INVESTIGATING THE EFFICIENCY OF AUTHORING INTERACTIONS FOR AUGMENTED REALITY EXPERIENCES FOR DESGINERS

A Dissertation
Presented to
The Academic Faculty

by

Karan Jain

In Partial Fulfillment
of the Requirements for the Degree
Master of Industrial Design in the
School of Industrial Design

Georgia Institute of Technology
December 2020

INVESTIGATING THE EFFICIENCY OF AUTHORING INTERACTIONS FOR AUGMENTED REALITY EXPERIENCES FOR DESIGNERS

Approved by:

Dr. Young Mi Choi, Advisor
School of Industrial Design
Georgia Institute of Technology

Tim Purdy
School of Industrial Design
Georgia Institute of Technology

Dr. Sang-Won Leigh
School of Industrial Design
Georgia Institute of Technology

Date Approved: November 11, 2020

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Young Mi Choi, for giving me the opportunity to work on her research and being a great mentor during my time in graduate school. Her constant encouragement to perform better and pay attention to detail has allowed me to produce robust work for this thesis. I would like to thank her for constantly checking up on my progress for this thesis and providing the right guidance to deliver a good outcome.

I would also like to thank my committee members, Tim Purdy and Dr. Sang-Won Leigh, for always being there to answer any questions I had and providing detailed feedback to further improve my work.

Lastly, I would like to say a big thank you to my parents and my sister. Their moral support and encouragement have always motivated me to work harder and achieve the goals I desire.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AR	Augmented Reality
ID	Industrial Design
CAD	Computer-Aided Design
GUI	Graphical User Interface
TAR	Tangible Augmented Reality
SDK	Software Development Kit
UX	User Experience
UI	User Interface
SUS	System Usability Scale

SUMMARY

Recent advances in augmented reality (AR) have provided an opportunity for this technology to be used in the industrial design and development process, especially, for product visualization and representing product concepts in AR for usability testing. However, the adoption of this technology in the industrial design process is slow-moving due to the complex development process of AR experiences. Currently, AR authoring tools require the user to program interactions for their applications, which makes it challenging for designers who may not be skilled in programming to rapidly develop interactions. Further, there is not much literature on the design and user experience (UX) of graphical user interface (GUI) based tools for authoring AR interactions for designers or non-programmers. This thesis investigated how authoring interactions for AR experiences can be made more efficient for designers. After reviewing current AR tools and projects, two UX architectures for authoring AR interactions with a GUI were put together. These architectures were called Component-Based UX Architecture and Event-Based UX Architecture. An interactive prototype was developed for each of the architecture and users were asked to author interactions for an AR representation of a table lamp and a toy car. A remote usability study was conducted to evaluate the two prototypes with 22 participants and it was found that the event-based UX architecture is significantly more efficient and user friendly than the component-based UX architecture in authoring interactions for AR experiences. The implication of the results in the design of AR authoring tools for designers has been further discussed in this paper.

CHAPTER 1. INTRODUCTION

Several researchers have investigated how enhancing the physical world with digital overlays can make the industrial design (ID) and development process more effective and efficient. Recent advances in augmented reality (AR) software and hardware have made it possible for this technology to be more widely used by developers and designers. However, the adoption of this technology in the industrial design process is slow-moving due to the complex development process of AR experiences. Most designers are not trained in software programming, hence, they need to be able to develop AR experiences quickly with AR authoring tools. Authoring tools allow for developing programs or digital content without any programming, thereby reducing the time and technical expertise that may be required to develop a software application (Mota, Roberto, & Teichrieb, 2015). Developing an AR experience can be broken down into 3 parts: designing 2D and 3D content, developing interactions for objects, and exporting the application to a device that support AR viewing (Jain & Choi, 2019). Currently, designers have the necessary skills and availability of tools for designing content for AR. However, there are not many tools that allow designers to author interactions for AR experiences. "Interaction development" or "interaction authoring" for this study is defined as the experience of developing an interaction where a user input on one object triggers an output on another object or a user input on one object triggers a sequence of responses from the same object. A review of AR authoring tools showed that even though great strides have been made in the development of authoring tools for non-programmers, these tools are still in their infancy in providing an optimal user experience (UX) for authoring AR interactions with a GUI. It was determined that there is more room to explore and validate an efficient and user friendly AR interaction authoring experience for designers, especially industrial designers who want to use this technology in the ID process.

This thesis set out to investigate an efficient experience for authoring AR interactions for non-programmers, where non-programmers were defined as people who have no education in computer science or related field and do not have any work experience with programming. More specifically, this study will answer the research question: Is a component-based UX architecture more efficient than an event-based UX architecture for authoring interaction for AR for designers? The next part of this paper highlights the benefits of AR in the ID process and provides a review of current AR authoring tools. Then, the prototypes developed for this study are explained, and finally, the design of the study and results are discussed.

CHAPTER 2. BACKGROUND

2.1 Benefits of AR in the Industrial Design Process

Augmented Reality (AR) is a technology where computer-generated virtual objects coexist in the same space as the real world (Krevelen, 2007). An AR system consists of a display that can combine real and virtual images, a computer system that can generate interactive graphics that respond to user input in real time, and a tracking system that can find the position of the user's viewpoint and enable the virtual images to appear fixed in the real world (Billinghurst, Clark, & Lee, 2014). An early exploration of augmented reality (AR) in the design process was done for design visualization of computer-aided design (CAD) models for design development and collaboration. CAD is the use of computer systems to assist in the creation, modification, analysis or optimization of a design (Sarcar, Rao, & Naryan, 2008). Dunston et. al. developed an AR CAD system that allowed for visualization of 3D models in the real world from any viewpoint (Phillip S. Dunston, Xiangyu Wang, Mark Billinghurst, 2003). Viewing CAD models in a real world context provided a more natural way of looking at content than would be afforded by visualization systems that have a more constrained means of navigation, which facilitated design collaboration and ensured accurate shared understanding of the design models. AR has also been explored for assembly feature design by Pang et. al. (Pang, Nee, Ong, Yuan, & Youcef-Toumi, 2006). In assembly design, products are divided into two categories. The first category entails parts that are fixed and remain the same for a wide range of products, for example, a car manufacturer may reuse the chassis of a particular car model for other models too. The second category entails parts that are expected to change. Designers and engineers design these variable parts and add them to the fixed base parts that remain unchanged. Traditionally, physical prototyping was used to design and test the assembling of parts. However, physical prototyping is very time-consuming, expensive, and difficult to modify. Pang et. al developed a system that combined physical and virtual prototyping, where the fixed parts were physical and the variable parts were 3D models viewed through an AR medium over the fixed parts. This system allowed for quick and cost effective design iteration of the variable parts. Besides visualization and assembly, AR has also been explored for prototyping of industrial products for usability testing. An example study is Augmented Foam by Lee and Park (W. Lee & Park, 2005). They tested the usefulness of combining AR with physical representations, also known as tangible augmented reality (TAR), through a design example of a cleaning robot and found that their system elevated the feeling of immersion through multi-sense stimuli and spatial interactions. More recently, Choi explored the usability assessment of a space heater with AR and TAR design representation of the heater (Choi, 2019).

The above mentioned studies have shown how AR can be beneficial in the design and usability process by providing an accurate and contextual visualization of prototypes. Further, Choi et. al mention that for some cases, AR design representations can provide a valid replacement for the usability assessment of physical functioning prototypes that take longer to develop, are more expensive, and can only be available later in the design process. This can be extremely beneficial for designers since they can gather accurate feedback from users much earlier in the design process (Purdy & Choi, 2014). AR also provides an opportunity to prototype and test experiences that entail a physical and digital experience more efficiently. For example, if a product with a knob and a digital interface is represented in AR, the user can interact with the 3D model of the knob and see the response immediately on the interface. Even though the user may not get an understanding of the physical interaction with the knob, they will be able to understand the features and functionalities of the product.

2.2 Review of AR Authoring Tools

The review of AR authoring tools has been divided into two categories: authoring with programming/coding and authoring through a graphical user interface (GUI). Authoring interactions through programming entails writing code and using code libraries, which provides greater control over the development of the application as compared to tools that hide the low-level tasks and provide a GUI to author interactions. However, programming tools cannot be used by designers or, even if they are used by designers, they may not be very proficient in using them (Jain & Choi, 2020). Designers need to develop programs through visual-programming, where they edit graphical elements on an interface as compared to writing code or using code libraries to develop applications.

2.2.1 Authoring Interactions through Programming

Programming tools for developing AR experiences can be further classified into low-level libraries and high-level frameworks. Low-level libraries require the user to have strong programming/coding ability and generally require the user to develop programs in C++. In the context of developing AR applications, the main purpose of low-level tools is to provide core functionalities like image marker tracking, spatial registration of objects, and 3D rendering. They provide a high degree of performance and flexibility since they allow users to define every aspect of the program. However, high development time is the drawback in using low-level tools for authoring interactions (Hampshire, Seichter, Grasset, & Billinghurst, 2006). Some examples of low-level libraries that are commonly used for developing AR applications include ARToolkit, Wikitude SDK, and Vuforia SDK.

High-level frameworks for developing AR applications combine low-level libraries that provide core AR functionalities with other code libraries that provide functionalities for authoring media and interactions. This allow developers to worry about interactions between virtual objects rather than low-level tasks like tracking and object rendering. An example of a wrapper for AR development is osgART, which is a cross platform development library that combines computer vision based tracking libraries with the 3D graphics library OpenSceneGraph (Looser, Grasset, Seichter, & Billinghurst, 2006).

Interactions for AR can also be developed with a scripting language in a plug-in or stand-alone authoring tool. A scripting language is a high-level programming language, which does not require compilation steps and are interpreted at run time. They are used to develop specific functions of an application and software applications usually have their own limited-purpose scripting languages. Scripting languages are commonly used to add interactivity, visualization to applications, and develop communication between applications (NamanKed, 2018). Designers may be more familiar with scripting languages as compared to low-level programming languages. Plug-in tools are small applications that are developed to be integrated into preexisting applications. In the context of AR authoring tools, plug-in tools provide AR functionalities in a software that designers may be familiar with (Hampshire et al., 2006) . These tools allow programmers or people who are less proficient with programming to develop interactions through scripting languages. DART (Designer's AR Toolkit) is an AR authoring tool that was developed as a plug in for

Macromedia Director. DART was developed to aid early and often testing of AR experiential designs and it supported early design activities like the transition from story boards to working experiences (MacIntyre, Gandy, Dow, & Bolter, 2005). The target user was a skilled multimedia designer who was familiar with Macromedia Director. It was built with the assumption that “designers can and will venture out “into the code” and they will continue to use their existing tools (e.g. Photoshop, Maya) for content creation.”. DART was built on top of Director which included an object-oriented scripting language called Lingo. Since, all of the behaviors in DART were written in Lingo, it allowed developers to modify standard behaviors as well as write their own scripts from scratch. Designers could also add scripts to objects through a drag and drop interaction.

Unity is another stand-alone platform that allows users to author AR applications through a GUI and a scripting language (“Unity Real-Time Development Platform | 3D, 2D VR & AR Visualizations,” 2019). The Unity interface is a GUI, which allows users to use visual programming paradigms to author media content, like drag and drop objects into the Unity scene, click buttons, and drag sliders in the inspector window to edit appearance or behaviors of virtual components. However, authoring complex interaction behaviors in Unity is not possible through visual programming but by scripting in the C# language. Several C# libraries, like LeanTouch (“Lean Touch - Asset Store,” 2019) that are developed by third party developers, can also be found on the Unity Asset Store to author interaction behaviors for virtual components. Using such libraries does not require the user to write any scripts, however, the user still needs to have some understanding of object oriented scripting in order to understand how to use such libraries and slightly tweak the scripts to make them more applicable for their use case. Further, the workflow of Unity entails setting up the Unity scene and writing scripts for object behaviors in the desktop view and then exporting the application to the respective platform for testing the build or testing on the device through the Unity Editor. This flow makes the process of authoring and testing interactions slow since the user does not get immediate feedback to an action they perform. Amazon Sumerian is a tool that allows designers to build quick AR web experiences by dragging and dropping assets, like a fbx 3D model, from their desktop into the scene view. Since this is a web based tool, complex interactions can be developed with HTML, CSS, and Javascript ("Amazon Sumerian Overview," 2019).

