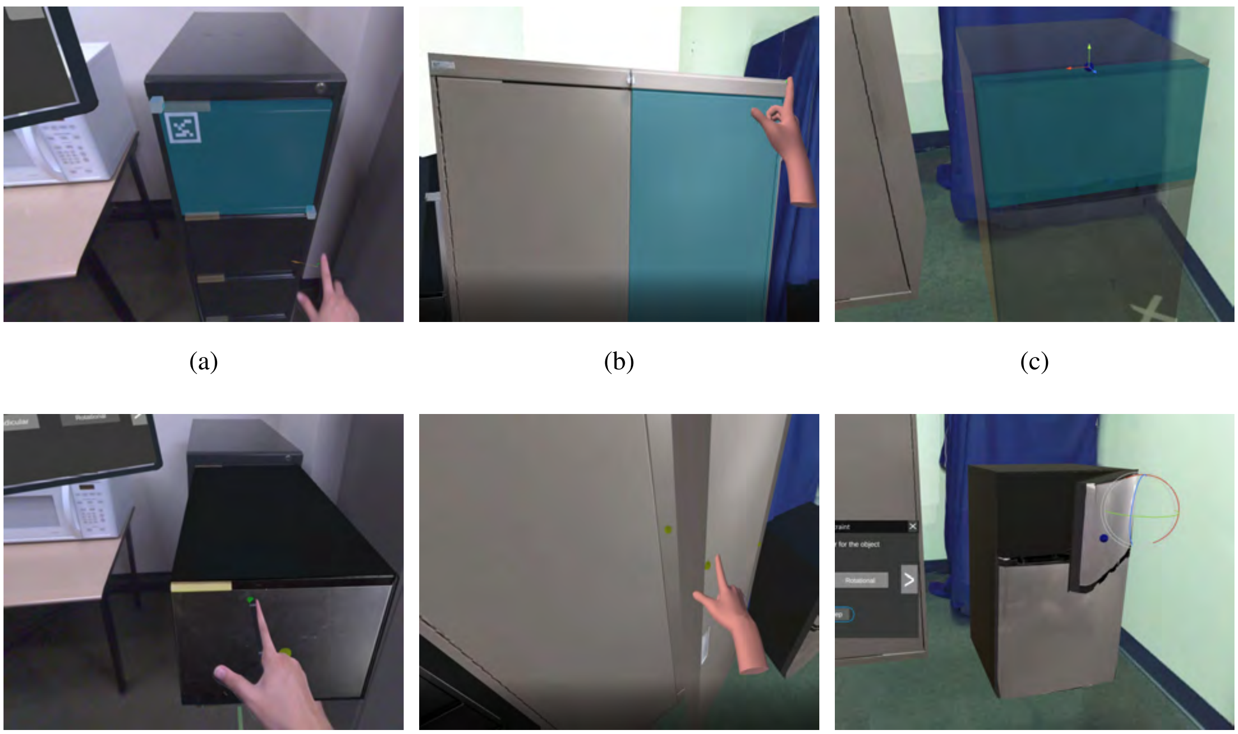
4.7 Conclusion

This chapter introduces AffordIt!, a set of techniques that author object components behaviors in Virtual Reality. Despite the limitations and observations found, usability results show that the interface and interaction techniques were well received by participants, as seen in the high usability scores for SUS, and had a low workload for the tasks, as shown in the low scores for TLX. Participants' comments showed that they enjoyed the experience. Furthermore, the affordance techniques scored higher than the mesh cutters which can be improved, as discussed in our future work section.

There is work to be done in refining AffordIt!, but we have shown that even our initial iteration allows 3D scene authors to intuitively segment and assign affordances to an object either for scene authoring or in the development of 3D assets for a variety of use cases.

CHAPTER 5: AUTHORING REAL AND VIRTUAL OBJECT COMPONENTS IN AR, VR AND THE DESKTOP

5.1 Introduction



(a) (b) (c)

(d) (e) (f)

Figure 5.1: Authoring tool across three different conditions. Top row (a), (b) and (c) show the component selection authoring. Bottom row (d), (e) and (f) component behavior authored for the three objects.

Encouraged by the results of the AffordIt! exploratory study, this chapter presents an evaluation of the authoring technique with real and virtual objects. In line with the requirements of this thesis to take advantage of the benefits of situated authoring, we first need to analyze this type of authoring from an in-situ approach. We also evaluate the technique against the Desktop and VR environments.

In the development of domain-specific training experiences, there is a requirement for objects to behave as expected for that domain. This creates a need for an expert that can aid with content generation and implementation for the training simulation. It is vital for this content to serve as a digital twin of the real object as it can drastically reduce costs, improve safety or provide a self-contained environment for a trainee. This type of training is grounded in principles of Situated Learning [60], with the objective to exploit the surrounding visual cues as means to provide a near to real-world experience. In a given scenario, virtual objects possess attributes and behaviors that can infer some kind of instruction in the scene such as in [35, 116] in the form of audio and visual cues. Conversely, some objects demand more complex actions such as rotating a knob or pressing a button as in [54] or opening doors or compartments as the examples in [93] or even deforming geometries of objects like the spring scenario from [148].

However, these experts are often not familiar with content generation tools. As such, they have difficulty authoring a representative simulation with ease. This is further complicated as the means of interaction used are diverse across conditions like Desktop [35, 84, 85, 127], Virtual Reality (VR) [25, 46, 116, 153], Augmented Reality [35, 72, 81, 104, 148] and even combined approaches as in [49] using Desktop and VR together.

In this chapter we analyze solutions to this problem by evaluating affordance frameworks across multiple conditions: AR, VR and desktop. To do this we begin with Kallmann et al. [65–68] which provides an organizational framework called smart objects based on actor-object interactions. This conceptual model identifies different ”interaction features” based on the type of data each contains. Our interest based on this definition is specific to parts, actions and commands. Interactions between users and objects have been captured using computer vision to identify objects and users’ demonstrations to record actions as in the work by Chidambaram et al. [20]. However, these types of interactions do not capture the intrinsic properties of the objects or what a specific object affords to do as in GesturAR [148] and AffordIt!, [93]. We believe providing a user a tool to author and record object properties and actions is more suited towards defining object components behaviors. While AffordIt! in Masnadi et al. [93] offers a solution to assign affordances and constraints to the intrinsic behaviors of an individual object, it has not explored how users perceive or perform in a human in the loop approach.

In our research we couple these ideas with the emerging technology of portable light detection and ranging (LIDAR) systems which can streamline the capture of the geometry of everyday objects. This is highly valuable given that assets can be generated directly from the training environment as opposed to being designed by a 3D artist. A 3D reconstructed object is created as a single continuous mesh which then can be segmented into meaningful parts using artificial intelligence as shown in George et al. [31] and Shamir et al.'s survey [129] or human in the loop approaches as seen in [146] and [105]. Ipsita et al. in VRFromX [53] provide a framework for turning real world scans into interactive virtual environments. In a similar fashion GesturAR in [148] captures geometry from real objects to then map gesture inputs to AR content behavior. ScalAR in [116] focuses on the semantics of the layout of the objects in a scene rather than interactions with part of the objects.

In this chapter, we present an evaluation across three different interface conditions of the concepts defined in AffordIt! [93]. We seek to identify advantages and challenges of object behavior authoring tools whether in-situ using AR/VR or computer aided using traditional interaction techniques. AffordIt! interactions were custom tailored for each interface condition, using mouse and keyboard for Desktop and hand tracking and gestures for AR and VR. Fiducials are used in AR and VR. In AR for improving precision in the mapping of real objects with the respective digital twins. In VR to align always the virtual environment to the same physical position in the physical room. The results of this study present users' preferences, usability and performance while authoring object component behaviors in each interface.

Our contributions in this chapter are:

1. First comparison of AR, VR and Desktop for authoring object component behaviors.
2. A user study to evaluate usability and user preference from participants.

5.2 System Design

Two different sets of hardware devices were used in this experiment as can be seen in Figure 5.2. For the desktop condition, we use a 55-inch flat-panel TV display connected to a PC with a dedicated graphics card. For AR and VR, we use the Varjo XR-3 HMD coupled with a pair of SteamVR Base Stations 2.0 for positional tracking connected to the same PC. A Vive controller is used for configuration by the proctor when aligning to anchored positions. For audio cues, a JBL Tune 510BT Wireless Bluetooth On-Ear Headphones was used in all conditions.

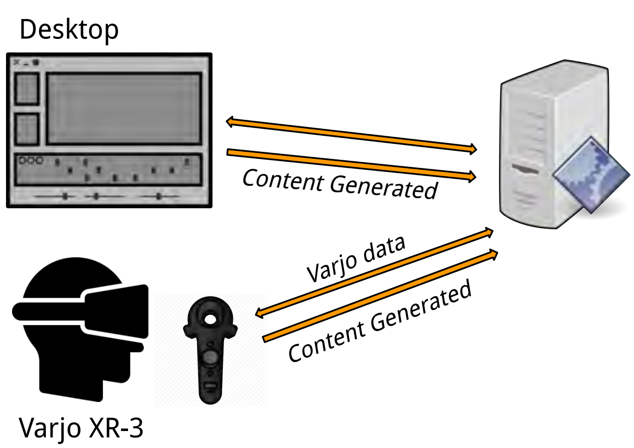


Figure 5.2: System Architecture with a Varjo XR-3 used for AR and VR conditions.

Three systems were developed implementing the techniques presented in [93]. The applications were built using Unity3D [144] version 2020.3.30f1. While interface and interaction techniques are different across the conditions the functionality remains the same. Authoring systems have four different aspects (1) select a region of interest (ROI) as a component of a virtual object, (2) attach a perpendicular pull behavior to the ROI, (3) attach a rotation pull behavior to the ROI, (4) visualize the result. Real objects in AR and virtual objects in VR and Desktop can be interacted with through collision with a pointer which spawns a step-by-step menu (see Figure 5.4). The pointer in the Desktop condition is the mouse, and for AR and VR it is a green sphere placed in the index fingers of participants as can be seen in Figure 5.3. The pointer triggers events by collision with the UI or objects in the scene. Similarly, an undo option is provided in the UI which allows participants to easily recover from mistakes. The goal of the study is to do a fair comparison, as such users use the same authoring techniques across different conditions and environments. Both Desktop and VR environments have been captured with LIDAR to simulate the real environment.

Figure 5.3: Left, hands visualized in AR. Right, virtual hands in VR. Green sphere are pointers to interact with elements in the scene.

5.2.1 UI Design Considerations

The system interaction is designed to be object centered, each virtual object when selected spawns a menu (see Figure 5.4) that lets the user author the different aspects of the 3D object. Options from the menu and undo action is then contextual to the object being interacted with. Positioning in the wrist is an accepted strategy explored in the work by Li et al. in [82]. Visual guidance is given from the menu using a light blue ring blinking around the possible next button to press if the user wishes to continue authoring.

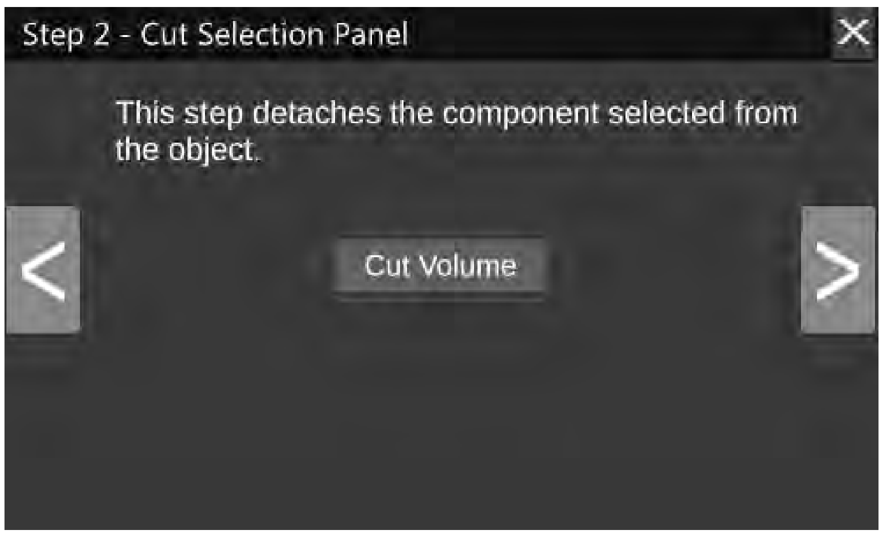


Figure 5.4: Step by step menu used across all conditions.

AR/VR Depth perception

When designing the interface, the perception of depth in AR and VR was analyzed. It was noted in informal pilot studies that participants perceived depth differently in some cases. Distance judgement is extensively explored in research by Masnadi et al. [92] and Pfeil et al. [113] which implies field of view as a possible factor. In our study interactions happen at short distances with the HMD selected for the study having a horizontal FoV of 115°. The Inter-Pupillary Distance (IPD) adjustment was configured using Varjo proprietary software. To mitigate this problem, additional audio and visual cues were provided for participants to better perceive depth. A ray is visible from the index finger to the virtual object, when the pointer is less than 10 cm away to the 3D object or the menu (see Figure 5.5).

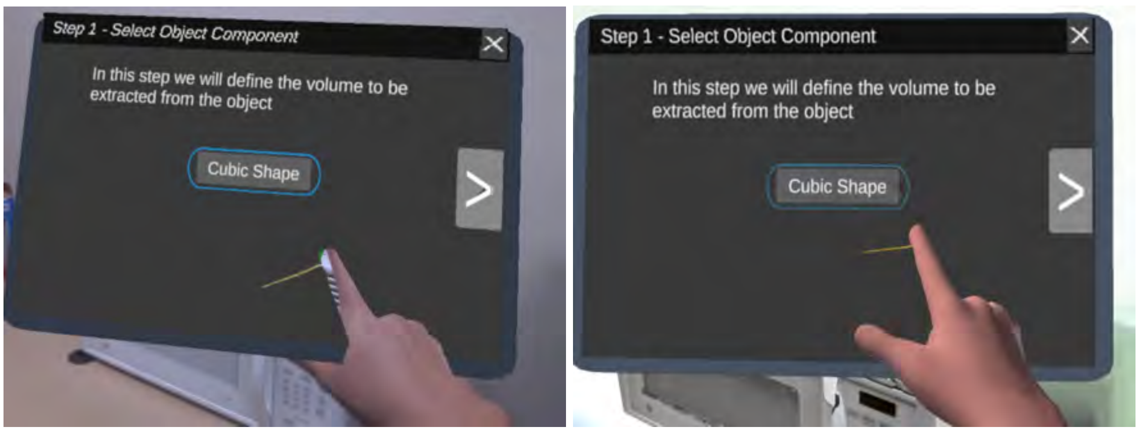


Figure 5.5: A ray between the index finger and the menu can be seen in both images left AR and right VR.