2.2.2 Authoring Interactions with GUI-Based Tools

Stand-alone tools provide a complete software to build end-to-end applications, where low-level libraries provide all the core AR functionalities and a GUI allows for authoring interactions and media content. Designers can use these tools for developing AR experiences more quickly as compared to programming interactions. However, the drawback with these tools is that users can only work with the predefined functionalities and object behaviors that are offered by the software and cannot develop new interactions. Developing more interactions generally entails coding them in a scripting language or using low-level code libraries. A common theme that emerged during the investigation of stand-alone tools was that these tools were developed with a component-based model or framework. Developing a software with a component-based approach means that the software is designed in a way where parts of the software can be reused. These reusable parts of the software are called components, or more formally, a binary code than can be reused in a software is called a component (Qureshi & Hussain, 2008). A component is designed to serve a specific purpose and it has specific properties. For example, a text box is a component of a software that has the property of writing text to it and displaying the text. A text box can be reused over and over again in an application. Abawi et. al. proposed a component-based authoring environment for creating multi-media rich mixed reality experiences so that end users can rapidly create AR experiences by reusing the components in the application. Their architecture also allowed developers to easily develop and integrate more components with the system (Abawi, Dörner, & Grimm, 2004).

AMIRE was a project whose goal was to “motivate people without programming skills (e.g. designers, artists, domain specific experts, etc.) to author mixed reality applications instead of coding them.” (Haller, Stauder, & Zauner, 2005). The AMIRE authoring tools was based on a component-based framework, which supported an authoring environment for PDAs and Tablet PCs. CATOMIR was a visual programming interface that was built on top of AMIRE. The interface followed a three step approach where users had to find the right components for the application, tweak the components respectively, and connect the components through a drag-and-drop interaction to define logical behaviors. Figure 1 shows this 3 step approach in CATOMIR, where the third image represents the visualization of a component network in the application. The drawback of CATOMIR was that AR applications could only have the functionality supported by the components, and it was difficult to add new components (Billinghurst et al., 2014). This limited the type and number of interactions that users could develop for virtual components. Similar to AMIRE, Wikitude Studio (“Wikitude Studio-Augmented Reality Creation & Management Tool,” 2019), which is built with the Wikitude SDK, is a more complete web based authoring tool that allows users to create mobile AR content and deploy either onto the Wikitude AR browser app or even create a custom mobile app (Billinghurst et al., 2014). Wikitude's studio editor supports simple drag-and-drop interface, an intuitive workflow, easy testing, and fast publishing. It allows a non-programmer to build experiences where 3D models can be viewed in AR and simple behaviors like rotation and scaling can be added to the models; however, the application does not allow for developing interactions between objects for AR.

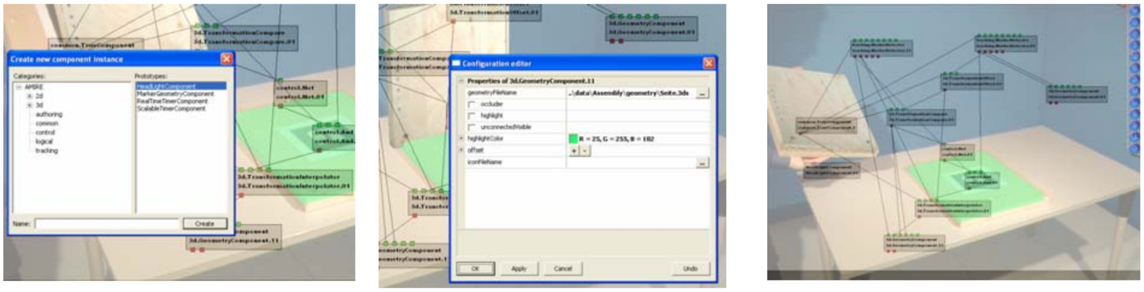


Figure 1 - Interaction authoring approach in CATOMIR

A tool that focused on authoring tangible interactions for AR experiences, without requiring any programming, was developed by (G. A. Lee & Kim, 2009). The goal of their study was to evaluate a tool for authoring AR content behavior and interactions from within the AR interface, which they called "immersive authoring". The development environment and the execution environment was the same in "immersive authoring" and the authoring environment provided the full experience of the building contents by itself. Just like AMIRE, Lee et. al. proposed a component based application model for TAR applications. User interactions involved browsing through a list of components and properties for those components with a physical object that acted as a pointer. The user identified properties of different components and then connected the properties to author interactions. Such a component based framework allowed for easy and direct manipulation of 3D virtual objects with immersive authoring AR environments. A drawback of this tool was that users could not author complex interaction behaviors because the component based framework allowed for matching the preexisting properties and did not allow the user to create new properties.

Another common framework that was found to be used for developing scenarios or behaviors in AR authoring tools was the event-driven architecture. Park. developed a rapid prototyping framework for an AR applications called AR-Room. He used an event-based action model for handling interactions in his framework (Park, 2011). An event can be described as a change in state of the system, where, when an event occurs, the associated action gets activated. An event for an AR application can be an input from the user, an input from an external device, an event from a marker detector, or even an internal change in the system. Typical examples of actions can be changes in attributes of an object like change in position, in scale, in speed, or even attributes like an animation starts playing. An event-driven architecture allows for developing and easy handling of state changes for an application. Such a framework was used in ComposAR, which is a GUI based tool for authoring AR applications for users with non or little programming knowledge (Seichter, Looser, & Billinghurst, 2017). The intermediate level of this system implemented an action-reaction model where input from external sensors could be used to trigger predefined actions. It supported both visual programming (drag and drop interface) and interpretive scripting. The GUI consisted of a tree layout, where a node in the tree structure can be activated with a click, highlighting the respective 3D object in the scene and showing manipulation handles. Adobe Aero was an AR authoring tool where the event-driven architecture was visualized in the front end of the application for authoring interactions ("Create augmented reality experiences | Adobe Aero," 2019). The workflow of Aero allows users to anchor virtual content to a physical space and then resize and reposition the objects. Further, users can set a triggers, such as touch, swipe, etc., that would set off an action or response in the object. The actions could only be selected from a list of predefined behaviors like jump, rotate, etc. Figure 2 shows two behaviors that are developed for the object in the AR view with an event-driven approach. The first behavior has a tap trigger that sets off a bounce and spin action. The second behavior has an enter proximity trigger that sets off a move action.

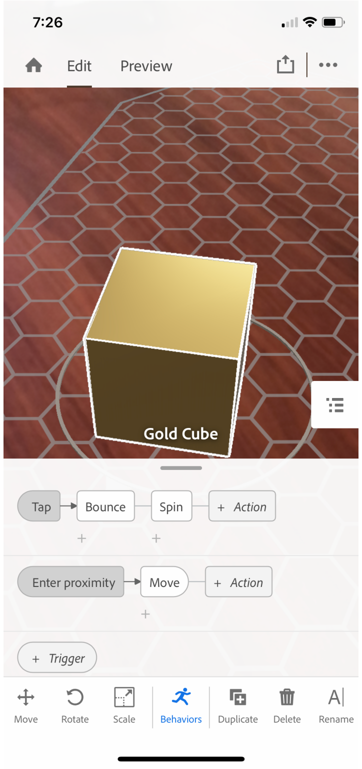


Figure 2 – Interaction authoring approach in Aero

CHAPTER 3. PROTOTYPE DEVELOPMENT

The Gulf of Execution and Evaluation was a theoretical framework, developed by Donald Norman, to help understand what the user is doing when using a physical machine or digital system (Norman, 1986). In his framework, Norman highlights that the person's goal is expressed in psychological terms and the system's mechanisms and states are expressed in physical terms. This discrepancy in what the user is thinking and what the system is doing needs to be bridged by the design of the system. The novelty of interaction authoring for AR, especially for non-programmers, makes it challenging for users to understand and predict what the system is doing and how it is supposed to work. In order to make the process of authoring interactions efficient, it needs to be ensured that the user's mental model of the system can align with how the system is actually functioning. As a first step in developing an optimal user experience (UX) for AR interaction authoring, a framework for the information architecture or UX architecture was required to be put together and then validated for how easily it can be understood by users. Two UX architecture frameworks were put together, which were inspired by the system architecture and functionalities of the authoring tools that were reviewed for this study. These UX architectures were named: Component-Based UX Architecture and Event-Based UX Architecture. An interactive prototype of an AR authoring tool was developed for each UX architecture, so that the architectures could be compared for their efficiency and usability. The next section discusses the details, advantages, and disadvantages of the two architectures.

3.1 Details on UX Architectures

CATOMIRE, AMIRE, and Immersive Authoring were three stand-alone AR authoring tools that were developed with a component-based development framework. The interesting thing about these tools was that even the way users authored interactions was influenced by the component-based approach. To author interactions, users generally followed a three step approach where they first identified a component, developed a property for the component or identified a property they wanted to use, and then linked that property to the property of another component. Components in an AR scene can be 3D models that have been imported into the system, text boxes, buttons, etc. Such an interaction flow requires the users to understand how the system has been developed, where every object in the AR scene is a component that has properties and object behaviors or interactions can be developed by linking properties of different components together. Figure 3 visually represents this component-based approach to developing interactions. It was assumed that such a framework for developing interactions is optimal because once users understand the simple concept of components, properties, and linking properties, they will be able to easily author interactions and will not get lost in the authoring process. In this study, such a framework for authoring AR interactions was called the Component-Based UX Architecture.

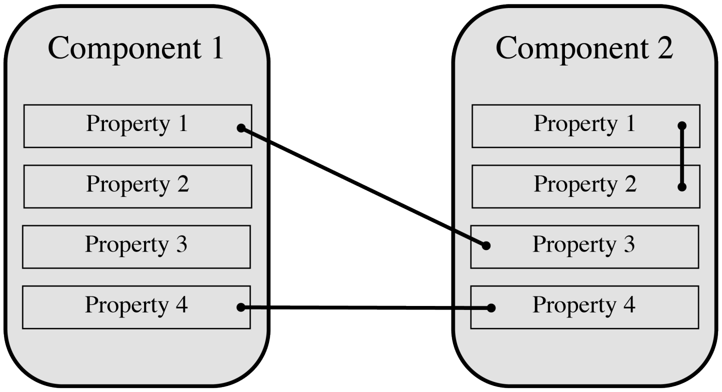


Figure 3 - Component-Based UX Architecture

Tools like AR-Room, ComposAR, and Adobe Aero discussed the concept of an event-based model. In simple terms, an event-based model is a framework where a change of state in one part of the system causes a change in another part of the system. Since such a paradigm can be easy to understand at a high-level, it has not only been used to develop visual programming tools for children but also been explored for creating business model frameworks (Michelson, 2006). Scratch was a visual programming tool that was developed to teach children from the ages of 8 – 16 programming basics and encourage then to tinker with and create digital media experiences using visual programming (Resnick et al., 2009). Different functionalities were represented through blocks, which could be linked to one another to create an experience. This process of linking blocks was based on the concept of events taking place one after another, which forms the basis of using an event-based model for authoring interactions. Using an event-based model for authoring AR interactions would entail linking an action/response to a trigger/input. Hence, this paradigm is also commonly known as trigger-action programming and it takes the form of "if trigger, then action" (Ury, McManus, Ho, & Littman, 2014). Trigger-action programming is also used in the website, IFTT, for allowing non-programmers to connect smart home devices to a network and develop home experiences with these devices. Since the event-based model is commonly used in visual programming applications, it was assumed that users will be able to understand this concept of linking event for authoring interactions for AR. For this study, a framework called Event-Based UX Architecture was developed for AR interaction authoring process, which is visually represented in Figure 4. In an event-based UX architecture, users will first create an event for any interaction they want to develop. Then, they will add a trigger or input to the event and link that to the action that takes place once the trigger is detected. An event can only have one trigger but it can have multiple actions that are linked to one another and get set off one after another.

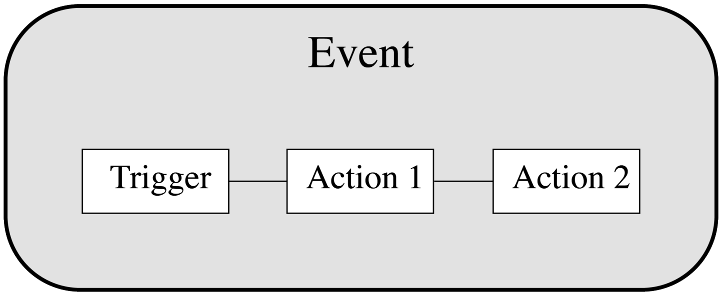


Figure 4 - Event-Based UX Architecture

3.2 Detail on Study Prototypes

An interactive prototype was developed for each of the UX architecture to test the architecture for its efficiency in allowing designers to author interactions for AR experiences. The goal for these prototypes was to design an easy to understand user interface (UI) and an effective process for allowing users to author interactions for specific scenarios. The prototype that was developed to test the component-based UX architecture was called the component-based prototype and the prototype that was developed for the event-based UX architecture was called the event-based prototype.

3.2.1 Usability Testing Scenarios

The design process for developing the prototypes started with creating scenarios for which users will author interactions. Since this study was motivated by improving the process for industrial designers to develop AR representations of physical products, the scenarios for which the study prototypes would be tested entailed developing interactions for physical products that were represented in AR. The next consideration for developing the scenarios was how “develop interactions” was defined for this study. As discussed earlier, “develop interactions” for this study was defined as the experience of developing an interaction where a user input on one object triggers an output on another object or a user input on one object triggers a sequence of responses from the same object. The first scenario that was developed for the study entailed authoring interactions for a table lamp that had a physical knob as its point of interaction and would satisfy the condition of developing an interaction where a user input on one object triggered a response on another object. As represented in Figure 5, the knob on the lamp had to be translated up to turn the light on and the knob had to be rotated to change the color of the light. A 3D model of a table lamp was selected to be represented in AR for the study because a lamp is an object that is familiar to most people and the working of the knob was also considered to be straightforward. It was ensured that no effort goes into understanding the scenario and hence, a simple object that works in a straightforward way was selected.

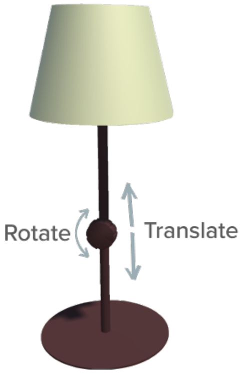


Figure 5 - 3D model of a table lamp

The second scenario entailed developing interactions for a toy car, where the interactions entailed holding the finger down on the car to make it move and then tapping on it again to make it rotate, continue to move in the direction it rotated in, and then come to a stop after 10 seconds. Similar to the table lamp, it was assumed that a toy car is an object people are familiar with and the car allowed for developing interactions where a single user input could set off multiple actions from the same object. Figure 6 shows a visual representation of the 3D model of the toy car that was represented in AR for the study.

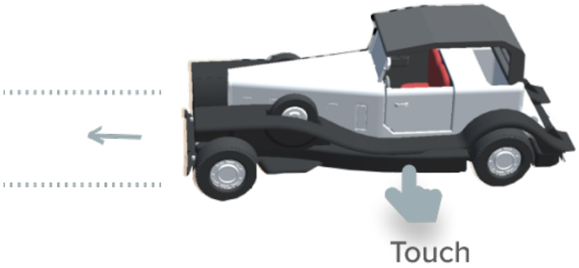


Figure 6 - 3D model of a toy car

3.2.2 Low-Fidelity Digital Prototype and Testing

Lee et. al. discuss the concept of what you see is what you get output (WYSIWYG) in their immersive authoring project (G. A. Lee & Kim, 2009). Users can evaluate their actions more quickly if they author interactions in the augmented view and get immediate feedback to their action. This makes it clear that an efficient user experience for authoring AR interactions should happen in a way where the authoring and testing platform is the same. Hence, for this study, the prototypes were designed to run on an Ipad as a mobile application and not run on a desktop as a desktop application. This is because the users would be able to author interactions in the augmented view and get immediate feedback for their actions instead of authoring on the desktop and exporting their project to a mobile device for testing.

After identifying the scenarios for which the prototypes will be designed for, an iterative design process was followed to develop the interfaces. Based on the tasks that users would have to complete (discussed in detail in the next section) a task analysis was put together and then wireframe sketches were made to identify the flow for the prototypes.

Next, a grayscale digital prototype was developed with more defined interface elements and information text. The purpose of this first digital prototype was to test the interface and UX architecture for their effectiveness in completing the tasks. This would entail gathering qualitative feedback on the UI elements and the communication of instructions. Figure 7 shows example screens the grayscale digital for the component-based prototype and the event-based prototype.

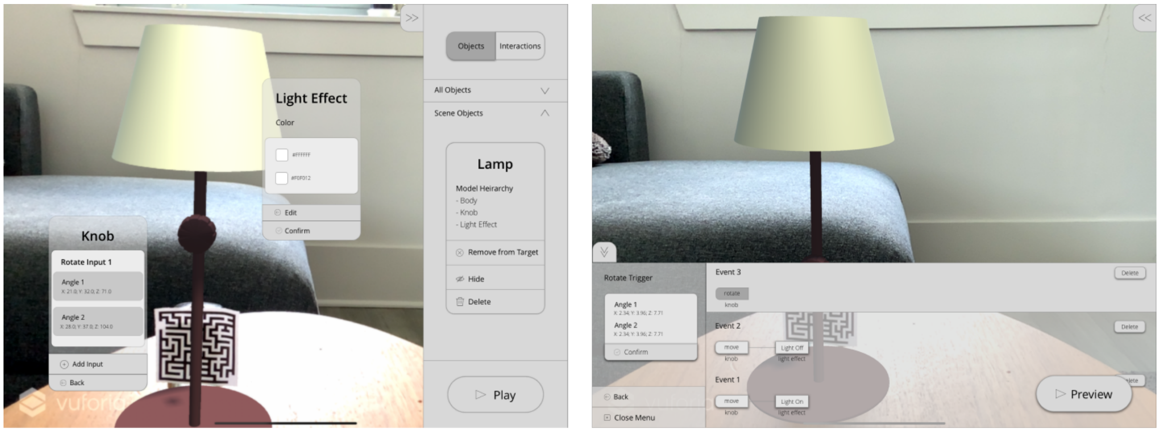


Figure 7 - Screen to author the interaction where the light changes color when the knob is rotated for the Component-Based Prototype (left) and the Event-Based Prototype (right).

A cognitive walkthrough was conducted with 5 usability experts to gather feedback on the digital prototypes. This research method was selected because the goal of this activity was to learn what interface elements hinder task completion, which would ensure that participants can complete the tasks in the final study. Usability experts for this activity were students of Georgia Institute of Technology, Atlanta, who were either enrolled in the

Master of Science in Human-Computer Interaction or Master of Industrial Design program. All the participants for this study had at least 6 month of full-time or internship work experience as UX designers or UX researchers.

3.2.3 Final Prototype

Both the prototypes were updated to account for the feedback that was received from the first round of testing. A significant change for the component-based prototype was that the UX flow was streamlined by adding a "link property" button to the last step of setting up the input. Previously, users had to set up the input, then go back to the main menu, set up the output, then go back to the main menu, and select to link the input to the output. In the updated design, users could continue the authoring process by selecting "link property" right after setting up the input. Secondly, certain interface elements in component-based prototype were also grouped together to reduce divided attention and keep the users focus on one part of the screen. This made information that was relevant for linking properties appear close together, making it easier to understand and link properties together. Finally, users mentioned that the instructional text for some actions was unclear, which was updated in the design. For the event-based prototype, the significant feedback was about redundancy in the UX flow. Users had to confirm certain action two times, which they thought was unnecessary. The interface and the flow for the event-based model was updated to reduce this redundancy.

The next version of the prototypes was developed at a higher fidelity with improvements in the UX flow and the aesthetics. The digital prototypes were designed in Sketch and interaction for the prototypes was developed in Framer. Framer was used for putting together the final prototype because it allowed for developing interactions, like the light turning on when the knob is moved up on the lamp, through a scripting language. Further, it also allows for sharing the prototype through a web URL with users for usability testing. Figure 8 and Figure 9 show the process of authoring interactions for the lamp scenario with the component-based prototype. First, the user needs to set up an input, which is represented in Figure 8. The user selects the knob object, selects the “move” input, confirms the first position, then adds the second position, and confirms the “move” input. Next, the user needs to link the input to the state of the light effect. As seen in Figure 9, the user selects “link input”, selects the “light effect” object, and selects the “light state” property. The user then links “position 1” to “light off” and confirms that interaction. Finally, the user links “position 2” to the “light on” state. This process resulted in the authoring of the interaction where, when the knob is moved up, the light turned on and when the knob is moved down, the light turned off. Details about the interactions that have been authored are displayed on the right part of the interface.

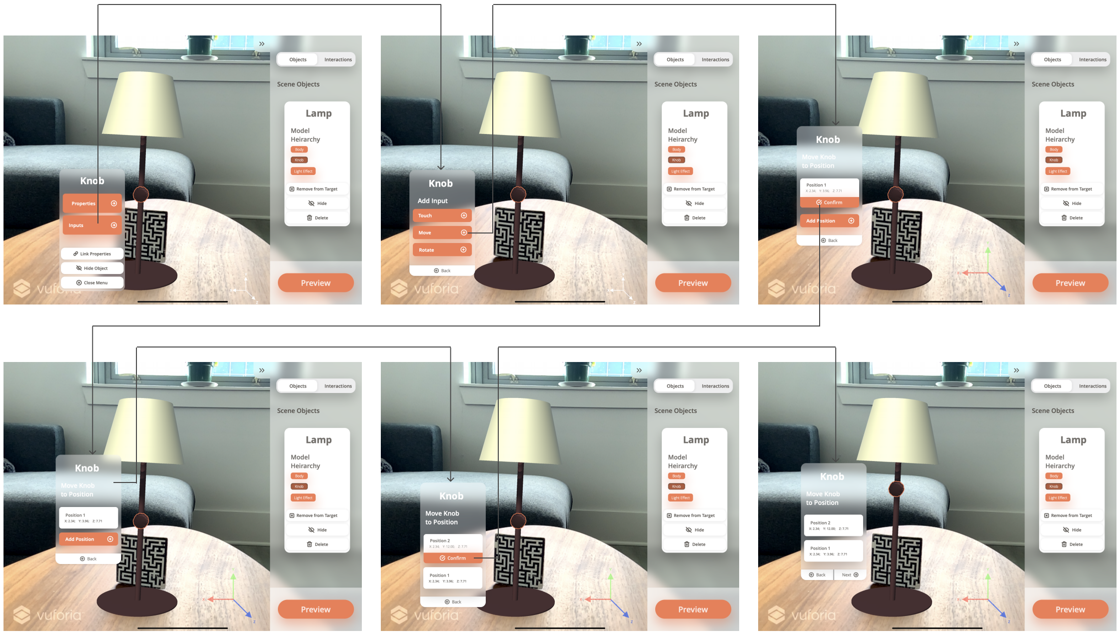


Figure 8 - Interaction flow of setting up the knob positions with the component-based prototype.