AR/VR Drag and Submit Gestures

The menu is draggable in the three conditions. In Desktop this is done by left clicking and holding the title bar while moving the mouse. For AR and VR the menu is draggable by using the pinch gesture to the frame surrounding the 3D canvas, which then enables the menu to follow the hand transform until the release gesture is performed [96]. A custom gesture is used as well to submit the component selection and to submit the authoring of a behavior. The gesture is invoked by closing the thumb finger as a clenched fist as can be seen in Figure 5.6.

Figure 5.6: Custom gesture to submit a completed component selection from the drawer.

Component Selection Cuboid manipulation

Once the component selection is completed transform tools to scale or translate the cuboid are enabled. In Desktop traditional transform gizmos were implemented in the interface (e.g. see left image in Figure 5.7). For AR and VR 3D widgets in the form of a cube for scaling and a cone for translating were added. These widgets can be invoked from the main menu and when dragged using a pinch gesture, scale or translate the cuboid geometry in the direction of the movement as can be seen in the right image in Figure 5.7.

Figure 5.7: Left, translate gizmo in Desktop interface. Right, scale widget being manipulated.

5.2.2 AR Application

The AR condition is a video see-through application deployed for a mixed reality head mounted display (HMD). To provide a better sense of depth, an occlusion material was used for the virtual objects and hands. Markers were added to the physical objects to aid mapping between the virtual and real objects. For hand tracking the Ultraleap [143] package was integrated into the development pipeline. HMD tracking was possible with the integration of SteamVR 2.0 Lighthouses placed on the extreme ends of the physical room.

5.2.3 VR Application

The VR condition is also deployed to a mixed reality HMD. The virtual objects' materials in this case are textures generated from pictures taken from real objects and mapped to the 3D scanned geometries. A marker is used to align the objects' digital twins right in front of their physical counterpart versions. This was purposely designed to allow participants to safely walk the environment while immersed in the VR scenario. Ultraleap [143] was integrated in this version too, to handle hand tracking. HMD tracking was also provided by the SteamVR 2.0 Lighthouses.

5.2.4 Desktop Application

The Desktop application was implemented using the traditional graphical user interface (GUI) paradigm. The tool follows similar characteristics to popular 3D editor tools. The user interface adheres to the following design principles: graphic layouts, task-related grouping and direct manipulation [2, 131]. The main menu follows the graphic design by Fluent from Microsoft [98]. The interaction is performed by mouse and keyboard. The behaviors authored are equivalent to the ones produced by the AR counterpart.

Figure 5.8: Desktop interface, in the figure the cabinet is selected with a left click. The draggable object menu and undo button are visible.

5.3 User Study

A comparative study on three different interface conditions based on concepts in AffordIt! [93] was performed to identify preference and differences in usability and performance across conditions. We conducted a within-subjects design with one factor and three levels, the conditions were authoring tools deployed in 1) Augmented Reality, 2) Virtual Reality, and 3) Desktop. The order of each condition and the order of objects chosen in the experimental task were randomized to account for order effects. Qualitative data was collected using post-condition surveys after completing the tasks in each condition. A post-experience questionnaire was filled in by participants at the end of the study to gather their interface of preference for four aspects of the system. Post-condition surveys involved the use of custom (see Table 5.1) and standard questionnaires NASA-TLX [45] to measure workload and System Usability Scale (SUS) [12] to measure usability. Quantitative data was saved from participants' interactions with menus and objects during each condition. Time error rates and frequency of participants undoing an action are recorded using scripts in the applications. Headset and hand-tracking information were recorded as well.

Table 5.1: Post-Condition Questionnaire. Participants answered these questions on a 7 point Likert scale (1 = Not much, 7 = A lot). *Not present in Desktop condition.

#	Question
Q1	Rate the importance of the task using current technology
Q2	Rate your prior experience using the technology in this study
Q3	How realistic did you find the virtual objects in the scene?
*Q4	How much did the weight of the headset affected you?
*Q5	How accurate the hand tracking felt?
Q6	How easy was to press buttons in the UI?
Q7	How easy was to define the region for object component selection?
Q8	How easy was to add a rotation behavior to the selected part of the object? a) Mini-fridge, b) cabinet
Q9	How easy was to add a perpendicular pull behavior to the selected part of the object?
Q10	I enjoyed using the system overall.

5.3.1 Participants and Apparatus

Twenty one people (10 male, 9 female, 1 Non-binary, 1 Preferred not to say) aged 18 to 43 ($\mu = 26.43, \sigma = 6.28$) participated in our study. Participants were recruited from a university population from a variety of majors such as Computer Science, Electrical Engineering, Industrial Engineering, Biomedical Science, etc. All participants were right handed. Davis' Likert scale ratings [23] from 1 to 7 with 1 representing "little experience" and 7 "very experienced" was used to measure the following: overall expertise using computers ($\mu = 6.14, \sigma = 1.06$), participants experience in VR ($\mu = 4.00, \sigma = 1.67$) and experience with AR ($\mu = 3.48, \sigma = 1.72$).

The experiment was either conducted on a PC (Core i7-11700 CPU, Nvidia RTX 3080Ti graphics card, 32 GB RAM) via a 55-inch flat-panel TV display, or on the Varjo XR-3 HMD connected wired to the same PC. The specifications for the HMD are as follow: horizontal field of view of 115°. Ultra-low latency, dual 12-megapixel video pass-through at 90 Hz. Headset weight 594 g + headband 386g. From the Varjo Base manager configuration settings application, foveated rendering was disabled, the resolution quality was set to High (default) - 35PPD and Simple rendering was enabled. Foveated rendering was disabled to achieve a similar framerate for the AR and VR conditions, and also to avoid an extra eye-tracking calibration. The interpupillary distance was adjusted automatically for each participant.

5.3.2 Study Preparation

The room the study took place in, was 3D scanned using a FARO ultra-portable Focus Laser Scanner with settings to the highest resolution. Four objects were scanned using an Artec Eva 3D portable scanner. Three objects and the room background can be seen in Figure 5.9. The additional object scanned was a microwave which was used for training participants in the UI. Meshes were generated from the pointcloud captured. The virtual environment is an exact replica of the real physical room. The virtual environment was used for Desktop and VR conditions. For the AR condition, the virtual object meshes were used as occlusion material. Also, fiducials were taped to the real objects for increased precision when locating them with their digital twins. The rationale behind this decision was to better align the physical and virtual objects and also to recover tracking in case of drifting. For VR an additional fiducial was taped to a power plug on the wall which was used for aligning the VR environment and to maximize walking space for the participants inside the same room.

Figure 5.9: Left, a picture of the room. Right, a screenshot of the virtual room from a similar vantage point as in the left image.

5.3.3 Tasks

The following tasks were required to be completed by participants for each of the virtual and real objects in the scene. The tasks involve segmenting components of a virtual object to then author behaviors on these new parts [93]. Participants performed these tasks using the three interface conditions.

Object component selection

The objective of a component selection is to surround the region of interest with a 3D primitive which will define the part of the virtual object to author behaviors from. For this study, a cubic shape is the primitive chosen due to the geometry of the objects. A cubic shape is defined by three way-points, the first two positioned on the opposite corners of the object component forming a rectangle. A third point completes the cuboid shape in the inside direction of the object as can be seen in the left image from Figure 5.10. For AR and VR the points are placed in the position the index fingertip collides with the virtual object. For the Desktop the first two points are positioned by left clicking on top of the object with the mouse. After, the third point will be visible and can be dragged using a translation gizmo. The submit gesture completes the step in AR and VR. On the Desktop left clicking the button "Complete step" from the menu and end the step. The next step is to click "Cut Volume" from the main menu, which creates a new object component instance.

Figure 5.10: Left, a user defining the bounds of the drawer component in AR. The Center shows the first point added to the cabinet in VR. Right image shows the door component selected for the mini-fridge in the Desktop.

Perpendicular pulling behavior

This behavior is generated by placing 5 points on the object component. The first three points define a plane and participants were asked to place the points in a non-linear way. The fourth and fifth points decide the perpendicular movement constraint (see left image in Figure 5.11. In AR and VR the result is immediately visualized in 3D (see Figure 4.1a) and the final fifth point is decided with a submit gesture. In Desktop the fifth point can be dragged with a translation gizmo to then ”Complete Step” from the main menu.

Rotational behavior

A rotation is defined by four points. The first two points establish the rotation axis, and the third point is placed at the position they would grab the component from to open it. The result is shown immediately with the component opening based on the hand movement. In AR and VR, the step is completed by performing a submit gesture at the angle the component is opened, this positions the fourth point too. In Desktop a rotation gizmo is spawned around the pivoting axes to rotate the component as desired. The final position is recorded upon left clicking “Complete Step” in main menu.

Figure 5.11: The left image shows the last point added to define a perpendicular behavior in AR. The right image shows in yellow the points to enable the rotation of the cabinet door in VR.

Visualize result

Once a behavior is authored participants could select from the main menu the option to ”Animate” the behavior. A user can then visualize how the component is constrained to move based on the points added to it. The animation can be stopped as well, so this to not cause an additional distraction while authoring behaviors in a different object.

5.3.4 Procedure

The study was designed to take around 90 minutes. Each participant was guided to the study room and while seated in front of a computer a consent form was handed over explaining the experiment procedure. Upon agreeing to participate the participant was asked to fill in a demographics and prior experience questionnaire. The problem was then explained for about five minutes to then proceed to the randomly selected condition to start the study. A video tutorial of approximately three minutes before starting each condition was provided to familiarize the participant with the concepts and user interface. Equipment used for the study (head mounted display and headphones for AR/VR just headphones for Desktop) is then handed over to the participant.

For the VR condition, a participant is asked to hold the HMD in front of a tag placed on the wall of the room. The proctor then presses the primary button on the controller twice. This is to align the virtual scene always to the same position for all participants. The HMD and headphones can then be worn.

For the AR condition, participants wear the HMD and headphones first and then before interacting with each object the participant is asked to look for some seconds at the marker on the physical object. In Figure 5.12 the left image shows a green square which is invoked by the proctor pressing the primary button to start alignment, and a blue square is shown when the object alignment is completed by the proctor with a second primary button press. This is to align the digital twin object to the physical one.

Figure 5.12: Left image shows feedback for alignment start, middle and right image show feedback for alignment complete.

Each intervention is expected to last from 10-15 minutes to minimize the simulator sickness risk. The study starts with a training session involving a task similar to the ones performed in the study. For AR and VR during training the user is asked to perform the thumb gesture 5 times each one spawns a green cube for feedback, to familiarize the participant with this interaction. For Desktop instead, participants are asked to familiarize themselves with the camera manager interface. This interface was chosen as it is the standard across 3D editing tools e.g. Unity3D editor. The rest follows user interface design principles such as: graphic layouts, metaphors, and direct manipulation [2, 131]. The study then starts and the user assigns behaviors to each one of the objects in the scene in random order. The screen session is video recorded. A post-condition questionnaire is then filled in by the participant to evaluate the interface assigned at the end of the session. Once all conditions are completed, a post-experience questionnaire is provided, to gather preference information and thoughts from the conditions experienced.

5.3.5 Covid

Each participant was required to follow the protocols and guidelines by the CDC (Centers for Disease Control) related to COVID19 measures to avoid virus spread. To reduce risk, the proctor kept socially distancing and provided instruction from a position at least 6ft away from the participant. The devices were thoroughly cleaned and disinfected to reduce the risk of COVID19 transmission.

5.4 Results

Quantitative and Qualitative data were gathered and analyzed in the following sections. For quantitative data, time and error rates were collected with scripts during task completion. For qualitative data, standard and custom questionnaires 5.1 were filled in by participants based on the user experience on each condition. All participants successfully completed all tasks assigned by the proctor. The goal of the following statistical analysis is to determine differences in usability and performance between participants solving the tasks assigned across the different conditions.

5.4.1 Time and Error rates

The time spent by participants while performing the tasks assigned was calculated per condition from the moment participants pressed the button on the menu to start the object component selection until the moment they visualize the result by pressing the ”Animate” button. This is for each of the three objects interacted with. Total time distribution is shown in Figure 5.13 for each condition in the experiment. A Shapiro-Wilk test on each condition times, showed that Desktop and VR were not normally distributed as can be seen in Figure 5.14 with the quantile-quantile plot. Therefore a Friedman's test was used and revealed no significant differences in the times spent by participants to complete the tasks in AR, VR and Desktop $\chi^2(2) = 5.81, p = .055$. A post-hoc analysis using Wilcoxon signed-rank test for pairs revealed that participants spent significantly more time in AR than Desktop ($\boldsymbol{Z = -2.53, p < .05}$) to complete their tasks. No difference was found between the other two pairs.