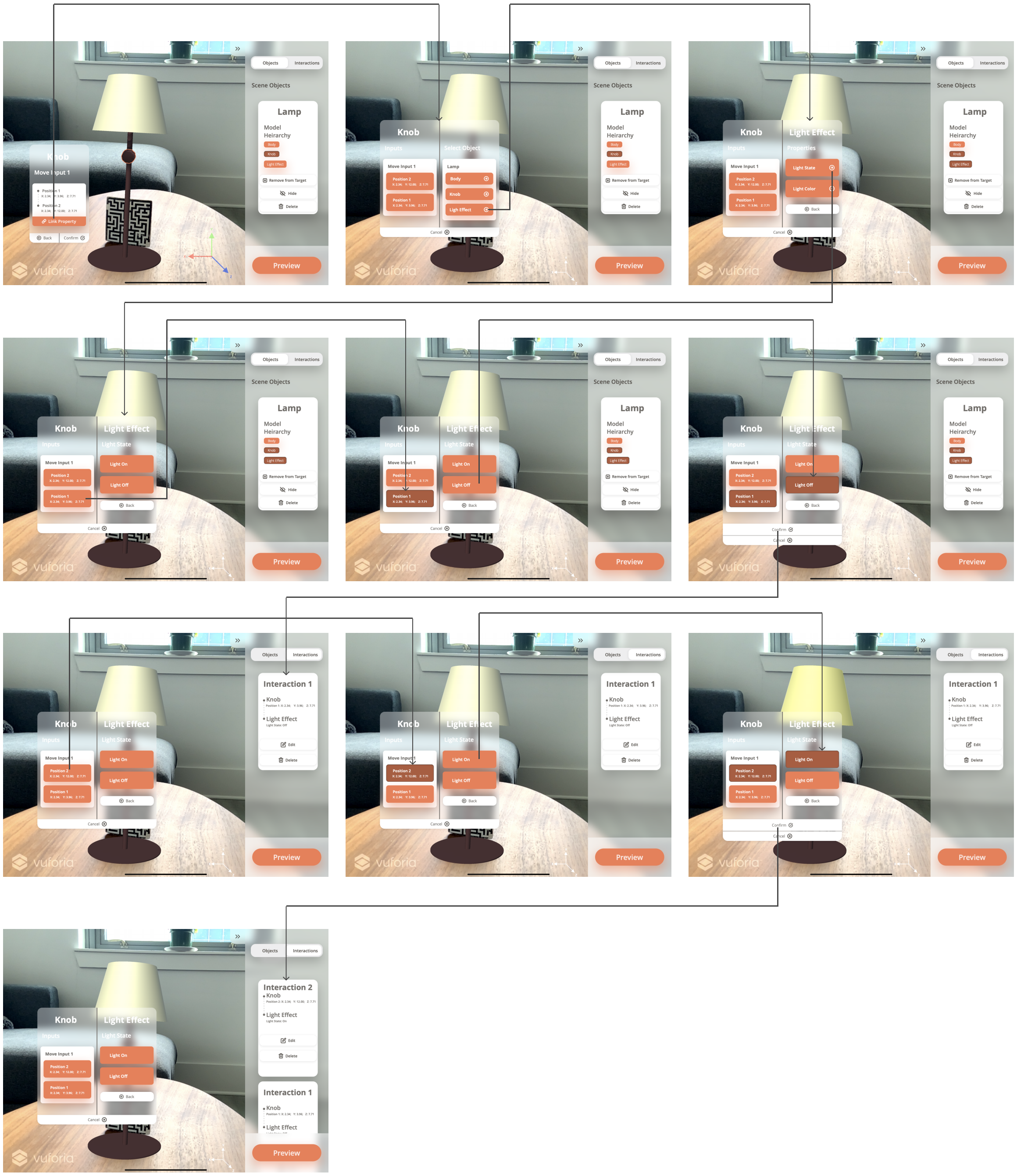


Figure 9 – Interaction flow of linking the knob positions to the light states with the component-based prototype.

Figure 10 and Figure 11 visually represent the flow for authoring interactions for the table lamp the event-based prototype. In Figure 10, the user is setting up the input of moving the knob down. The user first starts with creating an event, then selects the knob object for the trigger, selects the “move” input, and sets up the two positions between which the knob will be moved. In Figure 11, the user is setting up the action for this move trigger. The user selects “Add Action”, then selects the “light effect” object, selects the “light state” property, and finally selects “light off”. The user can view the input and the output of the action they just set up at the center of the screen. This process results in the authoring of the interaction where, when the knob is moved down, the light turns off.

Figure 10 - Interaction flow of setting up the input of moving the knob down in the event-based prototype.

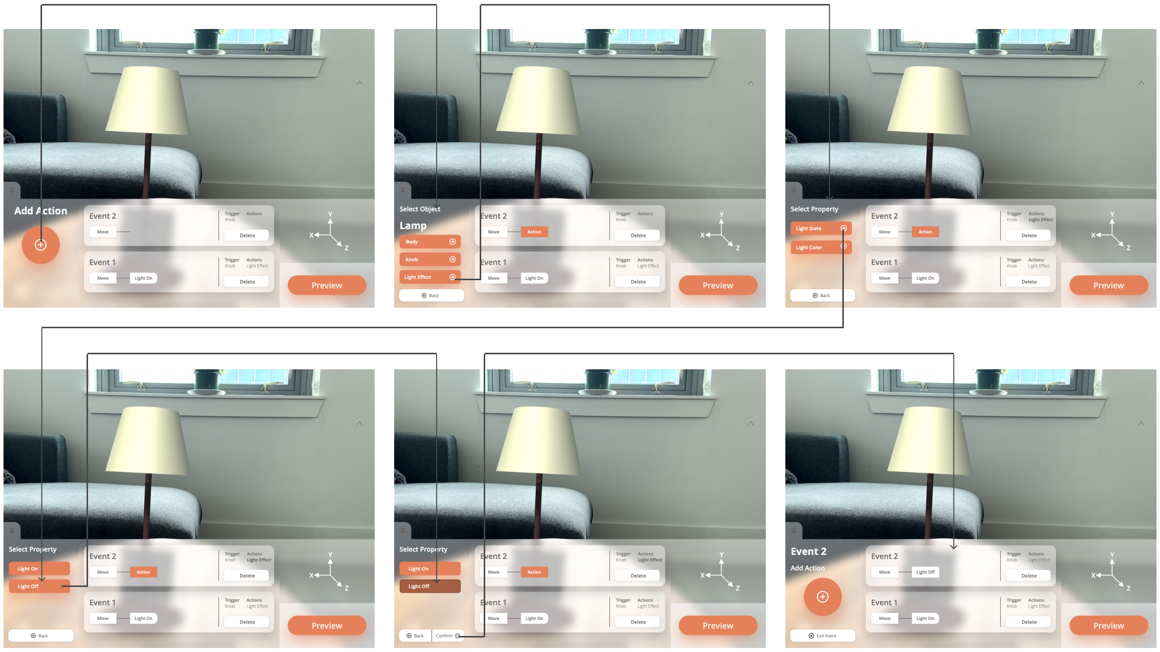


Figure 11 - Interaction flow of turning the light off when the knob is moved down in the event-based prototype.

The final prototypes that were developed for the study did not provide a true AR experience but they mimicked an AR experience. The interface background was an image that represented the physical world, and an image of a 3D model was placed on top of the background image. This created the illusion of a digital 3D model being viewed in the physical world. This was considered to be adequate for the study because the goal of this study was to evaluate the UX architectures or the process of authoring interactions and not the prototype itself. The users could understand that this was an AR experience and they could click through the prototype to achieve their task. The only difference between the two prototypes was the process of authoring interactions and the information relevant to the process of authoring. All other features and functionalities, like setting up the inputs or object properties, in the two prototypes were designed in the same was, to ensure that they do not have an effect on the results.

The prototype that was updated after the first round of testing was determined to be the final prototype that was to be used for the final user study. This was decided since the feedback from the round of testing did not entail any significant problems with the way the prototypes worked and all minor issues were updated for the final prototype. A pilot user study was conducted with each of the prototypes. During the pilot studies, users were able to complete the tasks with the prototype and no issues were highlighted. This ensured that the prototypes were developed to a point that allowed users to complete the tasks.

CHAPTER 4. METHODOLOGY

4.1 User Study Design

A remote usability study was conducted to compare the efficiency and the usability of the two prototypes for authoring interactions for AR. Participants tested the prototypes on their web browser that ran on their desktop or laptop. A between-subjects design was used where each participant performed a usability evaluation with only one condition. The independent variable in the study was the type of prototype used for developing interactions: component-based prototype or event based-prototype and the dependent variable was the time it took the participants to complete the assigned tasks. The condition was provided to participants in a randomized order and it was ensured that half of the participants performed tasks for the lamp scenario first and the other half performed tasks for the car scenario first.

4.2 Accommodations for Remote Testing

This study was conducted remotely to ensure the safety of the participants and researchers during COVID-19. Several factors were considered in the design of the prototypes and the study methodology to accommodate for a remote usability study. Firstly, even though the prototypes were designed for mobile AR, it was decided that participants will test the study prototypes on their desktop or laptop. Testing the prototypes on an iPad would require the participants to use their own iPad, which would cause variability in the study materials since the prototypes will be tested on different versions of the iPad that have varying screen sizes. This confounding effect cannot be controlled by the researcher. Further, it is more challenging for the researcher to monitor the actions that are taken by the user on an Ipad as compared to monitoring the user's desktop screen through a screen share. As discussed in the previous section, the prototypes were not a functioning software which provided a true AR experience. It was decided that the prototypes should mimic an AR experience because a true AR experience would entail the users seeing their own environment in the augmented view, print the image target themselves, and setup their space with the Ipad and the image target. To reduce the effect of variability in the environment and study setup on the results, it was decided that all participant should work with a prototype and study setup, where they see and do the same thing. Figure 12 shows a screen shot of a remote study, where the user is using the component-based prototype and has shared his screen with the researcher.

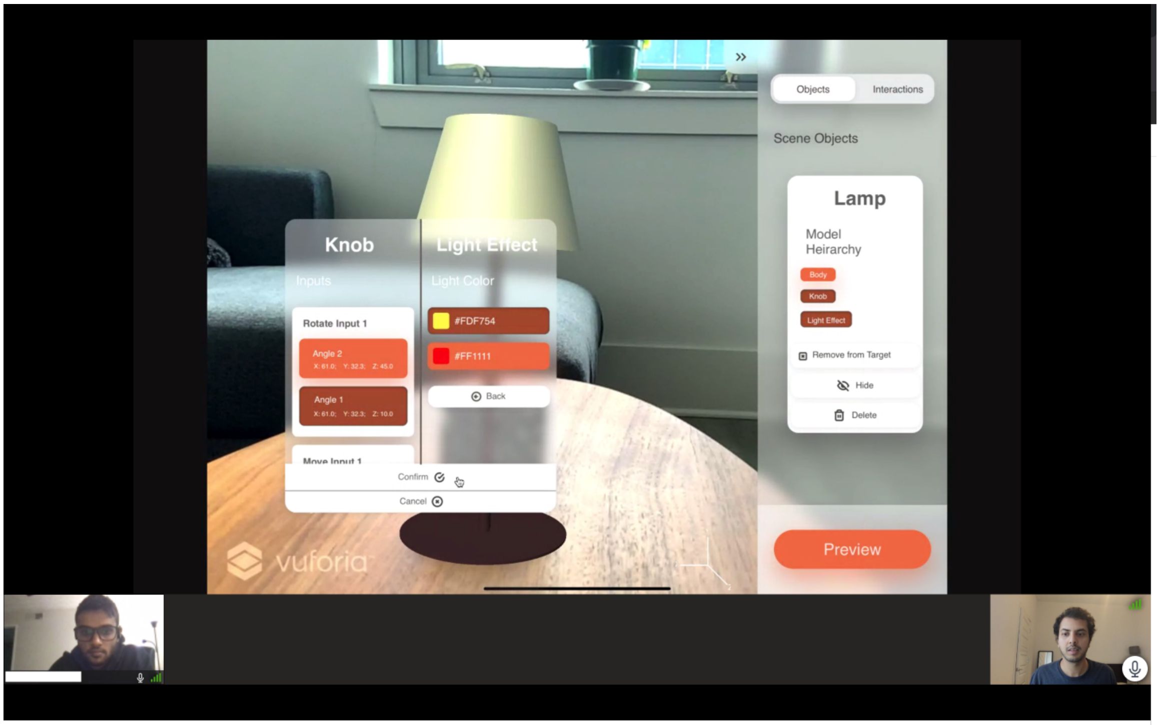


Figure 12 – Screen sharing for the remote usability study.

4.3 Participants

22 (13 female, 9 male) participants were recruited through word of mouth or a recruitment survey. All participants were current students or alumni of the Georgia Institute of Technology, Atlanta and had at least one degree or certification in a design related field. The participants also had at least 6 months of professional work experience in industrial design or UX design, which was confirmed through the screening survey. Since, the students who enrolled in the study were in a master's program, they had professional work experience prior to starting the program or through internships. It was required that all participants be non-programmers, i.e., participants could not have an education in computer science or related field and did not have any work experience related to software engineering. 9 participants claimed that they have taken some online programming classes but have never developed a software application and 5 participants have programmed design prototypes for testing but have never developed something professionally. Out of the 22 participants, 13 were very familiar with AR and have used it several times and 8 were somewhat familiar, where they have heard about the technology but have never seen an AR experience. Only 1 participant claimed that they did not know anything about AR. Even though the participants were randomly assigned a condition, the distribution of participants by familiarity with AR was fairly even between the two conditions. The component-based condition had 6 participants who were very familiar and 5 participants who were somewhat familiar with AR whereas the event-based condition had 7 participants who were very familiar, 3 participants who were somewhat familiar, and 1 participant who was not familiar with AR. With such a distribution, it can be assumed that the level of familiarity with AR did not skew the results.

4.4 Procedure

Participants had to develop interactions for two scenarios:

Scenario 1: Interactions for a Table Lamp

Task 1: The vertical translation of the lamp’s knob should trigger a change in the state of the light. When the knob is moved up, the light should turn on and when the knob is moved down, the light should turn off.

Task 2: The rotation of the knob should change the color of the light. When the knob is rotated clockwise once, the light turns red and when the knob is rotated back to its original position, the light turns back its default color of yellow.

Scenario 2: Interactions for a Toy Car

Task 1: Press and hold the car to make the car move towards the left along the X-axis.
Task 2: Tap on the car to make it rotate by 90 degrees around the Y-axis.
Task 3: The car should continue moving in the direction it rotated in for 10 seconds and then come to a complete stop.

Participants were clearly explained their goal for the study, each task for the respective scenario, and the logic of the UX architecture they would be working with. The respective UX architecture was explained to the participant because the goal of this study was to evaluate the efficiency of the architecture in authoring interactions and not to evaluate how long it takes the participant to understand the architecture. It was decided that a providing brief explanation of the architecture will put the focus of the study on how the participant is being able to use that logic as compared to solely trying to figure out what the logic is. Time was recorded, manually with a stopwatch, for each of the scenarios separately and then summed to provide the total time. The time recording began once the researcher asked the participant to start the task and the recording was stopped once the participant completed the task of developing interactions, right before they previewed their work. The usability of the prototypes was also evaluated with the System Usability Scale (SUS) at the end of the study session to gather additional user feedback from participants. The descriptive analysis and significance testing for the time and SUS data was done in RStudio.

CHAPTER 5. RESULTS

The Wilcox Rank-Sum Test was used to evaluate the significance in the difference of the time data and SUS response. This non-parametric test was chosen due to the independent variables being nominal and the interval dependent outcome violating the normality assumption. Table 1 and Table 2 show the time results, in seconds, for the component-based prototype and the event-based prototype. The Wilcox Rank-Sum test indicated that the time taken for completing the tasks with the event-based prototype *(Mdn* = 309.33) was less than the time taken to complete the tasks with the component-based prototype (*Mdn* = 433.33), Z = 94, p = .03. For the lamp scenario, users spent less time with the component-based prototype (*Mdn* = 158.75) as compared to the time spent on the event-based prototype (*Mdn* = 188.96). However, the Wilcox Rank-Sum test indicated that this result was not significant, Z = 50, p = 0.51. The result for the toy car scenario was contrasting from the result for the lamp scenario, where users spent more time with the component-based prototype (*Mdn* = 220.59) as compared to the time spent on the event-based prototype (*Mdn* = 120.65), Z = 112, p = .0008.

Table 1 - Descriptive statistics for component-based prototype (in seconds)

Scenario	Mean	Median	Std. Deviation	Skewness	Kurtosis
Table Lamp	168.07	158.75	53.66	-0.03	-1.42
Toy Car	236.37	220.59	70.55	0.88	0.09
Total	404.44	433.33	97.60	0.09	-0.74

Table 2 - Descriptive statistics for event-based prototype (in seconds)

Scenario	Mean	Median	Std. Deviation	Skewness	Kurtosis
Table Lamp	189.37	188.96	36.32	0.32	0.14
Toy Car	126.02	120.65	49.13	0.22	-1.79
Total	315.40	309.33	60.88	0.45	1.04

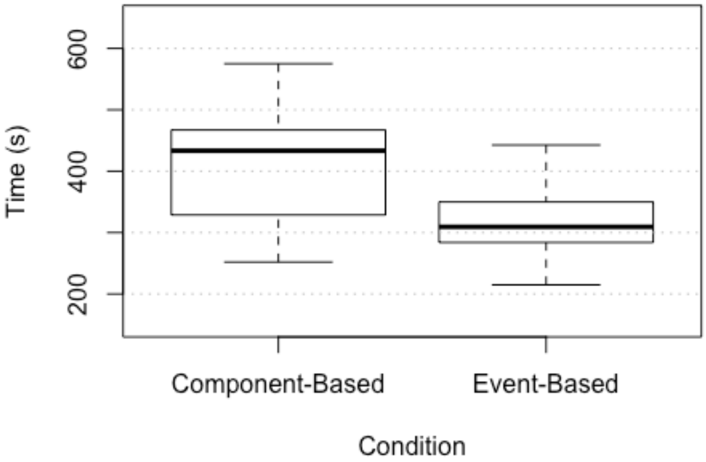


Figure 13 - Box plot of total time for the component-based and event-based prototypes

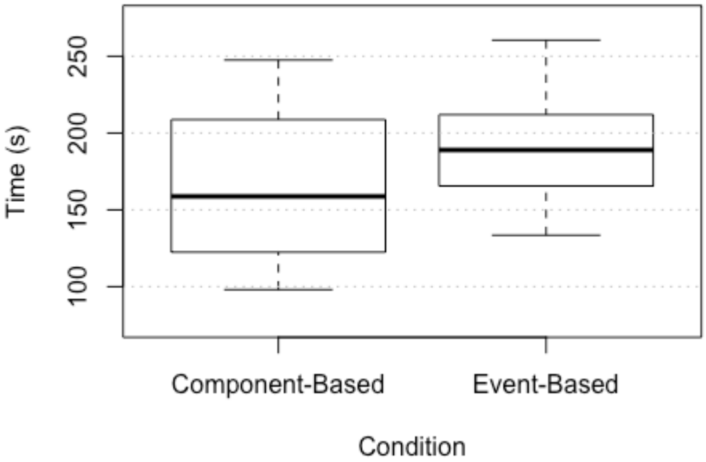


Figure 14 - Box plot of the time taken to complete the table lamp scenario

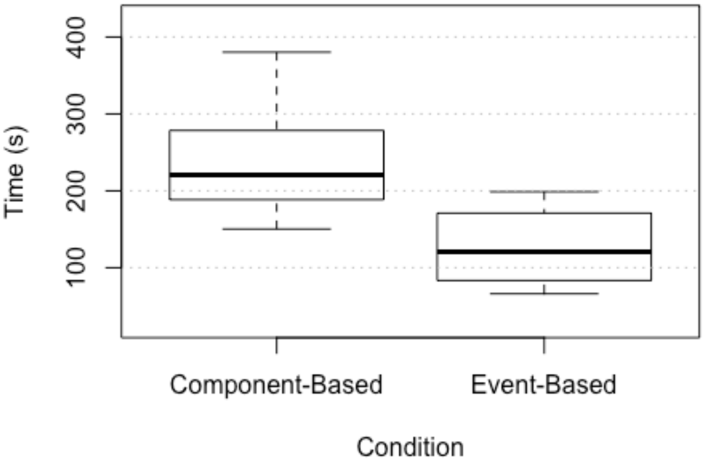


Figure 15 - Box plot of the time taken to complete the toy car scenario

For the SUS scores, The Wilcox Rank-Sum Test indicated that the participants preferred the event-based prototype (*Mdn* = 75) over the component-based prototype (*Mdn* = 67.5), Z = 26, p = .024. Table 3 shows the descriptive statistics of the SUS ratings for the two conditions.

Table 3 - SUS scores for the component-based and event-based prototypes

Scenario	**Mean**	**Median**	**Std. Deviation**	**Skewness**	**Kurtosis**
Component-Based	66.14	67.5	15.83	-2.32	6.77
Event-Based	76.60	75	13.93	-1.30	3.45

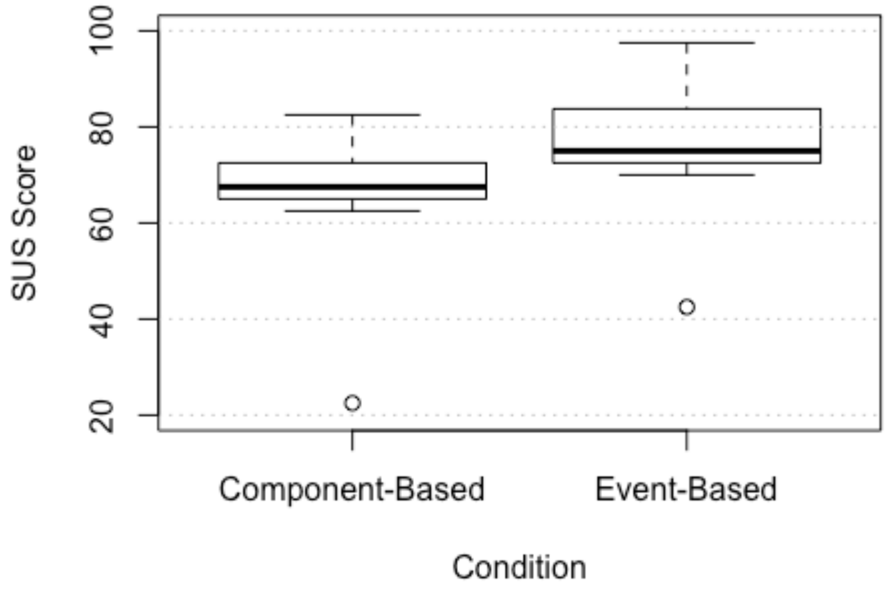


Figure 16 - Box plot of SUS scores for the component-based condition and the event-based condition

The results show that the event-based prototype performed significantly better than the component-based prototype in both, the evaluation for efficiency and usability. The SUS score of the component-based prototype for this study (*M* = 66.14) was calculated to be less than 68, which is considered to be unacceptable as per SUS usability standards (Sauro, 2011).