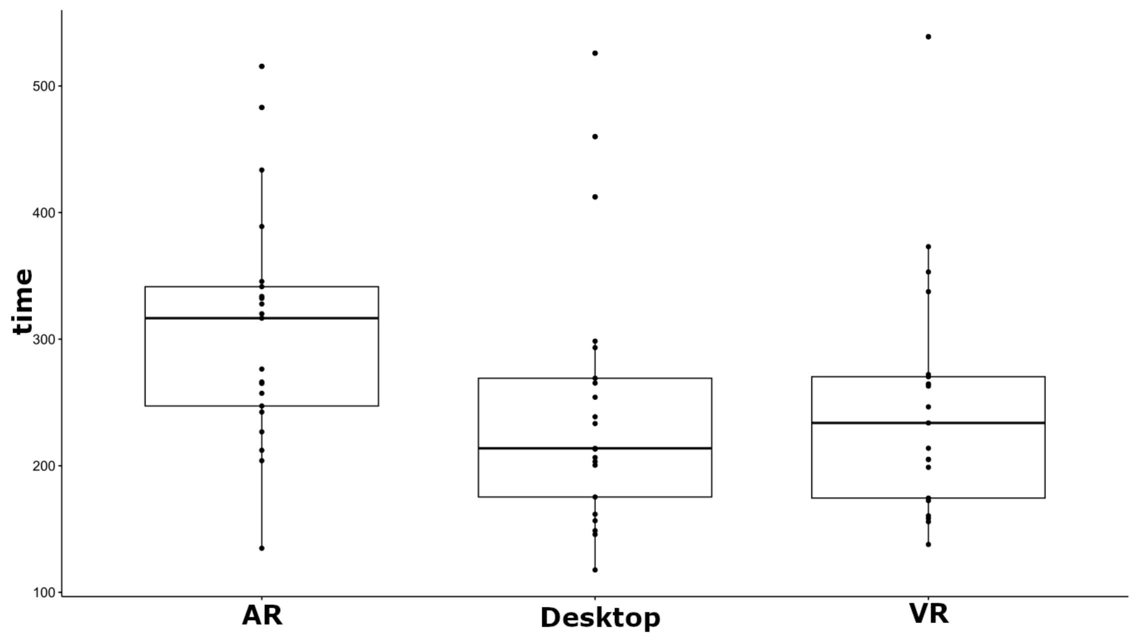


Figure 5.13: Box plot shows the total time distribution in seconds per condition.

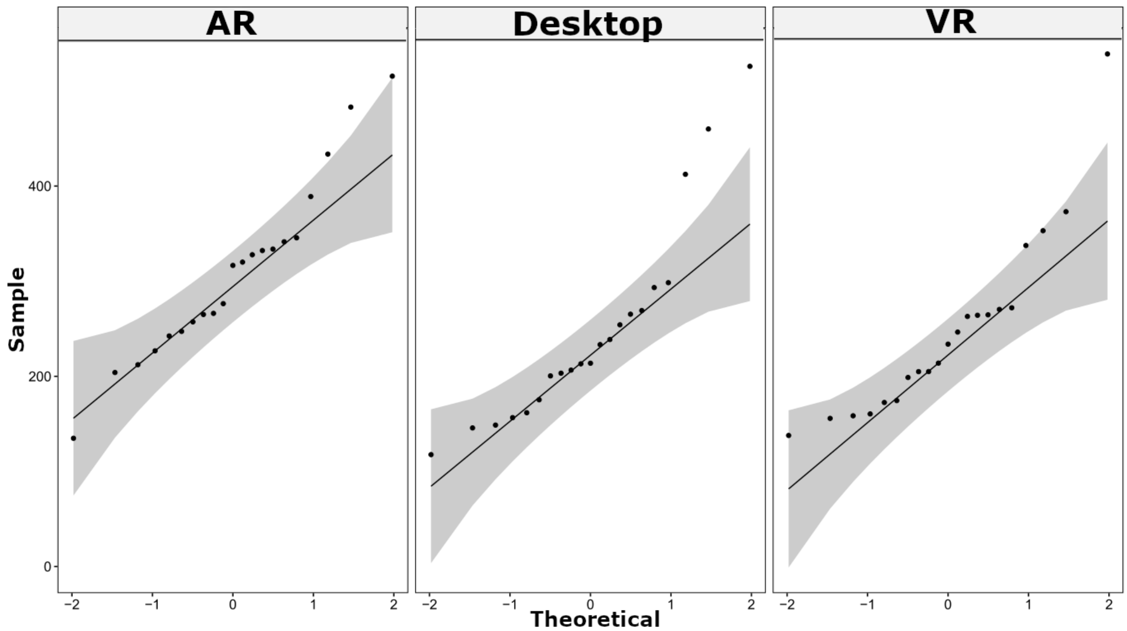


Figure 5.14: QQ plot showing data points for VR and Desktop don't follow a normal distribution.

Error rates were calculated for the object component selection sub task. The object component behavior definition was performed correctly by all participants. For the cubic shape the error was calculated using three points from the cube selector. The error was calculated by summing up the euclidean distances between points positioned by participants with the correct calculated point position. This procedure was repeated on each object. The total error rate distribution can be seen in Figure 5.15. A Shapiro-Wilk test showed the data was normally distributed in each condition and non extreme outliers data points were found. Therefore, a repeated measures ANOVA test was used and revealed that error measurements were statistically significantly different at the different conditions, $\boldsymbol{F(2,40) = 61.79, p < 0.0001}$. Post-hoc analyses with a Bonferroni adjustment revealed that all the pairwise differences, between error measurements, were statistically significantly different ($\boldsymbol{p < 0.0001}$) refer to the exact values in Figure 5.15.

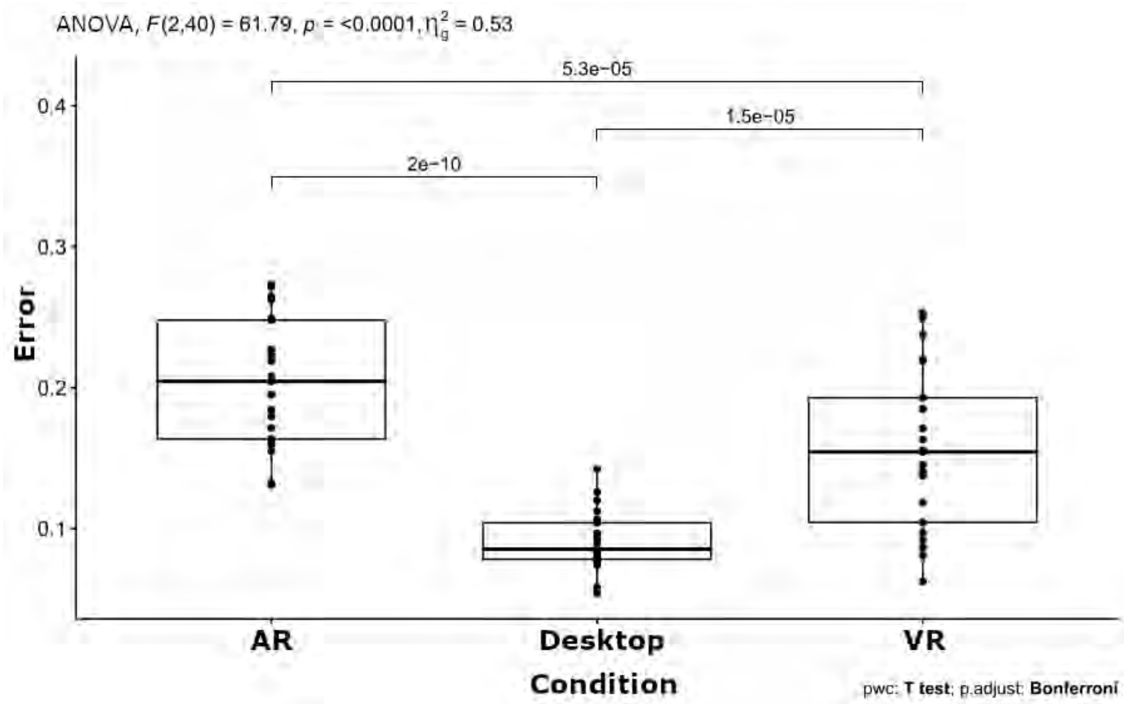


Figure 5.15: Box plot shows the total error measurements distribution in meters.

5.4.2 Usability and Perception

A Friedman's test followed by a post-hoc analysis using Wilcoxon signed-rank tests for pairs is used to analyze the difference between the SUS scores per participant across the three conditions. The SUS score was statistically significantly different across conditions $\boldsymbol{\chi^2(2) = 19.81, p < .00005}$ using Friedman test. Pairwise Wilcoxon signed rank test between interface conditions revealed statistically significant differences in SUS scores between Desktop and AR ($\boldsymbol{Z = -3.74, p < .001}$), Desktop and VR ($\boldsymbol{Z = -2.59, p < .01}$) and VR and AR ($\boldsymbol{Z = -2.88, p < .005}$). The overall usability scores from SUS were for AR ($\mu = 59.88, \sigma = 18.25$), VR ($\mu = 72.5, \sigma = 14.4$) and Desktop ($\mu = 82.38, \sigma = 11.69$).

Table 5.2 shows results for the likert scale ratings from Table 5.1. For question 4 a Mann-Whitney U Test revealed no significant difference on the perceived weight of the headset between participants in AR and VR conditions. Question 5 on the other hand with the same test showed that hand tracking was perceived significantly better in VR (Md=6, n=21) than AR (Md=3, n=21), $\boldsymbol{(U = 112.5, p < .01)}$. The rest of the questions can be summarized as follows:

- Participants rate the importance of the task on the Desktop significantly higher than if performed in AR.
- Participants reported significantly having more prior experience in Desktop than in AR and VR.
- There was no significant difference found on how realistic participants perceived the objects in the scene in each condition.
- Pressing buttons in the UI canvas was more difficult in AR than in Desktop and VR.
- Desktop was perceived as significantly easier to use to complete the tasks assigned than AR and VR.
- AR was significantly less enjoyable experience than VR and Desktop.

Table 5.2: Results on Friedman's test and post-hoc analysis for post-condition likert scale data from Table 5.1.

#	Friedman's Test	AR vs VR	AR vs Desktop	VR vs Desktop
Q1	$\boldsymbol{\chi^2(2) = 10.449, p < .01}$	$Z = -1.79, p = .074$	$\boldsymbol{Z = -2.93, p < .005}$	$Z = -1.55, p = .122$
Q2	$\boldsymbol{\chi^2(2) = 12.926, p < .005}$	$Z = -1.54, p = .123$	$\boldsymbol{Z = -2.89, p < .005}$	$\boldsymbol{Z = -2.41, p < .05}$
Q3	$\chi^2(2) = 3.304, p = .192$	$Z = -1.10, p = .273$	$Z = -1.44, p = .149$	$Z = -0.89, p = .374$
Q6	$\boldsymbol{\chi^2(2) = 21.913, p < .00005}$	$\boldsymbol{Z = -3.50, p < .0005}$	$\boldsymbol{Z = -2.77, p < .01}$	$Z = -1.42, p = .156$
Q7	$\boldsymbol{\chi^2(2) = 23.912, p < .00005}$	$\boldsymbol{Z = -3.07, p < .005}$	$\boldsymbol{Z = -3.51, p < .0005}$	$\boldsymbol{Z = -2.77, p < .01}$
Q8a	$\boldsymbol{\chi^2(2) = 16.095, p < .0005}$	$\boldsymbol{Z = -2.64, p < .01}$	$\boldsymbol{Z = -3.34, p < .001}$	$\boldsymbol{Z = -2.47, p < .05}$
Q8b	$\boldsymbol{\chi^2(2) = 19.902, p < .00005}$	$\boldsymbol{Z = -3.08, p < .005}$	$\boldsymbol{Z = -3.36, p < .001}$	$\boldsymbol{Z = -2.54, p < .05}$
Q9	$\boldsymbol{\chi^2(2) = 16.889, p < .0005}$	$\boldsymbol{Z = -2.98, p < .005}$	$\boldsymbol{Z = -3.25, p < .005}$	$\boldsymbol{Z = -2.33, p < .05}$
Q10	$\boldsymbol{\chi^2(2) = 7.878, p < .05}$	$\boldsymbol{Z = -2.66, p < .01}$	$\boldsymbol{Z = -2.19, p < .05}$	$Z = -0.05, p = .959$

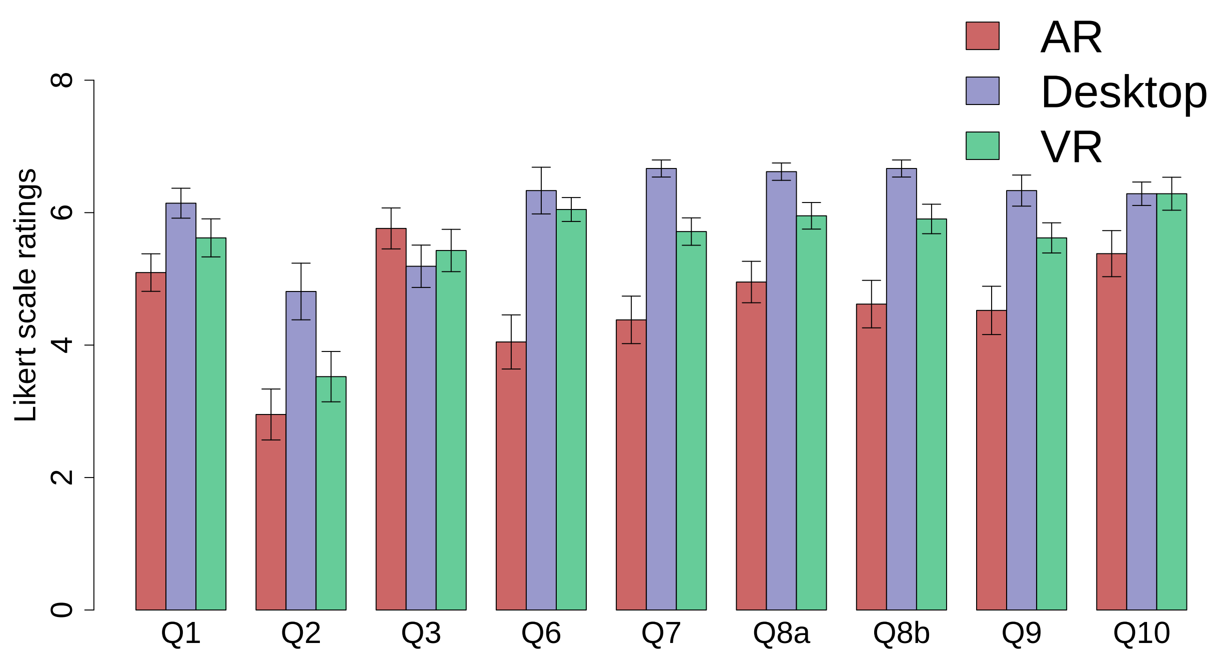


Figure 5.16: Plot shows the mean values and standard deviations for post-condition likert ratings from Table 5.1.

Table 5.3: Results on Friedman's test and post-hoc analysis for NASA-TLX workload ratings reported.

#	Friedman's Test	AR vs VR	AR vs Desktop	VR vs Desktop
MD	$\boldsymbol{\chi^2(2) = 22.235, p < .0005}$	$Z = -1.63, p = .102$	$\boldsymbol{Z = -3.09, p < .005}$	$\boldsymbol{Z = -2.82, p < .005}$
PD	$\boldsymbol{\chi^2(2) = 33.787, p < .0005}$	$Z = -0.60, p = .546$	$\boldsymbol{Z = -3.62, p < .001}$	$\boldsymbol{Z = -3.33, p < .001}$
TD	$\boldsymbol{\chi^2(2) = 6.259, p < .05}$	$Z = -0.85, p = .393$	$Z = -1.89, p = .058$	$Z = -1.05, p = .292$
OP	$\boldsymbol{\chi^2(2) = 8.600, p < .02}$	$Z = -0.38, p = .700$	$\boldsymbol{Z = -2.54, p < .05}$	$Z = -1.87, p = .062$
EF	$\boldsymbol{\chi^2(2) = 9.848, p < .01}$	$Z = -1.02, p = .306$	$\boldsymbol{Z = -2.63, p < .01}$	$\boldsymbol{Z = -2.30, p < .05}$
FL	$\boldsymbol{\chi^2(2) = 21.344, p < .0005}$	$\boldsymbol{Z = -2.25, p < .05}$	$\boldsymbol{Z = -3.27, p < .005}$	$\boldsymbol{Z = -2.21, p < .05}$

5.4.3 Workload

To analyze the workload ratings, we used Friedman's test followed by a post-hoc analysis using Wilcoxon signed-rank tests for pairs (Table 5.3). Average ratings for the workload are summarized in Figure 5.17. From the results that we obtained we concluded the following:

- Participants perceived workload similar between AR and VR in 5 of 6 factors measured.
- Frustration levels were significantly higher between AR and VR.
- AR and VR when compared to Desktop show significantly higher workload ratings in 4 of 6 factors measured.
- Own performance was perceived significantly better when using the Desktop condition than AR.
- No significant difference in temporal demand was found across conditions.

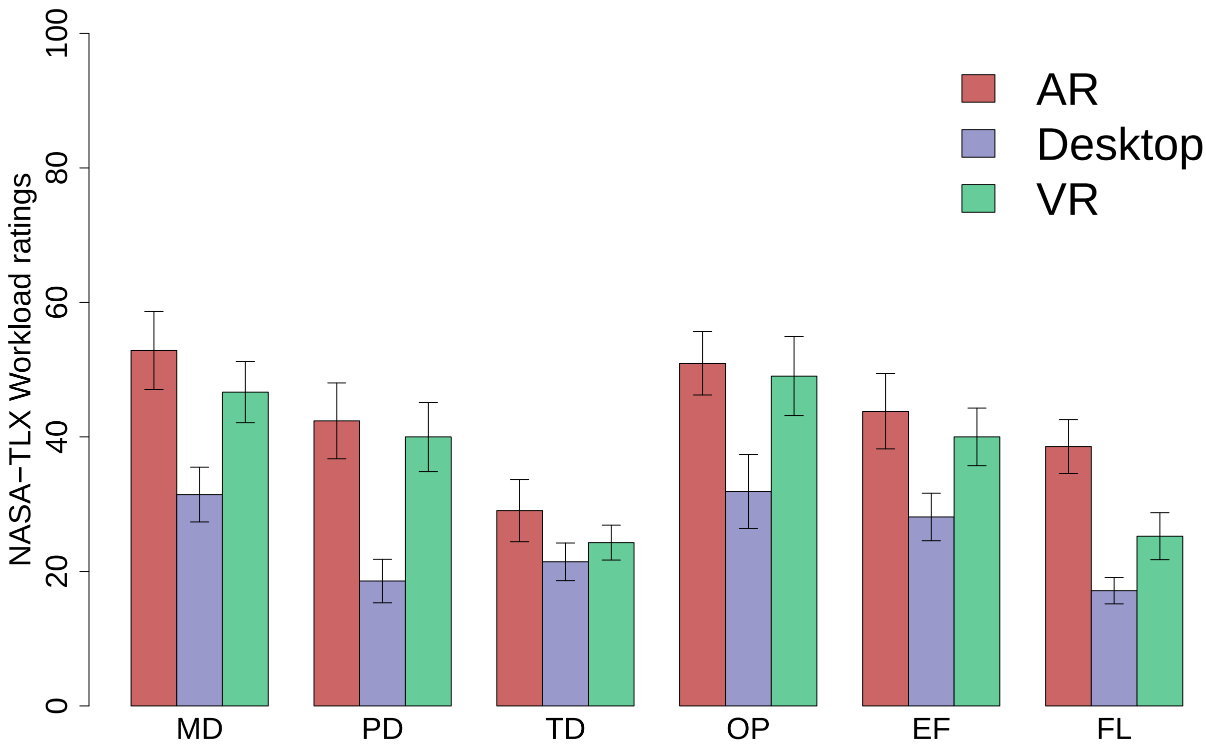


Figure 5.17: Plot shows the mean values and standard deviations for NASA TLX workload ratings.

5.4.4 Preference

In our preference survey, participants were asked to pick a condition in which they would perform the following actions: 1) add a component selection, 2) add a perpendicular pulling behavior, 3) add a rotation behavior and 4) visualize outcome. A chi-square Goodness of Fit Test was performed to determine whether the proportion of the observed number of subjects choosing a condition was equal to a group with proportions equally distributed. The proportion of subjects preference did differ by task 1, $\boldsymbol{\chi^2(2,21) = 8, p < .019}$. The proportion of subjects picking a condition did not differ by task 2, 3 and 4.

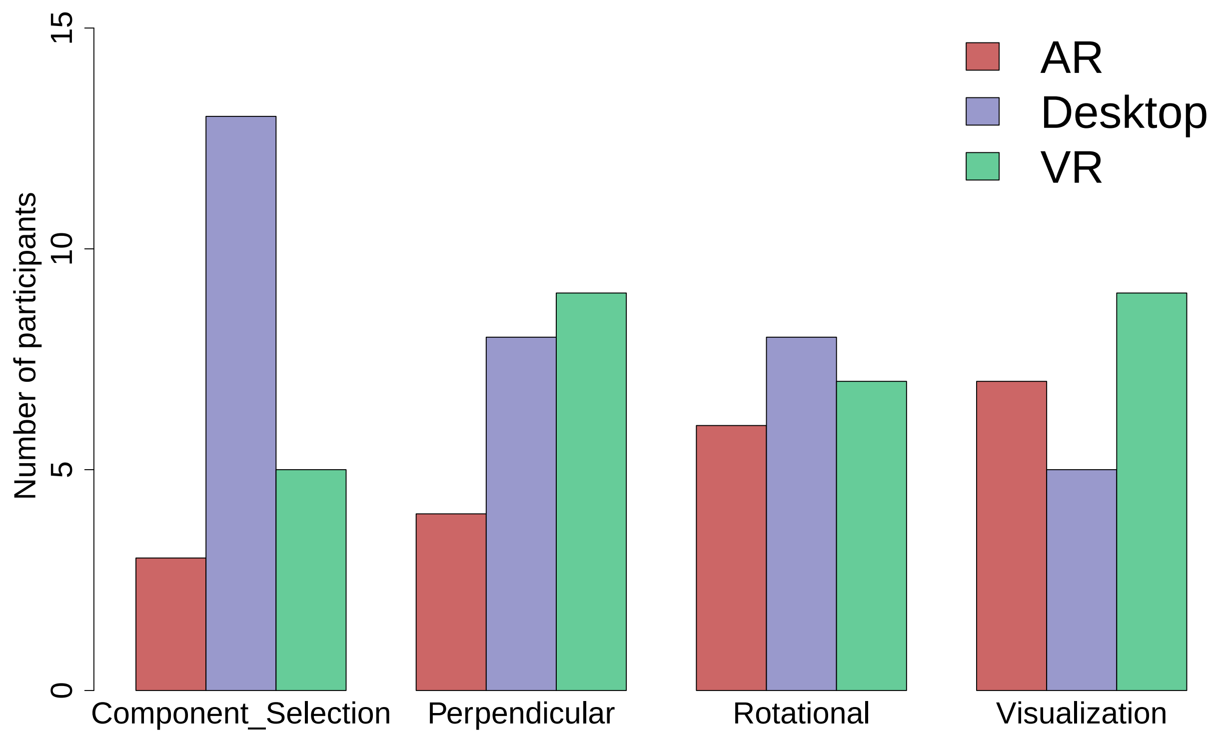


Figure 5.18: Preference for each of the four aspects.

5.5 Discussion

In line with results from AffordIt!, the concepts to define the object component's behaviors were well received. Overall, participants were able to complete the tasks assigned in the three conditions. A similar interaction workflow was followed and implemented across the three interfaces. Interaction techniques and visualization changed in each condition evaluated. In the following paragraphs, our findings are discussed and future directions for more usable AR and VR interfaces are provided.

5.5.1 Authoring Time

Hand tracking contributed to Authoring in AR demanding significantly more time than in Desktop. Interestingly, hand tracking in AR was perceived as less accurate than VR even though the same component was used. The possible reason behind perceiving hand tracking as an issue in AR is due to the little feedback used for the hand while being tracked in AR. For instance, in VR hand tracking lost would involve not seeing the virtual hands at all at any given time. However, in AR the pointer in the index finger was the only visual cue for hand tracking to be perceived. When losing tracking in AR the real hand would be still visible as opposed to VR the hand disappears. Tracking is lost due to participants placing their hands away from the headset view or by the hand getting occluded by the real object. In VR participants intuitively kept their hands in a better position and no occlusion from real objects happened. Some participants commented after the study:

> User 19: *"AR seemed to be more inconsistent with its tracking compared to VR"*

> User 3: *"Tracking is also a problem for AR and VR"*

> User 2: *"AR was really hard to realign my hands and do selections"*

5.5.2 Tasks requiring precise interaction

The object component selection with the cubic shape, required participants to look for corners within the object to place points and accurately surround the component with a selector. Desktop provided a clear advantage for such a task. For tasks requiring higher precision we recommend using a more precise input tool or a completely new interaction technique possibly involving the use of both hands. Participants in AR and VR commented:

User 13 while in VR: *"It was easy to perform the initial interaction with the object. The challenge was lining up the virtual maker points with the objects edges and not performing the gesture".*

Users 16, 17 and 19 in AR: *"I found it easiest to estimate the distances on the system, but it was a bit difficult to get the system to interpret the exact points I wanted to locate."*, *"The easiest was to measure volume. a challenge was trying to get the points exactly where I wanted them to be"*, *"I think the easiest task was selecting the area to cut the volume of the door."*

Despite lack of precision some participants in Desktop reported issues while estimating depth of the cube for the component selection. While this was an issue in Desktop they did not report it in AR or VR. Figuring out the depth for an object component selection involved in most cases navigating the scene with the camera or changing camera perspective which was reported by five participants challenging in Desktop.

Desktop participants 16, 9 and 5: *"I found it easy to select the size of the door at first. It was a bit difficult to calculate how far back the cabinet door went"*, *"Determining depth was more of an estimation and slightly more difficult"*, *"The easiest action to perform was clicking on the object itself, the most challenging one was to figure out depth of the object"*

The abovementioned issues justify the significant difference between conditions in error measurements (see Figure 5.15). The desktop distribution's calculated errors are more compact than VR and AR.

5.5.3 Usability

Workload results showed significantly higher ratings for AR and VR. This is expected due to exertion and more physical demand when navigating the scene. However, AR presents a higher frustration level than VR and Desktop. Similarly, the raw SUS scores are significantly lower for the AR condition, with a mean value below the average. The issues with hand tracking and gesture detection discussed above could have contributed to these scores. Low familiarity of participants with the immersive systems compared to Desktop is another factor to consider for these results. Despite this, participants' comments were positive towards using AR and VR for authoring object component behaviors:

> User 5: *"The AR let's you see how to object would behave more closely in real life"*

> User 4: *"authoring through AR and VR seems more immersive and natural rather than desktop, also it resembles a real life experience when it comes to authoring that behavior."*

> User 1: *"VR experience was more fun and more intuitive"*

> User 2: *"Desktop interface is very good! I would use it the most I think. But it is more fun to open a door in 3D. It really feels like I'm opening and closing it more there."*

> User 19: *"Authoring in 3D has great advantages from doing the experience. It is a lot easier to navigate to the objects, and with AR, you can physically touch the objects*

and model them more accurately. I think as tracking improves, it is definitely the next step for modeling objects and is much more intuitive"

5.5.4 Recommendation for authoring object components' behaviors

For tasks involving precise interactions, this work recommends using a Desktop interface. Further research needs to consider higher precision input tools in 3D that simulates the mouse effect in a Desktop environment. The object component selection was by far the most difficult part in AR and VR and most participants preferred a Desktop interface to author such tasks as seen in Figure 5.18. Authoring a behavior did not require high precision in the positioning of points. According to the preference scores, participants did not significantly choose one condition or the other for these types of tasks. However, participants found it significantly easier to perform these tasks on Desktop. As for visualizing the authored behavior, no interface was preferred significantly more than the other. This work recommends to author behaviors in desktop and visualize results in 3D immersive environments as it was found more compelling experience as participants report post experience.