CHAPTER 6. DISCUSSION

This study set out to investigate whether a component-based UX architecture is more efficient than an event-based UX architecture for authoring interactions for AR for designers. The results proved the opposite, where the event-based architecture was more efficient and user friendly than the component-based UX architecture. The overall difference in the time taken to complete the tasks between the two conditions was approximately 1 minute and 30 seconds. This time difference has a significant practical impact because in a real world scenario, users will be working with more complex projects that entail authoring a large number of interactions. When an application becomes more complex and users have to author a much larger number of interactions, the time it would take to complete each task with the component-based UX architecture would be greater than the time it would take to complete each task with the event-based UX architecture, thereby, resulting in a significantly large cumulative time difference between the two conditions. A possible explanation for the event-based UX architecture producing more efficient results can be seen in the SUS ratings of the two conditions. Users provided a better rating for “easy to use” and “I found the system unnecessarily complex” for the event-based prototype as compared to the rating for the component-based prototype, implying that the easier understanding of the logic behind the event-based prototype resulted in the users being able to complete the tasks more quickly with the event-based prototype. Even though the respective architecture was explained to the participant during the study, for the component-based prototype, it was noticed that participants often got lost in the process when it came to linking properties to author the interaction. They could successfully set up the property, but they could not recall that the way to develop the interaction is by linking properties together. However, it was observed that displaying information to show an action linked to a trigger, in the event-based prototype, reminded the participants where they were in the process and what action needed to be performed next.

Comparing the time values for the two scenarios in Table 1 and Table 2 shows that the component-based prototype was more efficient than the event-based prototype for the table lamp scenario. Even though this result was not statistically significant, it encourages for a discussion of an important property of the event-based UX architecture that was considered for this study. An event can only have 1 trigger/input but it can have multiple actions/responses to that trigger, which are set off one after another. Considering this property, each task for the table lamp scenario had to be broken down further into two subtasks. For example, in the component-based model, to develop the interaction where the knob is translated vertically to turn the light on and off, participants had to setup two position properties for the knob, one associated with the light turning on and the other associated with the light turning off, and then they linked the position property to the respective state of the light. Both the position properties could be setup in one go. However, for the event-based prototype, users had to first setup the trigger of the knob being moved up followed by setting up the respective action of the light turning on for that trigger. After this interaction was previewed by the participant, they had to setup another event for the light turning off when the knob is moved down. This process was a little more tedious as compared to the process of setting up such an interaction in the component-based prototype, and hence, it resulted in participants taking more time to develop the interactions for the lamp scenario with the event-based prototype. This property of the event-based UX architecture will have significant practical impact when it is used at scale to author a large number of interactions in a project. This challenges in scaling trigger-action programming for use in practical settings was also discussed in Ur et. al.'s study (Ury et al., 2014). They proposed another framework for the event-driven approach where the authoring process would entail setting up a trigger, a condition, and an action or response. When the trigger comes true and the condition is met, the response takes place. Reality Editor 2.0 is an AR authoring tool that is addressing the shortcomings of the event-based prototype by combining multiple approaches for authoring AR interactions ("Getting Started – Reality Editor," 2020). All physical and digital objects are represented as components, where a component can be dropped into the AR scene and linked with one another to allow for sending data between components. A logic node can also be added between the components, which can be edited with an event-based paradigm, to further customize the object behavior or interaction. Such an authoring approach can be explored further, where the component-based and the event-based approaches are combined to provide an optimal experience of authoring interactions for AR.

Earlier in the paper it was explained that for this study interaction development/authoring is being defined by two types of scenarios. The first, is a scenario where a user input on one object triggers an output on another object and the second, is an interaction where a user input on one object triggers a sequence of responses from the same object. One is a situation where a user gets a single immediate response to their input and the other is a situation where an input sets off a series of events, which is analogous to pressing a button to start an animation. Evaluating the results from this perspective, it can be reported that the event-based UX architecture performs significantly faster than the component-based UX architecture for scenarios where a user provides a single input to set off a series of actions. This was evident in the scenario for the toy car, where the user developed an interaction where they tapped on the car and the car rotates, moves forward, and then comes to a stop in 10 seconds. The concept of placing actions or properties one after another is also common in other video editing software like, Adobe Premier Pro, iMovie, and even other AR media authoring tools like DART. In these software, events are placed at different time stamps and they occur one after the other once the video has started playing. This process of authoring applications has been well established and is familiar by most designers. The authoring process in these software's further support the results of this study that the mental model of placing a trigger and adding responses to those trigger in a chronological manner can be easily understood when used to author applications, even when the applications that are being authored are novel and not familiar to the user.

CHAPTER 7. LIMITATIONS

The purpose of this study was to evaluate the process of authoring interactions for AR experiences and not to evaluate the usability of the prototypes that was developed for the study. The confounding effect of the UI elements on the results posed a challenge to the accuracy of the findings for this study and hence, measures were taken in the design of the prototypes to reduce the effect of the UI elements on the results. Firstly, a user study was conducted to test the interface elements, of the low-fidelity digital prototype, for their clarity in assisting the user with completing the tasks. This ensured that the interface was designed to a level where the users could complete the tasks as efficiently as possible. Secondly, the same visual design was used for both the prototypes. The only difference between the two prototypes was the process of authoring interactions and any feedback information that is relevant to that process. All other functionalities, like setting up the position of objects, was designed in the same way for both the prototypes. These measures ensured that if a UI elements caused an effect on the results of one prototype, it would cause the same effect on the results for the other prototype.

Certain accommodations had to be made to the design of the prototypes and the testing methodology to accommodate for remote testing. Testing a functioning software in-person, in a controlled lab setting, could provide more accurate results as compared to testing the an interactive prototype remotely. Seeing a true augmented view, through a mobile device, and interacting with digital objects that are anchored in the real world with a fiducial marker, provides a more realistic AR experience, where less abstraction about the whole

AR system setup is required. Testing a functioning software will also provide more realistic feedback to a user’s actions and hence, future work should entail testing the two architectures with a more complete functioning software that provides a true AR experience.

CHAPTER 8. CONCLUSION

The goal of this study was to investigate whether a component-based UX architecture is more efficient than an event-based UX architecture for authoring interactions for AR experiences for designers. To answer this question, two digital prototypes were developed to represent each of the architectures' and the prototypes were remotely tested with 22 participants for their efficiency and usability. Participants used a prototype to complete tasks for two scenarios: developing interactions for an AR table lamp and developing interactions for an AR toy car. It was found that the event-based prototype was significantly faster than the component-based prototype and it also provided a better UX. Even though the event-based prototype was faster, a limitation of this prototype was that it only allowed for adding one input/trigger for every event, which can prove to be inefficient in some scenarios. However, in the design of a complete system for authoring interactions for AR through a GUI, an event-based UX architecture should be used since this proved to be efficient and easy to use and learn, as compared to the component-based UX architecture. The purpose of this thesis was to inform the design of a complete software that would allow non-programmers to author interactions for AR experiences so that designers can adopt AR for developing design prototypes of physical products in the industrial design process. This study provides information on the UX architecture of such a tool and what is an optimal logic that users can easily understand to get going with the system. However, future studies need to evaluate how this architecture will work for developing a larger number of interactions in a complex application and how this architecture will work with a system that also allows for developing a large number of properties for objects in the AR scene.

APPENDIX A. RECRUITMENT AND SCREENING SURVEY

Block 1

I am a master's student in the Industrial Design program at Georgia Institute of Technology, Atlanta. This survey is part of my master's thesis, whose goal is to compare the efficiency of two different interfaces and architectures in developing interactions for augmented reality (AR) without writing any code. I am seeking participants for this study who have an education and/or experience in a design related field. This recruitment screening survey has around 17 questions and should not take more than 5 minutes to complete.

Please, reach out to Karan Jain at kjain33@gatech.edu if you have any questions or comments regarding the study or this survey.

Default Question Block

What is your gender?

Female
Male
Other

In what year were you born?

Have you successfully completed your undergraduate education?

Yes
No

List all majors and minors pursued in your undergraduate education.

Have you successfully completed or are in the process of obtaining a graduate education?

Yes

No

List all majors and minors pursued in your graduate education.

Do you have any other education or certification in a design related field? If Yes, Please list the name of the course(s).

Yes

No

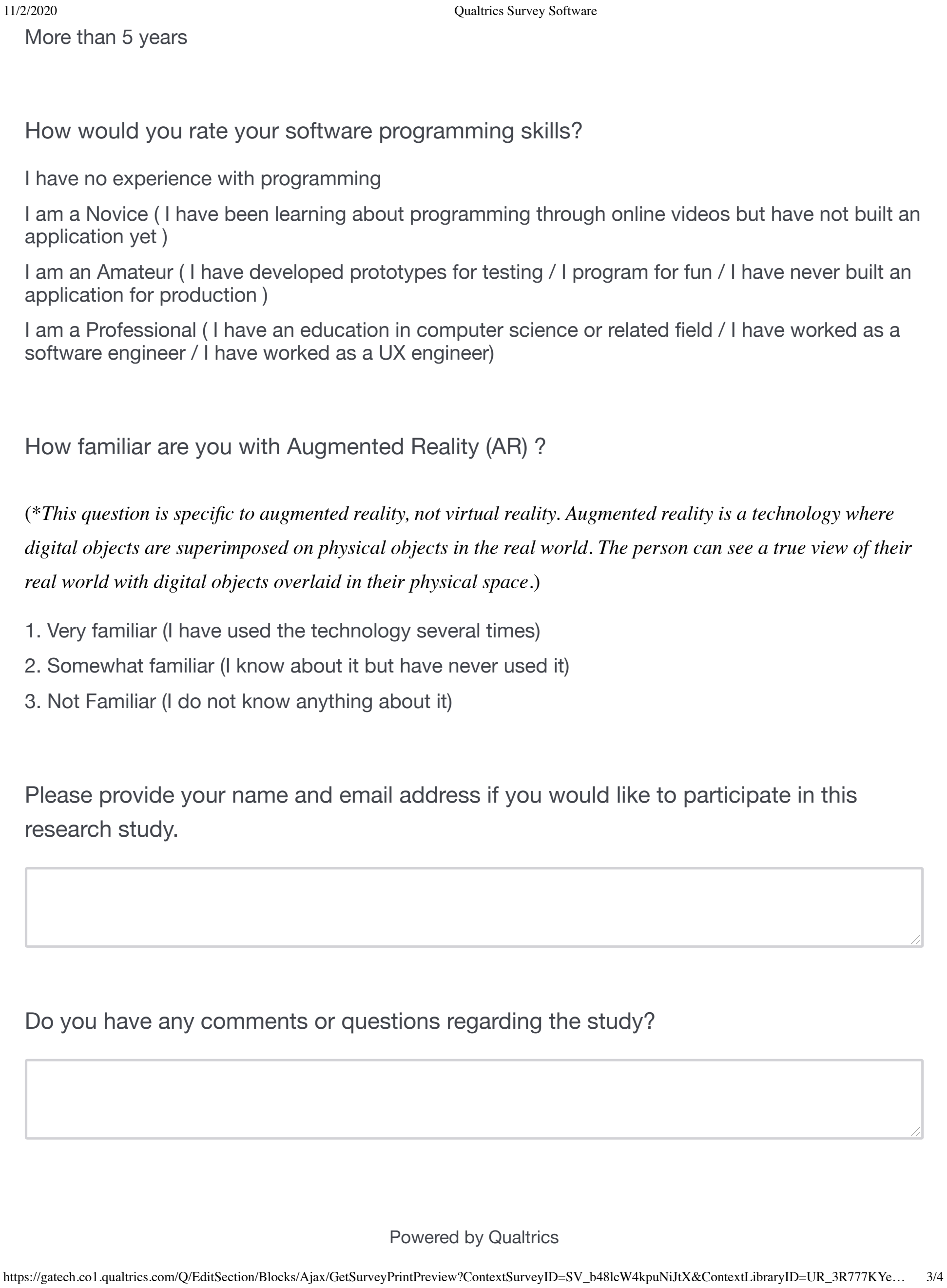
How many years of professional work experience do you have in a design related field (UX Design, Graphic Design, Industrial Design, Human-Computer Interaction, etc.)? This includes full-time and part-time experience.