User 17: *"Augmented reality made it feel more real since real objects were being used, so it was easy visualize it a bit more"*,

User 14: *"AR and VR were pretty similar to me, I feel AR more safe since you are interacting with the actual environment but I understand cases were VR would be necessary."*

User 21: *"It is very different in terms that we get to visually see what it is in real life and you can measure with your eyes to get a better feel for how an object should* *behaved. In VR, you are looking at an animated version therefore it is a tad different and desktop is the same it is a 2d or it does not feel like we are in there for real."*

With the increasing details in realism and graphics capabilities coupled with hardware that provides high pixel density and higher resolutions for head mounted displays. It is interesting to understand how much realism affects the experience as participant number 18 mentions in the following quote:

> *"I think, in AR, it was slightly more difficult to draw the line between reality and what's augmented. In the sense of, I am in this virtual world, but these are real objects made virtual. As a result, the lines were more blurred. In VR, I know it's virtual, even if the world around me is real but I know that I do not have to physically interact with the items. I think that creates a slight mental shift, in terms of accuracy and expected behaviors. For the desktop, I think it's easy to get a feel for the behavior to expect due to the fact that there is no sense of virtual attached to it. It's an item on a desktop and you know what outcome to expect in your behaviors."*

Blending highly realistic graphics with reality can change what one expects to do or how an experience is perceived. This work recommends further research into how interaction and graphics fidelity affect user experience.

5.6 Future Work & Conclusion

This work implements the AffordIt! concept across three different conditions to better understand participants perceived ease of use and performance. The conditions followed a similar interaction flow and differ in the interaction techniques and display. Results from the experiment show that authoring on the desktop is perceived as more efficient and easier to perform. AR and VR usability was affected by limitations on hand tracking, to overcome this issue a higher precision input tool could be used. A different interface unbounded from the room scenario needs to be proposed to combat higher AR and VR workloads. An exploration of seated VR and AR coupled with scale-down objects that fit on a Desktop could help reduce workload as participants would need less exertion.

We acknowledge that there is a limitation in the study given that the systems were tested with one task. In the future, a complete evaluation would involve different types of tasks and objects with different geometries and components within the context of a given scenario.

An additional area for exploration is the seamless use of a 2D canvas for higher precision tasks within an immersive environment that could allow to visualize the authoring outcome as it happens. Blending 2D and 3D interactions could alleviate some of the challenges found in this work. As explained above limitations in hand tracking could have hindered better results for AR and to less extent with VR. With a more precise interaction tool e.g. stylus, results would improve to be comparable to or better than desktops. Finally, an additional area for further exploration is how two or more participants would collaborate to author component behaviors on more complex digital twins.

This chapter evaluates three custom-built applications adopting the AffordIt principles defined in the previous chapter. Differences were found across the three interfaces in error measurements but not in time spent performing tasks. Significance was only found between AR and Desktop for time completion. Participants favored the Desktop interface due to its ease of use and precision. However, participants' perception of AR and VR was reported as very positive and encouraging from preference results and interview questions. Visualizing the result of the outcome in AR and VR was well received in comments post-experience. Our findings and challenges were presented in this chapter, also proposed solutions for future iterations.

CHAPTER 6: CONCLUSION AND FUTURE RESEARCH

This dissertation presents a set of tools to author scenario-based training simulations. The different tools from this dissertation were custom-built using AR, VR and the Desktop as interface conditions. Authoring was performed in a real and virtual environment with virtual and real objects. To minimize the difference in the environment setup, a 3D scan of the room the experiments took place was used. In addition, the objects used in the experiments resemble digital twins in the real world. Based on results from our studies overall at the current state when authoring scenario-based training simulations in Desktop and AR no significant difference was found in performance time or perceived usability. Similarly, when authoring object component behaviors no difference was found in time performance between VR and Desktop, but AR and Desktop were significantly different. As for error measurements for precise tasks, as in the object component selection, the Desktop was substantially more accurate than AR and VR. Despite the limitations of the studies, AR and VR were found enjoyable and well-received from participants' comments. In addition, AffordIt! with VR controllers yield slightly better results in usability than the counterpart with hand-tracking interaction.

In the following paragraphs, we will reference each of the experiments with an abbreviation to identify where the recommendation was drawn from. For chapter 3, scenario authoring, we will use **SCN**, for chapter 4, AffordIt! the choice is **AFF** and for chapter 5, for the object component behavior authoring in real and virtual objects, the abbreviation is **OCB**. The identifier is added at the end of the recommendation.

Recommendations based on our findings in the studies and future work are as follow:

When authoring a scenario:

1. When authoring from an egocentric point of view, one can lose the global state within the scenario. A global work progress status is required while in an immersive environment (SCN).

2. Provide a mode to toggle between author mode and playback mode so as to experience the scene while authoring (SCN).

3. In Desktop and VR tools, a 3D reconstruction of the scenario is required for context, in AR the context is given by the real environment coupled with good spatial mapping (SCN, OCB).

4. Visualizing the outcome was a favorite moment from the AR/VR. When possible we recommend visualizing the outcome while immersed in a 3D environment (SCN).

Future research aspects like the authoring flow from the study in chapter 3 (graph modeling) need further exploration in 3D environments. This is a feature to improve for situated AR authoring, and in general how to visualize the progression of the scene authoring while being immersed. A possible solution could be to use flat areas in the real world, such as walls, or a notebook metaphor to provide a 2D general view of the progress. Another alternative could be the use of World In Miniature techniques [138] to have a God-mode perspective.

Adding attributes and actions between real and virtual objects can enhance the realism and fidelity of the scenario, increasing the participants feeling of immersion in the AR condition. It is an open question as to how visual realism affects the user's experience in the Desktop setup, and how participants are affected by working with different mesh qualities. Currently, high-detail scans require very expensive devices. In addition, attributes and actions could be made more complex, for instance, timing or closing events can be added as means to start another action or the flow of actions can be non-linear. In a future iteration, predefined events should be easy to add nonprogrammatically and support the possible generation of animations.

When interacting with 3D floating panels and UI components

1. We recommend using laser pointing for selection and filling forms in room-size scenarios (SCN).

2. Floating panel canvas for UI interaction was generally well received and a familiar paradigm in the three studies performed (SCN, AFF, OCB).

3. Interaction with floating panels and UI components was preferred when user-centric and easy to reach as opposed to anchoring the 3D canvas to each virtual or real object (SCN, AFF).

4. From observation, dragging a 3D canvas with a controller by colliding it with the canvas while holding a button yielded better results than using a pinch gesture while colliding the hand with the canvas (AFF, OCB).

Particularly, from observation participants felt more comfortable using controllers than hand-tracking for interacting with 3D UI components such as buttons or moving a 3D canvas to a specific position. This could be due to familiarity with the use of the controllers and a better sense of tracking. Participants rather preferred to have the 3D canvas within their field of view instead of having to walk to a particular object to interact with. The use of the wrist or specific parts of the body is an area that could be explored further as it was well received the positioning of an Undo button on the non-dominant hand of the user recommended by work in [78, 82].

When authoring objects' behaviors

1. We suggest creating or generating primitives to select and segment mesh components (AFF).

2. When possible while authoring is being performed show the behavior outcome in real-time (AFF).

3. Generate interaction templates that can be easily extendable by combining them or attaching them to one or more objects (AFF, OCB).

4. Bring objects to the users rather than users to the objects. This can be explored further as the scope of this dissertation was on real-size objects (AFF, OCB).

Future exploration can look into segmenting the objects' parts automatically through approaches such as [76, 132, 149]. AffordIt! can be used together with these tools as a human-in-the-loop tool to modify or adjust the outputs of the automatic segmentation. Intertwining the automatic approaches with AffordIt! will provide the user with an easy-to-use interface to correct the errors on the automatically segmented areas or use the quickly segmented areas to create affordances/behaviors. One participant commented:

> ”have the person open the item in AR and have the computer automatically detect the door and its rotation.”

AffordIt! can be extended with more affordances and mesh cutters with a possible combination of them to produce more complex behaviors. For instance, we could have an interaction that requires moving an object in a particular trajectory while rotating it simultaneously, such as the behavior of a screwdriver. The mesh cutter can be extended to allow for more shape flexibility; for instance, we could create convex polyhedrons as shown in [133]. Furthermore, we intend to adopt an affordance framework as seen in Kapadia et al. [69]. Also, we can develop interactions similar to [133] such that a user can draw the region of interest, snapping points to the most likely portion of the object, rather than relying on pre-defined selection shapes. This could provide increased accuracy and remove human error.

For future work, alternatives to reduce workload can explore seated VR/AR while immersed in a 3D environment. The use of traditional interaction techniques within the immersive environment should be considered [37]. Another alternative to reduce the workload is to work with miniature replicas that can fit within a desktop space. The authoring can happen in the miniature digital twin and is visualized in the real environment.

An additional exploration for object behavior authoring is using passive haptics to author real objects. A participant can be immersed in VR, and the visuals are entirely virtual, mapped to the real position and dimensions of the real objects in the scene. Finally, collaboration can be explored more in the context of authoring scenario-based training simulations and object component behaviors.

About input devices and visual cues

1. For tasks requiring precise interaction, explore a different tool, such as a stylus or develop a different set of 3D interaction techniques for it (OCB).

2. Provide visual cues in AR for notifying hand lost tracking without breaking immersion (OCB).

3. Provide more visual aids to help participants perceive better depth (AFF, OCB).

The only task requiring precise interaction involved finding corners in each of the objects to create a region of interest, this was particularly challenging for participants as in some cases hand shaking or distance perception would hinder the exact positioning of the point within the expected bounds of the component. An additional challenge was found when the hand lost tracking however was still visible in the view of the participant since the hand was not hidden as in the VR counterpart. These precise interactions are similar to modelling or drawing which require a higher sensitive input device to provide the users with the ability to point and click as in a traditional Desktop with a mouse. Another approach to solving precise interactions is to snap to corners or edges in the objects. To achieve this a more intelligent assistive technique is required for further exploration.

We provided different guidelines and strategies that could further improve the usability and reduce the workload of authoring tools implementations in the context of scenario and object behavior's authoring. Decisions taken in user interfaces and system designs were grounded in previous work as well as principles and design guidelines for building 3D UI interfaces. Through experimentation, different approaches and interaction devices were used to test its effectiveness and details are given in each chapter to justify the findings. This dissertation considers scenarios that take place within a room size and object behaviors positioned permanently in the environment. At the macro level, Future work can focus on:

- A different area of the space continuum defined in Figure 1.1.
- Narrow down to complete definitions of scenarios with linear and non-linear progression.
- Explore the affordance templates to create nested interactions with more complex behaviors.
- Consider other aspects from the object interaction definition by Kallmann [64]. Aspects such as objects' manipulation metaphors.

While the Desktop interface presented clear advantages such as familiarity and well-known established interaction techniques. Results from studies also showed participants' high engagement and joy while using immersive approaches. This thesis concludes that an immersive approach does not necessarily need to provide better capabilities than a Desktop interface in all aspects. However, in order to take full advantage of situated authoring, excessive workload and low usability are needed to be improved so as to not be a factor that could hinder the potential benefits of authoring in-situ augmented reality scenario-based training experiences.

APPENDIX A: IRB APPROVAL FIRST STUDY

University of Central Florida Institutional Review Board
Office of Research & Commercialization
12201 Research Parkway, Suite 501
Orlando, Florida 32826-3246
Telephone: 407-823-2901 or 407-882-2276
www.research.ucf.edu/compliance/irb.html

Approval of Human Research

From: **UCF Institutional Review Board #1**
FWA00000351, IRB00001138

To: **Andres Vargas Gonzalez** and Co-PI: **Katelynn A. Kapalo**

Date: **August 08, 2018**

Dear Researcher:

On 08/08/2018 the IRB approved the following human participant research until 08/07/2019 inclusive:

Type of Review:	UCF Initial Review Submission Form Expedited Review Category 7
Project Title:	Authoring Intelligent Tutoring Scenarios in Augmented Reality
Investigator:	Andres Vargas Gonzalez
IRB Number:	SBE-18-14157
Funding Agency:	US Army Research Laboratory
Grant Title:	
Research ID:	N/A

The scientific merit of the research was considered during the IRB review. The Continuing Review Application must be submitted 30days prior to the expiration date for studies that were previously expedited, and 60 days prior to the expiration date for research that was previously reviewed at a convened meeting. Do not make changes to the study (i.e., protocol, methodology, consent form, personnel, site, etc.) before obtaining IRB approval. A Modification Form **cannot** be used to extend the approval period of a study. All forms may be completed and submitted online at https://iris.research.ucf.edu .

If continuing review approval is not granted before the expiration date of 08/07/2019, approval of this research expires on that date. When you have completed your research, please submit a Study Closure request in iRIS so that IRB records will be accurate.

Use of the approved, stamped consent document(s) is required. The new form supersedes all previous versions, which are now invalid for further use. Only approved investigators (or other approved key study personnel) may solicit consent for research participation. Participants or their representatives must receive a copy of the consent form(s).

All data, including signed consent forms if applicable, must be retained and secured per protocol for a minimum of five years (six if HIPAA applies) past the completion of this research. Any links to the identification of participants should be maintained and secured per protocol. Additional requirements may be imposed by your funding agency, your department, or other entities. Access to data is limited to authorized individuals listed as key study personnel.

In the conduct of this research, you are responsible to follow the requirements of the Investigator Manual.

This letter is signed by:

Signature applied by Gillian Morien on 08/08/2018 10:18:48 AM EDT

Designated Reviewer

APPENDIX B: IRB APPROVAL SECOND STUDY

UNIVERSITY OF CENTRAL FLORIDA

Institutional Review Board
FWA00000351
IRB00001138Office of Research
12201 Research Parkway
Orlando, FL 32826-3246

APPROVAL

January 2, 2020

Dear Sina Masnadi:

On 1/2/2020, the IRB reviewed the following submission:

Type of Review:	Initial Study
Title:	VR Manipulation
Investigator:	Sina Masnadi
IRB ID:	STUDY00001233
Funding:	None
Grant ID:	None
IND, IDE, or HDE:	None
Documents Reviewed:	• Email.docx, Category: Recruitment Materials; • HRP-502 - TEMPLATE - Consent Document (Adult) (1).pdf, Category: Consent Form; • HRP-503 - TEMPLATE - Protocol (3).docx, Category: IRB Protocol; • post-questionnaire, Category: Survey / Questionnaire; • pre-questionnaire, Category: Survey / Questionnaire; • VR Environment, Category: Other;

The IRB approved the protocol from 1/2/2020.

In conducting this protocol, you are required to follow the requirements listed in the Investigator Manual (HRP-103), which can be found by navigating to the IRB Library within the IRB system.

If you have any questions, please contact the UCF IRB at 407-823-2901 or irb@ucf.edu. Please include your project title and IRB number in all correspondence with this office.

Sincerely,

Adrienne Showman
Designated Reviewer

Joseph J. LaViola Jr.
Charles N. Millican Professor of Computer Science
Department of Computer Science

Tuesday, November 29, 2022

Joseph J. LaViola Jr.
Charles N. Millican Professor of Computer Science Director,
Interactive Computing Experiences Research Cluster
University of Central Florida
Department of Computer Science
Orlando, FL 32816-2362
Phone: 407-882-2285

To Whom It May Concern:

This letter states that the study with IRB ID STUDY00001233 is connected to the research done for the paper published in Proceedings of Graphics Interface 2020 with title "AffordIt!: A Tool for Authoring Object Component Behavior in Virtual Reality." Chapter 4 in this dissertation is based on this published work.

Sincerely,

Dr. Joseph J. LaViola Jr.
Professor, CS

Department of Computer Science
College of Engineering and Computer Science • University of Central Florida
PO Box 162362 • Orlando, FL 32816-2362 • Phone (407) 823-0185 • Fax (407) 823-1488
www.cs.ucf.edu

APPENDIX C: IRB APPROVAL THIRD STUDY

UCF
UNIVERSITY OF CENTRAL FLORIDA

Institutional Review Board
FWA00000351
IRB00001138, IRB00012110
Office of Research
12201 Research Parkway
Orlando, FL 32826-3246

APPROVAL

October 31, 2022

Dear Andres Vargas Gonzalez:

On 10/31/2022, the IRB reviewed the following submission:

Type of Review:	Initial Study, Expedited Category 7; Waiver of Written Documentation of Consent
Title:	A Comparison of Virtual Reality, Augmented Reality and Desktop for Authoring object components behaviors and task interactions for virtual and real objects.
Investigator:	Andres Vargas Gonzalez
IRB ID:	STUDY00004711
Funding:	None
IND, IDE, or HDE:	None
Documents Reviewed:	• ad, Category: Recruitment Materials; • AR-Participant view, Category: Other; • ConsentFormUpdated, Category: Consent Form; • Demographics and Prior Exp.pdf, Category: Survey / Questionnaire; • PostConditionQuestionnaire.pdf, Category: Survey / Questionnaire; • PostExperience Preference.pdf, Category: Survey / Questionnaire; • Protocol - AuthoringToolsMR, Category: IRB Protocol; • VR-Participant view, Category: Other;

The IRB approved the protocol on 10/31/2022.

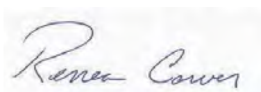
In conducting this protocol, you are required to follow the requirements listed in the Investigator Manual (HRP-103), which can be found by navigating to the IRB Library within the IRB system. Guidance on submitting Modifications and a Continuing Review or Administrative Check-in is detailed in the manual. If continuing review is required and approval is not granted before the expiration date, approval of this protocol expires on that date.

Use of the stamped version of the consent form is required. To document consent, use the consent documents that were approved and stamped by the IRB. Go to the Documents tab to download them.

When you have completed your research, please submit a Study Closure request so that IRB records will be accurate.

If you have any questions, please contact the UCF IRB at 407-823-2901 or irb@ucf.edu. Please include your project title and IRB number in all correspondence with this office.

Sincerely,



Renea Carver
Designated Reviewer

APPENDIX D: DEMOGRAPHICS & PRIOR EXPERIENCE QUESTIONNAIRE FIRST STUDY

UNIVERSITY OF CENTRAL FLORIDA

What is your age in years? (YOU MUST BE 18 or OLDER TO PARTICIPATE IN THE STUDY)

What gender do you identify as?

- Male
- Female

What is your major?

Which hand do you normally favor (writing, sports, etc.)?

- Right
- Left

Please rate how familiar you are with 3D modeling applications (with 1 = not familiar and 7 = very familiar)

	1	2	3	4	5	6	7
I am familiar with 3D modeling tools such as blender, maya, etc.	○	○	○	○	○	○	○

Have you ever played an augmented reality game before?

- Yes
- No

If you answered "Yes" to the previous question, please describe the capacity in which you played.

Have you ever used an authoring or instructional design tool to create e-learning content?

Yes

No

If you answered yes to the above question, which tool did you use and in what capacity?

>>

UNIVERSITY OF CENTRAL FLORIDA

The following survey will ask you about your prior experience and knowledge related to 3D modeling and 3D user interfaces.

1. Please rate your current experience and knowledge on the scale below.

	1 -Very little experience	2	3	4-Neither very experienced or inexperienced	5	6	7-Very experienced
3D modeling (Maya, Blender, Rhino, 3D Studio Max) or 3D interfaces in genera	○	○	○	○	○	○	○
Authoring (creation) of online tutoring classes and/or experiences	○	○	○	○	○	○	○

Please rate your agreement with the following statement:

	1-- Strongly Disagree	2	3	4-- Neither Agree nor Disagree	5	6	7-- Strongly Agree
I use the computer frequently for work	○	○	○	○	○	○	○
I use the computer frequently for school-related activities	○	○	○	○	○	○	○
I use a computer frequently for personal use (surfing the internet, youtube, music, etc.)	○	○	○	○	○	○	○
I prefer to use a desktop setup	○	○	○	○	○	○	○
I prefer to use a laptop	○	○	○	○	○	○	○

How often do you use...

	1-Not often at all	2	3	4-Once in a while	5	6	7-Very Often
Training Applications/Online Tutoring Materials	○	○	○	○	○	○	○
Virtual Reality Applications	○	○	○	○	○	○	○
Augmented Reality Applications	○	○	○	○	○	○	○

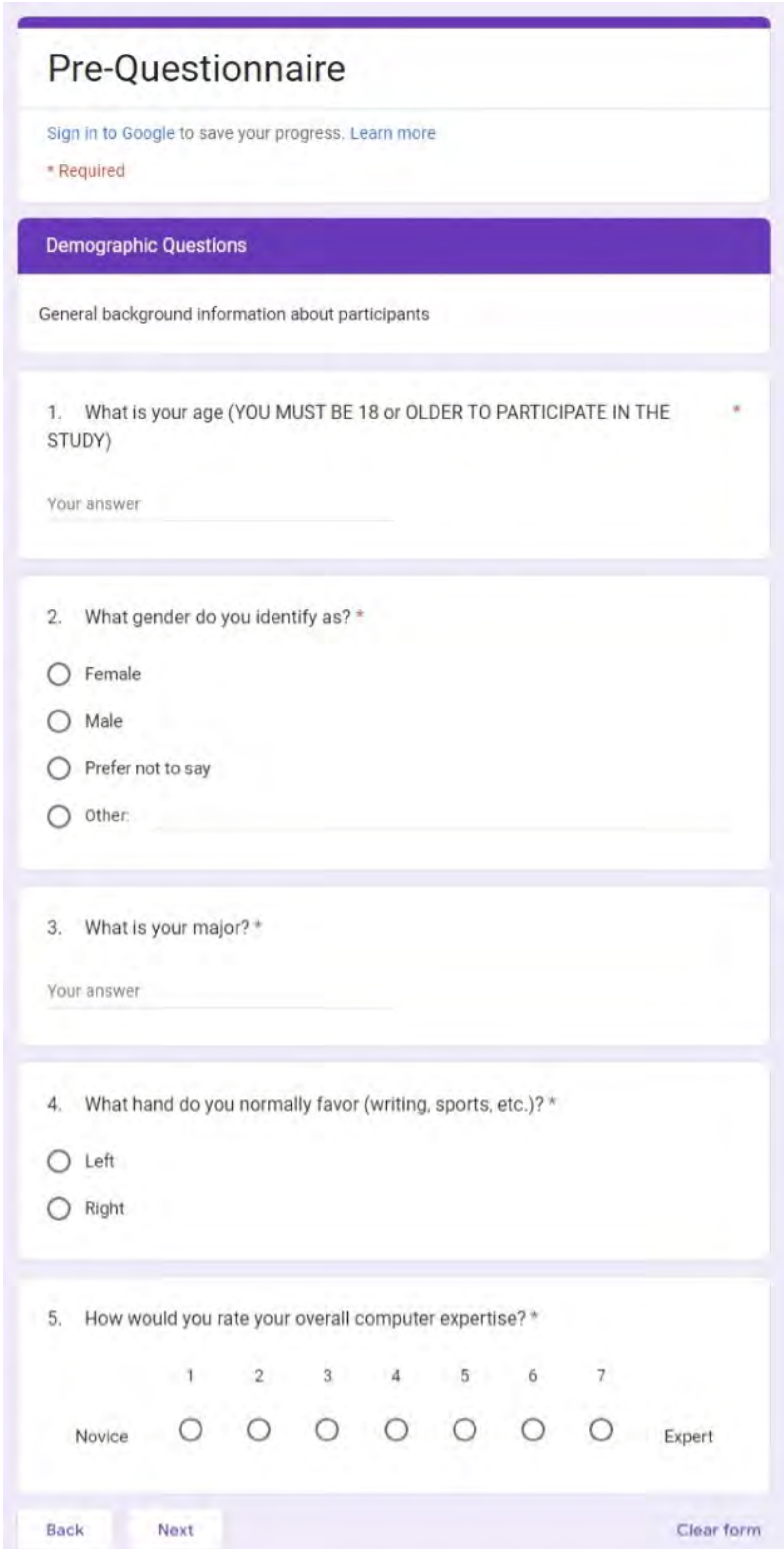
How would you rate your understanding of the following tools:

	1--Never use this tool	2	3	4--Occasionally use this tool	5	6	7--Use this tool very frequently
Unity 3D	○	○	○	○	○	○	○
Maya	○	○	○	○	○	○	○
3D Studio Max	○	○	○	○	○	○	○
Blender	○	○	○	○	○	○	○

The experimenter will enter a number in this section.

>>

APPENDIX E: DEMOGRAPHICS & PRIOR EXPERIENCE QUESTIONNAIRE SECOND STUDY



Pre-Questionnaire

Sign in to Google to save your progress. Learn more

* Required

Demographic Questions

General background information about participants

1. What is your age (YOU MUST BE 18 or OLDER TO PARTICIPATE IN THE STUDY) *

Your answer

2. What gender do you identify as? *

- ◯ Female
- ◯ Male
- ◯ Prefer not to say
- ◯ Other:

3. What is your major? *

Your answer

4. What hand do you normally favor (writing, sports, etc.)? *

- ◯ Left
- ◯ Right

5. How would you rate your overall computer expertise? *

	1	2	3	4	5	6	7	
Novice	◯	◯	◯	◯	◯	◯	◯	Expert

Back Next Clear form

Pre-Questionnaire

Sign in to Google to save your progress. Learn more

* Required

Background

1. Please rate your experience in VR systems. [1=Have never used VR, 4=Have used VR sometimes, 7=Have used VR systems a lot] *

	1	2	3	4	5	6	7	
Have never used VR	○	○	○	○	○	○	○	Have used VR a lot

2. How frequently do you play computer or video games [1= not frequently, 7=very frequently] *

	1	2	3	4	5	6	7	
Not frequently	○	○	○	○	○	○	○	Very frequently

3. Please rate how familiar you are with 3D modeling applications [with 1 = not familiar and 7 = very familiar]. Blender, Maya, 3D Studio Max, Sketch Up, etc. *

	1	2	3	4	5	6	7	
Not familiar at all	○	○	○	○	○	○	○	Very familiar

4. I use the computer frequently for work or school related activities. *

	1	2	3	4	5	6	7	
Not so much	○	○	○	○	○	○	○	Very frequently

5. Have you ever used Virtual Reality to author or create something on a scenario *

○ Yes

○ No

6. If you answered yes to the above question, what did you do in the VR experience. (Enter "N/A" if not applicable) *

Your answer

Back Submit Clear form

APPENDIX F: DEMOGRAPHICS & PRIOR EXPERIENCE QUESTIONNAIRE THIRD STUDY

UNIVERSITY OF CENTRAL FLORIDA

The experimenter will enter a number in this section.

What is your age (YOU MUST BE 18 or OLDER TO PARTICIPATE IN THE STUDY) ?

What gender do you identify as?

- Male
- Female
- Non-binary / third gender
- Prefer not to say

What is your major?

What hand do you normally favor (writing, sports, etc.)?

- Left
- Right

How would you rate your overall expertise using computers?

1 2 3 4 5 6 7

Expertise

How would you rate your overall expertise using VR applications?

1 2 3 4 5 6 7

Expertise

How would you rate your overall expertise using AR applications?

1 2 3 4 5 6 7

Expertise

If previous VR/AR experience if applicable did you use hands, controllers or other input technique

- Hands
- Controllers
- Hands and Controllers
- No previous experience, not Applicable

If previous VR/AR experience do you remember the application name, if so please add it below

→

LIST OF REFERENCES

[1] C. B. Achour. Guiding scenario authoring. In *In: 8th European-Japanese Conference on Information Modelling and Knowledge Bases*, pages 152–171. IOS Press, 1998.

[2] I. Apple Computer. *Macintosh Human Interface Guidelines*. Addison-Wesley Publishing Company, USA, 1992.

[3] S. AR. Worklink create your own smart instructions, https://www.scopear.com/products/worklink/, 2019.

[4] S. Aukstakalnis. *Practical augmented reality: A guide to the technologies, applications, and human factors for AR and VR*. Addison-Wesley Professional, 2016.