No Experience

0 - 0.5 years

0.6 - 2 years

3 - 5 years



More than 5 years

How would you rate your software programming skills?

I have no experience with programming

I am a Novice (I have been learning about programming through online videos but have not built an application yet)

I am an Amateur (I have developed prototypes for testing / I program for fun / I have never built an application for production)

I am a Professional (I have an education in computer science or related field / I have worked as a software engineer / I have worked as a UX engineer)

How familiar are you with Augmented Reality (AR) ?

(*This question is specific to augmented reality, not virtual reality. Augmented reality is a technology where digital objects are superimposed on physical objects in the real world. The person can see a true view of their real world with digital objects overlaid in their physical space.*)

1. Very familiar (I have used the technology several times)
2. Somewhat familiar (I know about it but have never used it)
3. Not Familiar (I do not know anything about it)

Please provide your name and email address if you would like to participate in this research study.

Do you have any comments or questions regarding the study?

Powered by Qualtrics

APPENDIX B. RAW TIME DATA

Component Based (seconds)								
Participant ID	**Lamp 1**	**Lamp 2**	**Car 1**	**Car 2**	**Car 3**	**Total Lamp**	**Total Car**	**Total Time**
P2	101.41	93.46	175.94	90.61	113.71	194.87	380.26	575.13
P6	52.74	45.31	124.3	79.15	58.27	98.05	261.72	359.77
P4	162.09	85.43	92.26	31.86	61.69	247.52	185.81	433.33
P8	150.58	52.31	119.35	66.81	109.02	202.89	295.18	498.07
P11	56.27	45.53	113.7	36.45	51.53	101.8	201.68	303.48
P5	132	82.62	107	35.99	77.6	214.62	220.59	435.21
P13	97.65	45.45	81.03	29.06	57.62	143.1	167.71	310.81
P15	88.42	70.33	186.28	53.68	78.33	158.75	318.29	477.04
P16	98.91	56.61	80.21	55.54	55.44	155.52	191.19	346.71
P20	40.55	61.2	80.72	37.31	32.25	101.75	150.28	252.03
P21	149.04	80.84	93.25	47.32	86.77	229.88	227.34	457.22

Event Based (seconds)								
Participant ID	**Lamp 1**	**Lamp 2**	**Car 1**	**Car 2**	**Car 3**	**Total Lamp**	**Total Car**	**Total Time**
P3	69.25	79.52	128.97	23.48	46.32	148.77	198.77	347.54
P9	135.18	125.3	120.62	41.53	19.99	260.48	182.14	442.62
P10	133.4	76.1	58.08	30.49	32.08	209.50	120.65	330.15
P12	91.12	84.99	103.27	45.39	29.93	176.11	178.59	354.70
P7	89.92	99.04	87.16	33.53	42.62	188.96	163.31	352.27
P14	131.05	83.36	34.9	19.31	21.13	214.41	75.34	289.75
P17	79.43	75.56	72.31	54.34	18.53	154.99	145.18	300.17
P18	108.51	84.18	34.28	18.11	33	192.69	85.39	278.08
P19	66.29	67.2	41.97	24.67	14.73	133.49	81.37	214.86
P1	105.4	78.43	27.84	21.47	16.75	183.83	66.06	249.89
P22	139.87	80	47.47	19.88	22.11	219.87	89.46	309.33

APPENDIX C. SUS QUESTIONNAIRE

System Usability Scale

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
2. I found the system unnecessarily complex	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
3. I thought the system was easy to use	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
4. I think that I would need the support of a technical person to be able to use this system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
5. I found the various functions in this system were well integrated	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
6. I thought there was too much inconsistency in this system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
7. I would imagine that most people would learn to use this system very quickly	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
8. I found the system very cumbersome to use	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
9. I felt very confident using the system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5
10. I needed to learn a lot of things before I could get going with this system	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5

APPENDIX D. SUS RAW DATA

Component-Based											
Participant ID	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**SUS Score**
P2	3	2	4	2	5	1	4	1	4	2	80
P4	4	4	3	1	3	2	4	3	5	2	67.5
P5	1	5	2	4	3	4	2	5	2	3	22.5
P6	3	2	4	4	4	1	4	2	4	3	67.5
P8	3	2	4	2	3	2	5	2	3	2	70
P11	4	3	2	1	5	1	5	1	4	4	75
P13	2	1	4	3	4	3	4	2	4	4	67.5
P15	4	2	3	2	2	2	4	2	3	3	62.5
P16	2	2	3	2	3	1	3	2	4	3	62.5
P20	2	4	3	1	3	1	5	1	5	3	70
P21	5	1	4	2	5	1	4	1	4	4	82.5

Event-Based											
Participant ID	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**SUS Score**
P1	3	2	4	2	5	1	3	2	3	1	75
P3	3	2	5	1	5	1	4	1	5	4	82.5
P7	4	2	3	3	4	1	4	2	4	2	72.5
P9	4	1	2	2	4	2	5	1	3	2	75
P10	5	2	4	1	3	4	5	2	3	2	72.5
P12	5	1	5	1	4	1	5	1	5	1	97.5
P14	3	2	5	1	4	1	4	2	5	1	85
P17	4	2	4	2	5	1	5	1	5	2	87.5
P18	5	1	4	2	4	3	5	1	5	3	82.5
P19	4	4	5	3	4	1	3	2	4	2	70
P22	3	3	2	3	4	4	3	3	2	4	42.5

APPENDIX E. CONSENT FORM

CONSENT DOCUMENT FOR ENROLLING ADULT PARTICIPANTS IN A RESEARCH STUDY

Georgia Institute of Technology

Investigators: Karan Jain and Young Mi Choi, Ph.D.
Protocol and Consent Title: Authoring Tangible Augmented Reality Interactions
(Protocol H19549 12/13/19 v1)
You are being asked to be a volunteer in a research study

Purpose:
The purpose of this study is to compare the efficiency of developing interactions for augmented reality (AR) experiences with a component-based model interface vs. an event-based model interface. We hope to make the process of authoring interactions for AR more efficient by allowing users to build interactions with commonly known interaction paradigms, such as drag and drop and touch, with a 2D interface to author interactions instead of writing scripts/code.
We expect to enroll up to 20 participants in this study.

Exclusion/Inclusion Criteria:
Participants in this study must be current registered students or be an alumni of the Georgia Institute of Technology, Atlanta. It is required that the participant should have an education or certification in a design related field like Industrial Design, UX Design, or UX Research. Further, the participant also needs to have at least 6 months of work experience in design either through internships, jobs, or personal projects. All participants must be over 18 years of age and be proficient in understanding English. Individuals who are currently in an EU country cannot participate in this research.

Procedure:
If you decide to participate in this study, you will be randomly assigned to one of the conditions listed below. Regardless of the condition you are assigned to, your goal is to develop interactions for an AR design representation of a lamp and a toy car. This study will be conducted remotely through a BlueJeans video call. The link that was shared with you via email consists of an interactive prototype, which you will run on your web browser. You will have to share your screen so that the researcher can monitor your interactions and record the dependent variable. Please note that this session is not being video or audio recorded.

There are two main tasks you have to complete in the study. For your convenience, the two tasks are further broken down into sub-tasks:

Task 1: Interacting with a knob on a table lamp to change the state of light

a. Move the knob up to turn the light on
b. Move the knob down to turn the light off
c. Rotate the knob to the right to change the color of the light to red
d. Rotate the knob to the left (original position) to change the color of the light to yellow

Task 2: Interacting with a toy car to set up its path of motion

a. Tap and hold the toy car to start moving it to the left
b. Tap on the car to make it turn 90 degrees around the Y-axis
c. The car should continue moving in the direction it is facing, after rotating, for 10 seconds

The researcher will record the time to complete each of the tasks listed above. Further, once you have completed all the tasks, you will fill out a short questionnaire to give your opinion on the user experience of using such an interface for developing interactions for AR. You may stop at any time and for any reason during the study. Your participation is required for only 1 study session which will take no more than 75 minutes. No follow up involvement is required by you after this study session.

Condition 1: Component-Based Model Interface
The architecture of this model entails linking properties between components. Every object in the AR scene is a component that has specific, pre-defined, properties. The user can develop behaviors for components by linking these properties together.

Condition 2: Event-Based Model Interface
The architecture of this model entails creating trigger-action pairs. The user creates an event and every event has two components to it – a trigger and a response/action to that trigger. Each event can have only one trigger but multiple actions to one trigger.

Risks or Discomforts:
The possible risks involved are similar to using regular computer software interfaces.

Benefits:
You are not likely to benefit in any way by joining this study. The results of this study will inform the design of tools that will enable users to develop interactions for AR design representations or experiences more efficiently, which in turn will improve the prototyping time of design ideas in the product design process. Such a tool is only a small part of a much more complete and complex software that can ultimately be used by design teams to develop and implement AR design representations for usability testing. However, it is an important first step that will allow the design community to use AR technology in their design process, by making it easier for them to develop interactions for such experiences.

Compensation to You:
You will receive a $10.00 gift card for your participation in this study. Even if you decide to leave the study early, you will be compensated for the full amount.
U.S. Tax Law requires that a 1099-misc be issued if U.S. tax residents receive $600 or more per calendar year. If non-U.S. tax residents receive more than $75, mandatory 30% withholding is required. Your address and Tax I.D. may be collected for compensation purposes only. This information will be shared only with the Georgia Tech department that issues compensation, if any, for your participation.

Confidentiality:

The following procedures will be followed to keep your personal information confidential in this study: We will comply with any applicable laws and regulations regarding confidentiality. To protect your privacy, your records will be kept under a code number rather than by name. Your records will be kept in locked files and only study staff will be allowed to look at them. All data that is stored digitally will be saved in password-protected excel sheets, locally on the study staff's computers'. Data will be transferred among research personnel through email. Your name and any other fact that might point to you will not appear when the results of this study are presented or published. The Georgia Institute of Technology IRB and the Office of Human Research Protections may look over study records during required reviews.

Costs to You:

There are no costs to you, other than your time, for being in this study.

Questions about the Study:

If you have any questions about the study, you may contact Karan Jain at telephone (404-948-8939) or kjain33@gatech.edu

Questions about your Rights as a Research Participants:

- Your participation in this study is voluntary. You do not have to be in this study if you don't want to be.
- You have the right to change your mind and leave the study at any time without giving any reason and without penalty.
- Any new information that may make you change your mind about being in this study will be given to you.
- You will be given a copy of this consent form to keep.
- You do not waive any of your legal rights by signing this consent form.

If you have any questions about your rights as a research participant, you may contact
Ms. Melanie Clark, Georgia Institute of Technology
Office of Research Integrity Assurance, at (404) 894-6942.

If you sign below, it means that you have read (or have had read to you) the information given in this consent form, and you would like to be a volunteer in this study.

____
Participant Name (printed)

____ ______________
Participant Signature Date

____ ______________
Signature of Person Obtaining Consent Date

APPENDIX F. STUDY SCRIPT

User Study Script

Intro

Thank you for taking the time to volunteer in this study.

The goal of this research study is to compare the efficiency of two different types of interface architectures in developing interactions for augmented reality (AR) without writing any code. Currently, there are very few tools for non-programmers and/or designers to develop interactions for AR. Most existing tools require knowledge of low level programming or object oriented scripting. This is an issue for UX designers who are trying to prototype AR experiences or for Industrial Designers who may be trying to prototype a physical product in AR to represent how the product will work. The overarching goal of this project is to assist designers with prototyping AR experiences more quickly, and this particular study is going to focus on comparing the efficiency of two different models for developing interactions for AR.