[5] C. Baber. Designing smart objects to support affording situations: Exploiting affordance through an understanding of forms of engagement. *Frontiers in psychology*, 9:292, 2018.

[6] I. Barakonyi and D. Schmalstieg. Immersive content authoring for computer entertainment in augmented reality. In *Int. Workshop on Mixed Reality, 2006*, 2006.

[7] A. Bellarbi, N. Zenati, S. Otmane, H. Belghit, S. Benbelkacem, A. Messaci, and M. Hamidia. A 3d interaction technique for selection and manipulation distant objects in augmented reality. In *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, pages 1–5. IEEE, 2017.

[8] M. Billinghurst. Rapid prototyping for ar/vr experiences. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–4, 2020.

[9] D. A. Bowman and L. F. Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. *SI3D*, 97:35–38, 1997.

[10] K. Breakey, D. Levin, I. Miller, and K. Hentges. The use of scenario-based-learning interactive software to create custom virtual laboratory scenarios for teaching genetics. *Genetics. 2008*, 179:1151–1155, 2008.

[11] A. Brito. *Blender Quick Start Guide: 3D Modeling, Animation, and Render with Eevee in Blender 2.8*. Packt Publishing Ltd, 2018.

[12] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[13] J. S. Brown, A. Collins, and P. Duguid. Situated cognition and the culture of learning. *Educational Researcher*, 18(1):32–42, 1989.

[14] F. Buttussi and L. Chittaro. Effects of different types of virtual reality display on presence and learning in a safety training scenario. *IEEE transactions on visualization and computer graphics*, 24(2):1063–1076, 2017.

[15] R. Cabello et al. Three. js. *URL: https://github. com/mrdoob/three. js*, 2010.

[16] S. Cass. Tiltbrush: The killer app for vr. *IEEE Spectrum*, 2016.

[17] H. Chad Lane, M. G. Core, M. J. Hays, D. Auerbach, and M. Rosenberg. Situated pedagogical authoring: Authoring intelligent tutors from a student's perspective. In C. Conati, N. Heffernan, A. Mitrovic, and M. F. Verdejo, editors, *Artificial Intelligence in Education*, pages 195–204, Cham, 2015. Springer International Publishing.

[18] Y. S. Chang, B. Nuernberger, B. Luan, and T. Höllerer. Evaluating gesture-based augmented reality annotation. In *3D User Interfaces (3DUI), 2017 IEEE Symposium on*, pages 182–185. IEEE, 2017.

[19] R. Che, M. Yusoff, H. Badioze Zaman, and A. Ahmad. Evaluation of user acceptance of mixed reality technology. *Australasian Journal of Educational Technology*, 27, 12 2011.

[20] S. Chidambaram, H. Huang, F. He, X. Qian, A. M. Villanueva, T. S. Redick, W. Stuerzlinger, and K. Ramani. Processar: An augmented reality-based tool to create in-situ procedural 2d/3d ar instructions. In *Designing Interactive Systems Conference 2021*, pages 234–249, 2021.

[21] S. Cooper, W. Dann, and R. Pausch. Alice: a 3-d tool for introductory programming concepts. In *Journal of Computing Sciences in Colleges*, volume 15, pages 107–116. Consortium for Computing Sciences in Colleges, 2000.

[22] N. Dahlstrom, S. Dekker, R. van Winsen, and J. Nyce. Fidelity and validity of simulator training. In *Simulation in Aviation Training*, pages 135–144. Routledge, 2017.

[23] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, pages 319–340, 1989.

[24] M. F. Deering. Holosketch: a virtual reality sketching/animation tool. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 2(3):220–238, 1995.

[25] B. Ens, F. Anderson, T. Grossman, M. Annett, P. Irani, and G. Fitzmaurice. Ivy: Exploring spatially situated visual programming for authoring and understanding intelligent environments. In *Proceedings of the 43rd Graphics Interface Conference*, GI '17, pages 156–162, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2017. Canadian Human-Computer Communications Society.

[26] J. A. Erkoyuncu, I. F. del Amo, M. Dalle Mura, R. Roy, and G. Dini. Improving efficiency of industrial maintenance with context aware adaptive authoring in augmented reality. *CIRP Annals-Manufacturing Technology*, 2017.

[27] L. Feng, X. Yang, and S. Xiao. Magictoon: A 2d-to-3d creative cartoon modeling system with mobile ar. In *Virtual Reality (VR), 2017 IEEE*, pages 195–204. IEEE, 2017.

[28] J. A. Frank and V. Kapila. Mixed-reality learning environments: Integrating mobile interfaces with laboratory test-beds. *Computers & Education*, 110:88–104, 2017.

[29] B. Fraser, M. Kaji, G. S. G. Arrieta, and S. Natsume. Mixed reality head mounted display, Sept. 19 2017. US Patent D797,735.

[30] E. Games. Blueprints visual scripting, https://docs.unrealengine.com/en-us/engine/blueprints, 2018.

[31] D. George, X. Xie, and G. K. Tam. 3d mesh segmentation via multi-branch 1d convolutional neural networks. *Graphical Models*, 96:1–10, 2018.

[32] J. J. Gibson. The theory of affordances. *Hilldale, USA*, 1(2), 1977.

[33] J. J. Gibson. *The ecological approach to visual perception: classic edition*. Psychology press, 2014.

[34] J. Gimeno, P. Morillo, J. M. Orduña, and M. Fernández. A new ar authoring tool using depth maps for industrial procedures. *Computers in Industry*, 64(9):1263–1271, 2013.

[35] A. V. González, S. Koh, K. Kapalo, R. Sottilare, P. Garrity, M. Billinghurst, and J. LaViola. A comparison of desktop and augmented reality scenario based training authoring tools. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 339–350. IEEE, 2019.

[36] P. Grabarczyk and M. Pokropski. Perception of affordances and experience of presence in virtual reality. *Avant*, 7(2):25–44, 2016.

[37] J. Grubert, E. Ofek, M. Pahud, and P. O. Kristensson. Back to the future: Revisiting mouse and keyboard interaction for hmd-based immersive analytics. *arXiv preprint arXiv:2009.02927*, 2020.

[38] S. Guven. *Authoring and presenting situated media in augmented and virtual reality*. Columbia University, 2006.

[39] E. Hadar, J. Shtok, B. Cohen, Y. Tzur, and L. Karlinsky. Hybrid remote expert-an emerging pattern ofindustrial remote support. 2017.

[40] N. Haferkamp, N. C. Kraemer, C. Linehan, and M. Schembri. Training disaster communication by means of serious games in virtual environments. *Entertainment Computing*, 2(2):81–88, 2011.

[41] N. Hagbi, R. Grasset, O. Bergig, M. Billinghurst, and J. El-Sana. In-place sketching for content authoring in augmented reality games. In *Virtual Reality Conference (VR), 2010 IEEE*, pages 91–94. IEEE, 2010.

[42] R. Hanson and A. Tacy. *GWT in action: easy Ajax with the Google Web toolkit*. Dreamtech Press, 2007.

[43] S. Hart, P. Dinh, and K. Hambuchen. The affordance template ros package for robot task programming. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 6227–6234. IEEE, 2015.

[44] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006.

[45] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[46] D. Hayatpur, S. Heo, H. Xia, W. Stuerzlinger, and D. Wigdor. Plane, ray, and point: Enabling precise spatial manipulations with shape constraints. In *Proceedings of the 32nd*

Annual ACM Symposium on User Interface Software and Technology, pages 1185–1195. ACM, 2019.

[47] S. Henderson and S. Feiner. Opportunistic tangible user interfaces for augmented reality. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):4–16, 2010.

[48] B. Herbert, B. Ens, A. Weerasinghe, M. Billinghurst, and G. Wigley. Design considerations for combining augmented reality with intelligent tutors. *Computers & Graphics*, 77:166–182, 2018.

[49] R. Holm, E. Stauder, R. Wagner, M. Priglinger, and J. Volkert. A combined immersive and desktop authoring tool for virtual environments. In *Proceedings IEEE Virtual Reality 2002*, pages 93–100. IEEE, 2002.

[50] C. E. Hughes, C. B. Stapleton, D. E. Hughes, and E. M. Smith. Mixed reality in education, entertainment, and training. *IEEE computer graphics and applications*, 25(6):24–30, 2005.

[51] C. E. Hughes, C. B. Stapleton, D. E. Hughes, and E. M. Smith. Mixed reality in education, entertainment, and training. *IEEE Computer Graphics and Applications*, 25(6):24–30, Nov 2005.

[52] K. Huo. Exploring advance interactions for augmented reality: From casual activities to in-situ 3d modeling. In *Proceedings of the Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 725–728. ACM, 2017.

[53] A. Ipsita, H. Li, R. Duan, Y. Cao, S. Chidambaram, M. Liu, and K. Ramani. Vrfromx: from scanned reality to interactive virtual experience with human-in-the-loop. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2021.

[54] J. Izquierdo-Domenech, J. Linares-Pellicer, and J. Orta-Lopez. Supporting interaction in augmented reality assisted industrial processes using a cnn-based semantic layer. In *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 27–32. IEEE, 2020.

[55] K. James, G. Humphrey, T. Vilis, B. Corrie, R. Baddour, and M. Goodale. "active" and "passive" learning of three-dimensional object structure within an immersive virtual reality environment. *Behavior Research Methods, Instruments, & Computers*, 34(3):383–390, 2002.

[56] L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor. Affordances in psychology, neuroscience, and robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(1):4–25, 2016.

[57] L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor. Affordances in psychology, neuroscience, and robotics: A survey. *IEEE Transactions on Cognitive and Developmental Systems*, 10(1):4–25, 2018.

[58] S. Jang, J. M. Vitale, R. W. Jyung, and J. B. Black. Direct manipulation is better than passive viewing for learning anatomy in a three-dimensional virtual reality environment. *Computers & Education*, 106:150–165, 2017.

[59] F. Jasche, S. Hoffmann, T. Ludwig, and V. Wulf. Comparison of different types of augmented reality visualizations for instructions. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2021.

[60] L. Jean and E. Wenger. *Situated learning: legitimate peripheral participation*. Cambridge University Press, England, 1991.

[61] H.-K. Jee, S. Lim, J. Youn, and J. Lee. An immersive authoring tool for augmented reality-based e-learning applications. In *Information Science and Applications (ICISA), 2011 International Conference on*, pages 1–5. IEEE, 2011.

[62] J. Jung, J. Hong, S. Park, and H. S. Yang. Smartphone as an augmented reality authoring tool via multi-touch based 3d interaction method. In *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 17–20. ACM, 2012.

[63] E. Kaas. The ngrain technology difference explained a whitepaper for technical evaluators of visualization and simulation technologies. *NGRAIN Corporation. Vancouver, Canada*.

[64] M. Kallmann. *Object Interaction in Real-Time Virtual Environments*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2001.

[65] M. Kallmann. Interaction with 3-d objects. *Handbook of Virtual Humans*, pages 303–322, 2004.

[66] M. Kallmann and D. Thalmann. A behavioral interface to simulate agent-object interactions in real time. In *Proceedings Computer Animation 1999*, pages 138–146. IEEE, 1999.

[67] M. Kallmann and D. Thalmann. Modeling objects for interaction tasks. In *Computer Animation and Simulation'98*, pages 73–86. Springer, 1999.

[68] M. Kallmann and D. Thalmann. Modeling behaviors of interactive objects for real-time virtual environments. *Journal of Visual Languages & Computing*, 13(2):177–195, 2002.

[69] M. Kapadia, J. Falk, F. Zünd, M. Marti, R. W. Sumner, and M. Gross. Computer-assisted authoring of interactive narratives. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, pages 85–92. ACM, 2015.

[70] H. Kato. Artoolkit: Library for vision-based augmented reality. , 101(652 (PRMU2001 222-232)):79–86, 2002.

[71] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top ar environment. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR 2000)*, pages 111–119. Ieee, 2000.

[72] H. Kim, S. Hong, J. Kim, T. Jang, W. Woo, S. Heo, and B. Lee. Realitybrush: an ar authoring system that captures and utilizes kinetic properties of everyday objects. *Multimedia Tools and Applications*, 80(20):31135–31158, 2021.

[73] M. Krichenbauer, G. Yamamoto, T. Taketom, C. Sandor, and H. Kato. Augmented reality versus virtual reality for 3d object manipulation. *IEEE transactions on visualization and computer graphics*, 24(2):1038–1048, 2017.

[74] J. J. LaViola Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[75] T. Lavric, E. Bricard, M. Preda, and T. Zaharia. An ar work instructions authoring tool for human-operated industrial assembly lines. In *2020 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pages 174–183. IEEE, 2020.

[76] T. Le, G. Bui, and Y. Duan. A multi-view recurrent neural network for 3d mesh segmentation. *Computers & Graphics*, 66:103–112, 2017.

[77] F. Lécuyer, V. Gouranton, A. Reuzeau, R. Gaugne, and B. Arnaldi. Authoring ar interaction by ar. 2019.

[78] I. Lediaeva and J. LaViola. Evaluation of body-referenced graphical menus in virtual environments. 2020.

[79] C. Lee, G. A. Rincon, G. Meyer, T. Höllerer, and D. A. Bowman. The effects of visual realism on search tasks in mixed reality simulation. *IEEE transactions on visualization and computer graphics*, 19(4):547–556, 2013.

[80] G. A. Lee, G. J. Kim, and C.-M. Park. Modeling virtual object behavior within virtual environment. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 41–48, 2002.

[81] G. A. Lee, C. Nelles, M. Billinghurst, and G. J. Kim. Immersive authoring of tangible augmented reality applications. In *Proceedings of the 3rd IEEE/ACM international Symposium on Mixed and Augmented Reality*, pages 172–181. IEEE Computer Society, 2004.

[82] Z. Li, J. Chan, J. Walton, H. Benko, D. Wigdor, and M. Glueck. Armstrong: An empirical examination of pointing at non-dominant arm-anchored uis in virtual reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–14, 2021.

[83] C. A. Liebert, M. A. Zayed, O. Aalami, J. Tran, and J. N. Lau. Novel use of google glass for procedural wireless vital sign monitoring. *Surgical innovation*, 23(4):366–373, 2016.

[84] M. Lucrecia, S. Cecilia, P. Patricia, and B. Sandra. Authorar: Authoring tool for building educational activities based on augmented reality. In *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, pages 503–507. IEEE, 2013.

[85] B. MacIntyre, M. Gandy, S. Dow, and J. D. Bolter. Dart: a toolkit for rapid design exploration of augmented reality experiences. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 197–206. ACM, 2004.

[86] B. MacIntyre and T. F. Smith. Thoughts on the future of webxr and the immersive web.

[87] J. B. Madsen, M. Tatzqern, C. B. Madsen, D. Schmalstieg, and D. Kalkofen. Temporal coherence strategies for augmented reality labeling. *IEEE transactions on visualization and computer graphics*, 22(4):1415–1423, 2016.

[88] J. Maloney, L. Burd, Y. Kafai, N. Rusk, B. Silverman, and M. Resnick. Scratch: a sneak preview [education]. In *Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing, 2004.*, pages 104–109. IEEE, 2004.

[89] P. Marion, M. Fallon, R. Deits, A. Valenzuela, C. Pérez D'Arpino, G. Izatt, L. Manuelli, M. Antone, H. Dai, T. Koolen, et al. Director: A user interface designed for robot operation with shared autonomy. *Journal of Field Robotics*, 34(2):262–280, 2017.

[90] S. Masnadi, J. J. LaViola Jr, J. Pavlasek, X. Zhu, K. Desingh, and O. C. Jenkins. A sketch-based system for human-guided constrained object manipulation, 2019.

[91] S. Masnadi, J. J. LaViola Jr, J. Pavlasek, X. Zhu, K. Desingh, and O. C. Jenkins. Sketching affordances for human-in-the-loop robotic manipulation tasks. *2nd Robot Teammates Operating in Dynamic, Unstructured Environments (RT-DUNE)*, 2019.

[92] S. Masnadi, K. Pfeil, J.-V. T. Sera-Josef, and J. LaViola. Effects of field of view on egocentric distance perception in virtual reality. In *CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2022.

[93] S. Masnadi, A. Vargas, B. Williamson, and J. LaViola. Affordit!: A tool for authoring object component behavior in virtual reality. *Proceedings of Graphics Interface 2020*, 2020.

[94] A. Matassa and R. Simeoni. Eliciting affordances for smart objects in iot era. In *International Internet of Things Summit*, pages 77–81. Springer, 2014.

[95] A. McEwen and H. Cassimally. *Designing the internet of things*. John Wiley & Sons, 2013.

[96] Microsoft. Design for mixed reality, https://docs.microsoft.com/en-us/windows/mixed-reality/design, 2018.

[97] Microsoft. Dynamics 365 guides, https://dynamics.microsoft.com/en-us/mixed-reality/guides/, 2019.

[98] Microsoft. Fluent design system, https://www.microsoft.com/design/fluent/, 2022.

[99] R. C. Mota, R. A. Roberto, and V. Teichrieb. [poster]authoring tools in augmented reality: An analysis and classification of content design tools. In *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, pages 164–167. IEEE, 2015.

[100] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[101] K. Murdock. *Autodesk Maya 2019 Basics Guide*. SDC Publications, 2018.

[102] M. Nebeling and M. Speicher. The trouble with augmented reality/virtual reality authoring tools.

[103] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, volume 11, pages 127–136, 2011.

[104] G. Ng, J. G. Shin, A. Plopski, C. Sandor, and D. Saakes. Situated game level editing in augmented reality. In *Proceedings of the Twelfth International Conference on Tangible, Embedded, and Embodied Interaction*, pages 409–418. ACM, 2018.

[105] D. T. Nguyen, B.-S. Hua, L.-F. Yu, and S.-K. Yeung. A robust 3d-2d interactive tool for scene segmentation and annotation. *IEEE transactions on visualization and computer graphics*, 24(12):3005–3018, 2017.

[106] D. C. Niehorster, L. Li, and M. Lappe. The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research. *i-Perception*, 8(3):2041669517708205, 2017.

[107] G. Norton, M. Taylor, T. Stewart, G. Blackburn, A. Jinks, B. Razdar, P. Holmes, and E. Marastoni. Designing, developing and implementing a software tool for scenario based learning. *Australasian Journal of Educational Technology*, 28(7), 2012.

[108] B. Nuernberger, K.-C. Lien, T. Höllerer, and M. Turk. Interpreting 2d gesture annotations in 3d augmented reality. In *3D User Interfaces (3DUI), 2016 IEEE Symposium on*, pages 149–158. IEEE, 2016.

[109] J.-Y. Oh and W. Stuerzlinger. Moving objects with 2d input devices in cad systems and desktop virtual environments. In *Proceedings of Graphics Interface 2005*, pages 195–202. Canadian Human-Computer Communications Society, 2005.

[110] J. T. Paige, V. Kozmenko, T. Yang, R. P. Gururaja, C. W. Hilton, I. Cohn Jr, and S. W. Chauvin. High-fidelity, simulation-based, interdisciplinary operating room team training at the point of care. *Surgery*, 145(2):138–146, 2009.

[111] B. Pellens, O. De Troyer, W. Bille, and F. Kleinermann. Conceptual modeling of object behavior in a virtual environment. In *Proceedings of virtual concept*, pages 93–94, 2005.

[112] E. Peter Errington. Mission possible: Using near-world scenarios to prepare graduates for the professions. *International Journal of Teaching and Learning in Higher Education*, 23:84–91, 2011.

[113] K. Pfeil, S. Masnadi, J. Belga, J.-V. T. Sera-Josef, and J. LaViola. Distance perception with a video see-through head-mounted display. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–9, 2021.

[114] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, volume 17, pages 41–52. Wiley Online Library, 1998.

[115] I. Poupyrev, D. S. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani. Tiles: A mixed reality authoring interface. In *Interact*, volume 1, pages 334–341, 2001.

[116] X. Qian, F. He, X. Hu, T. Wang, A. Ipsita, and K. Ramani. Scalar: Authoring semantically adaptive augmented reality experiences in virtual reality. In *CHI Conference on Human Factors in Computing Systems*, pages 1–18, 2022.

[117] I. Radu. Ar spot: An augmented-reality programming environment for children. *URL http://ael. gatech. edu/lab/research/arspot*, 2020.

[118] S. Rajaram and M. Nebeling. Paper trail: An immersive authoring system for augmented reality instructional experiences. In *CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2022.

[119] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al. Scratch: programming for all. *Communications of the ACM*, 52(11):60–67, 2009.

[120] E. Rosenbaum, E. Klopfer, and J. Perry. On location learning: Authentic applied science with networked augmented realities. *Journal of Science Education and Technology*, 16(1):31–45, Feb 2007.

[121] D. Rumiński and K. Walczak. Creation of interactive ar content on mobile devices. In *International Conference on Business Information Systems*, pages 258–269. Springer, 2013.

[122] J. T. Samosky, D. A. Nelson, B. Wang, R. Bregman, A. Hosmer, B. Mikulis, and R. Weaver. Bodyexplorerar: enhancing a mannequin medical simulator with sensing and projective aug-

mented reality for exploring dynamic anatomy and physiology. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*, pages 263–270. ACM, 2012.

[123] C. Sandor. *A software toolkit and authoring tools for user interfaces in ubiquitous augmented reality*. PhD thesis, Technische Universität München, 2005.

[124] R. Schaer, F. Salamin, O. Alfonso, J. del Toro, M. Atzori, H. Müller, and A. Widmer. Live ecg readings using google glass in emergency situations. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 315–318. IEEE, 2015.

[125] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavári, L. M. Encarnaçao, M. Gervautz, and W. Purgathofer. The studierstube augmented reality project. *Presence: Teleoperators & Virtual Environments*, 11(1):33–54, 2002.

[126] J. Seddon, B. McDonald, and A. Schmidt. Ict-supported, scenario-based learning in preclinical veterinary science education: Quantifying learning outcomes and facilitating the novice-expert transition. *Australasian Journal of Educational Technology*, 28(2), 2012.

[127] H. Seichter, J. Looser, and M. Billinghurst. Composar: An intuitive tool for authoring ar applications. In *Proceedings of the 7th IEEE/ACM international symposium on mixed and augmented reality*, pages 177–178. IEEE Computer Society, 2008.

[128] A. W. Services. Amazon sumerian visual scripting, https://aws.amazon.com/sumerian/, 2018.

[129] A. Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008.

[130] T. Shao, W. Li, K. Zhou, W. Xu, B. Guo, and N. J. Mitra. Interpreting concept sketches. *ACM Transactions on Graphics (TOG)*, 32(4):56, 2013.

[131] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1997.

[132] O. Sidi, O. van Kaick, Y. Kleiman, H. Zhang, and D. Cohen-Or. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–10, 2011.

[133] G. Sketch. Gravity sketch vr. *Gravity Sketch,[Online]. Available: https://www.gravitysketch.com/.[Accessed 23 November 2019]*, 2019.

[134] G. Smith and W. Stürzlinger. Integration of constraints into a vr environment. In *VRIC'01: Proc. of the Virtual Reality Int'l Conf*, pages 103–110, 2001.

[135] R. A. Sottilare, K. W. Brawner, B. S. Goldberg, and H. K. Holden. The generalized intelligent framework for tutoring (gift), 2012.

[136] F. Spini, E. Marino, M. D'Antimi, E. Carra, and A. Paoluzzi. Web 3d indoor authoring and vr exploration via texture baking service. In *Proceedings of the 21st International Conference on Web3D Technology*, pages 151–154. ACM, 2016.

[137] A. Steed and M. Slater. A dataflow representation for defining behaviours within virtual environments. In *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*, pages 163–167. IEEE, 1996.

[138] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *CHI*, volume 95, pages 265–272, 1995.

[139] T. Taketomi, H. Uchiyama, and S. Ikeda. Visual slam algorithms: a survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, 2017.

[140] U. Technologies. Visual scripting in unity3d roadmap, https://unity3d.com/unity/roadmap, 2018.

[141] D. Thalmann. Geometric issues of object manipulation in task animation and virtual reality. *Expanding the Frontiers of Visual Analytics and Visualization*, pages 339–362, 2012.

[142] The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.0 edition, 2019.

[143] Ultraleap. Digital worlds that feel human, https://www.ultraleap.com/, 2022.

[144] Unity3D. The world's leading platform for real-time content creation. `https://unity.com/`, 2021. [Online; accessed 10-October-2022].

[145] A. Uthor. 3d scene manipulation with 2d devices and constraints. In *Proceedings of graphics interface*, pages 135–142. Citeseer, 2001.

[146] J. Valentin, V. Vineet, M.-M. Cheng, D. Kim, J. Shotton, P. Kohli, M. Nießner, A. Criminisi, S. Izadi, and P. Torr. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics (TOG)*, 34(5):154, 2015.

[147] A. N. Vargas González, K. Kapalo, S. L. Koh, and J. J. LaViola. Exploring the virtuality continuum for complex rule-set education in the context of soccer rule comprehension. *Multimodal Technologies and Interaction*, 1(4):30, 2017.

[148] T. Wang, X. Qian, F. He, X. Hu, Y. Cao, and K. Ramani. Gesturar: An authoring system for creating freehand interactive augmented reality applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 552–567, 2021.

[149] Y. Wang, S. Asafi, O. Van Kaick, H. Zhang, D. Cohen-Or, and B. Chen. Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG)*, 31(6):1–10, 2012.

[150] Y. Wang, T. Langlotz, M. Billinghurst, and T. Bell. An authoring tool for mobile phone ar environments. In *Proceedings of New Zealand Computer Science Research Student Conference*, volume 9, pages 1–4, 2009.

[151] G. Westerfield, A. Mitrovic, and M. Billinghurst. Intelligent augmented reality training for motherboard assembly. *International Journal of Artificial Intelligence in Education*, 25(1):157–172, Mar 2015.

[152] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. J. Leonard, and J. McDonald. Real-time large-scale dense rgb-d slam with volumetric fusion. *The International Journal of Robotics Research*, 34(4-5):598–626, 2015.

[153] C. Winterbottom and E. Blake. Designing a vr interaction authoring tool using constructivist practices. In *Proceedings of the 3rd international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 67–71. ACM, 2004.

[154] M. K. Young, G. B. Gaylor, S. M. Andrus, and B. Bodenheimer. A comparison of two cost-differentiated virtual reality systems for perception and action tasks. In *Proceedings of the ACM Symposium on Applied Perception*, pages 83–90, 2014.

[155] J. Yu, J. Jeon, J. Park, G. Park, H.-i. Kim, and W. Woo. Geometry-aware interactive ar authoring using a smartphone in a wearable ar environment. In *International Conference on Distributed, Ambient, and Pervasive Interactions*, pages 416–424. Springer, 2017.

[156] Zappar. The world's leading augmented reality platform and creative studio, https://www.zappar.com/, 2022.

[157] J. Zauner and M. Haller. Authoring of mixed reality applications including multi-marker calibration for mobile devices. In *Proceedings of the Tenth Eurographics Conference on Virtual Environments*, EGVE'04, pages 87–90, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

[158] J. Zauner, M. Haller, A. Brandl, and W. Hartman. Authoring of a mixed reality assembly instructor for hierarchical structures. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 237–246, Oct 2003.

[159] R. Zeleznik, K. Herndon, and J. Hughes. Sketch: An interface for sketching 3d scenes. In *Computer graphics proceedings, annual conference series*, pages 163–170. Association for Computing Machinery SIGGRAPH, 1996.

[160] Z. Zhang, Z. Li, M. Han, Z. Su, W. Li, and Z. Pan. An augmented reality-based multimedia environment for experimental education. *Multimedia Tools and Applications*, 80(1):575–590, 2021.

[161] J. Zhu, S. Ong, and A. Nee. An authorable context-aware augmented reality system to assist the maintenance technicians. *The International Journal of Advanced Manufacturing Technology*, 66(9-12):1699–1714, 2013.