Do you have any questions?

Consent Form

Before we move forward, I would like to briefly go over the consent form once again.

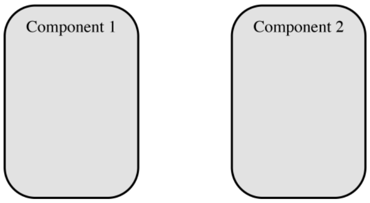
- Please note that there are no risks to you in this study.
- You are not likely to benefit in anyway from this study.
- You will be compensated with a $10 amazon gift card for your participation.
- There are no costs to you other than your time.
- There is no follow up involvement required from you.
- Please, note that all information discussed will be kept confidential and only the members involved in this study will have access to this information.
- Your participation in this study is voluntary. You may stop at any time you want.

If participant has not signed the consent form, ask them to open DocuSign link and sign it before proceeding.

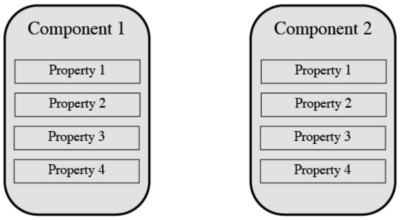
Procedure

As I had mentioned earlier, this study is comparing the interfaces and architectures of two different models for developing interactions for AR. These two models are: Component-Based Model and Event-Based Model. In this study, you will be using a Component-Based Model for developing interactions for two scenarios. A component based model interface is a front end architecture where every object in the AR scene or in your application is a component. Components are given pre-defined properties. A user can develop behaviors for the components or interactions for the components by matching the properties of different components. Here is a quick visualization for your understanding.

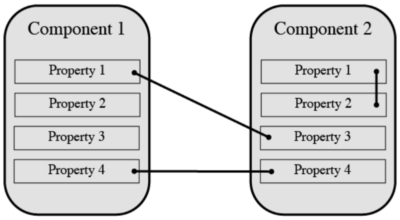
Components



Component Properties



Behaviors/Interactions



Do you have any questions?

There are two main tasks you have to complete in the study. These tasks are:

1) Interacting with a knob on a table lamp to change the state of light
2) Interacting with a toy car to set up its path of motion

For your convenience, the two tasks are further broken down into sub-tasks, which I will go over when you begin each task.

Please, note that I will record the time to complete each of the sub-tasks. Only begin interacting with the interface once I say “go”. I will share the link for each of the scenarios through the chat feature of this call. Please, click the link, full screen that web page, and share your screen with me so I can monitor what you are doing. You will be using an interactive prototype of a software for this study. The real software is intended to run on an iPad and will give you a live camera feed of your environment. However, this prototype is going to run on a web browser on your computer and is not giving you a live camera feed. The background of the interface is simulating an AR view. You have to abstract that you are looking at a digital object that is overlayed in the real world through the iPad.

Please, note that for the purpose of this study, I may not answer all the questions you have. I will guide you with specific questions but I cannot tell you what to click or how to move forward. You will have to figure that out on your own. However, feel free to ask me a question if you are stuck or something is very unclear.

Do you have any questions?

And finally, once you have completed all the tasks, you have to answer the short questionnaire that was attached to the scheduling email. Please, email me the filled out questionnaire. This survey is called a System Usability Scale (or SUS) and is assessing the user experience of using such an interface.

Do you have any questions before we begin?

The first scenario we will work with is a AR representation of a lamp. Here are the sub-tasks for this scenario:

a. Translate the knob up and down to change the state of the light. So when the knob is moved up, it turns on and when the knob is moved down, it turns off.
b. Rotate the knob to change the color of the light. When the knob is rotated once clockwise, it turns the color of the light to red and when the knob is rotated back counterclockwise, the color of the lamp turns back to the default yellow.

I am going to share my screen and show you what the final state of this overall task will look like.

CAR FIRST

(Mention that you should verbally notify me when you hit preview)

(Share screen and run the Framer Lamp project from researcher's computer)

Do you have any questions?

(Share the lamp component-based model project with through the chat. Make sure user has full screened this page and has shared his/her screen with you)

Highlight that there is no rotation interaction. User has to click on knob to rotate it.

Developing an AR experience entails two parts: 1) is creating a reference point in the physical world for digital objects. The reference point can be a barcode or image, a plane (like the surface), or it could be an object. 2) the second part is developing the interaction. To get you warmed up, your first task is going to be to place the digital model of the lamp on the image target. So go ahead and do that. This activity is not timed. So now you can see that your digital model of the lamp is placed in the environment, via the augmented medium

Again, only begin once I say "go"

1) Your first sub-task is to set up the interaction where you move the knob up and down to turn the light turn the light on and off. Please note that position 2 is associated with light on and position 1 is associated with light off.

"Go"

2) Your 2nd subtask is to rotate the knob to change the color of the light. When it is rotated clockwise once, the light turns red and when it is rotated counter clockwise once, the light goes back to yellow. I would like to highlight that this prototype does not have a rotation interaction. So you will just click on the knob to rotate it.

Great, you are finished with the first scenario of developing interactions for the lamp. We will move on to the second scenario now, which is developing interactions for a toy car.

a. Press and hold the toy car to start moving it to the left
b. Tap on the car to make it turn 90 degrees around the Y-axis
c. The car should continue moving in the direction it is facing, after rotating, for 10 seconds

I am going to share my screen and show you what the final state of this overall task will look like.

(Mention that she should verbally notify me when she hits preview)

(Share the car component-based model project with through the chat. Make sure user has full screened this page and has shared his/her screen with you.)

Again, only begin once I say “go”

1) Your first sub-task is to develop the interaction where you press and hold on the car and the car moves horizontally in the x-axis. In other words, the car will animate towards the left.

“Go”

2) Your second sub-task is to develop the interaction where you tap on the car and the car will rotate 90 degrees around the Y-axis.

“Go”

3) Your final sub-task is to develop the animation where the car continues to move in the direction it is rotated in (z-axis) for 10 seconds and then comes to a stop.

“Go”

Great, we are done with the second scenario.

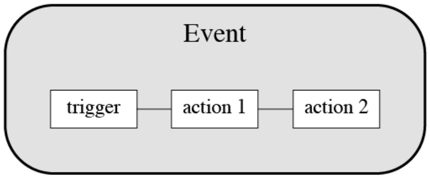
Finally, can you go ahead and open the SUS questionnaire. Please answer all the questions and email me the filled out questionnaire. Let me know if you have any questions.

Do you have any comments or questions regarding this study?

Thank you so much for participating in this study and please keep a look out for the Amazon Gift Card.

For Event-Based Model

In this study, you will be using a Event-Based Model for developing interactions for two scenarios. An event-based model interface is a front end architecture where a every interaction or behavior is developed as an event. An event has two components to it, a trigger and an action/response to that trigger. This is commonly also called trigger action programming and is a common architecture that is used in many tools that are made for non-programmers. A user first sets up a trigger and then links actions to this trigger. Note that an event can only have 1 trigger, but multiple actions. Here is a simple visualization to help you understand this model.



REFERENCES

1. Abawi, D. F., Dörner, R., & Grimm, P. (2004). A Component-based Authoring Environment for Creating Multimedia-rich Mixed Reality. *Proceedings of the Seventh Eurographics Conference on Multimedia*, 31–40.

2. Amazon Sumerian Overview. (n.d.). Retrieved December 23, 2019, from https://aws.amazon.com/sumerian/

3. Billinghurst, M., Clark, A., & Lee, G. (2014). A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, *8*(2–3), 73–272. https://doi.org/10.1561/1100000049

4. Choi, Y. M. (2019). Applying Tangible Augmented Reality for Product Usability Assessment. *Journal of Usability Studies*, *14*(4), 187–200. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=138601144&site=ehost-live

5. Create augmented reality experiences | Adobe Aero. (n.d.). Retrieved December 23, 2019, from https://www.adobe.com/products/aero.html

6. Getting Started – Reality Editor. (n.d.). Retrieved November 20, 2020, from http://realityeditor.org/getting-started/

7. Hampshire, A., Seichter, H., Grasset, R., & Billinghurst, M. (2006). Augmented reality authoring, 409. https://doi.org/10.1145/1228175.1228259

8. Haller, M., Stauder, E., & Zauner, J. (2005). AMIRE-ES : Authoring Mixed Reality once , run it anywhere. In *In Proceedings of the 11th International Conference on Human-Computer Interaction – HCII 2005*. Las Vegas, Nevada, USA. Retrieved from http://mi-lab.org/files/publications/Haller et al - HCII 2005 - AMIRE-ES, Authoring Mixed Reality once, run it anywhere.pdf

9. Heun, V., Hobin, J., & Maes, P. (2013, September). Reality editor: programming smarter objects. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication* (pp. 307-310).

10. Jain, K., & Choi, Y. M. (2019). Building Tangible Augmented Reality Models for Use in Product Development. *Proceedings of the Design Society: International Conference on Engineering Design*, *1*(1), 1913–1922. https://doi.org/10.1017/dsi.2019.197

11. Jain, K., & Choi, Y. M. (2020, July). Authoring Interactions for Tangible Augmented Reality. In *International Conference on Human-Computer Interaction* (pp. 43-50). Springer, Cham.

12. Krevelen, D. W. . (2007). Augmented Reality: Technologies, Applications, and Limitations. *Research Gate*, (April 2007), 1–25. https://doi.org/10.13140/RG.2.1.1874.7929

13. Lean Touch - Asset Store. (n.d.). Retrieved December 23, 2019, from https://assetstore.unity.com/packages/tools/input-management/lean-touch-30111
14. Lee, G. A., & Kim, G. J. (2009). Immersive authoring of Tangible Augmented Reality content: A user study. *Journal of Visual Languages and Computing*, *20*(2), 61–79. https://doi.org/10.1016/j.jvlc.2008.07.001

15. Lee, W., & Park, J. (2005). Augmented foam: A tangible augmented reality for product design. *Proceedings - Fourth IEEE and ACM International Symposium on Symposium on Mixed and Augmented Reality, ISMAR 2005*, *2005*, 106–109. https://doi.org/10.1109/ISMAR.2005.16

16. Looser, J., Grasset, R., Seichter, H., & Billinghurst, M. (2006). OSGART - A pragmatic approach to MR. *5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 06): Industrial Workshop*, 22–25. Retrieved from http://ir.canterbury.ac.nz/handle/10092/2370

17. MacIntyre, B., Gandy, M., Dow, S., & Bolter, J. D. (2005). DART: A toolkit for rapid design exploration of augmented reality experiences. *ACM Transactions on Graphics*, *24*(3), 932. https://doi.org/10.1145/1073204.1073288

18. Michelson, B. M. (2006). Event-driven architecture overview. *Patricia Seybold Group*, *2*(12), 10-1571.

19. Mota, R. C., Roberto, R. A., & Teichrieb, V. (2015). [POSTER] Authoring tools in augmented reality: An analysis and classification of content design tools. *Proceedings of the 2015 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2015*, (October 2017), 164–167. https://doi.org/10.1109/ISMAR.2015.47

20. NamanKed. (2018, September 21). Introduction to Scripting Languages. Retrieved November 03, 2020, from https://www.geeksforgeeks.org/introduction-to-scripting-languages/

21. Norman, D. A. (1986). Cognitive engineering. *User centered system design*, *31*, 61.

22. Pang, Y., Nee, A. Y. C., Ong, S. K., Yuan, M., & Youcef-Toumi, K. (2006). Assembly feature design in an augmented reality environment. *Assembly Automation*, *26*(1), 34–43. https://doi.org/10.1108/01445150610645648

23. Park, J. S. (2011). AR-Room: A rapid prototyping framework for augmented reality applications. *Multimedia Tools and Applications*, *55*(3), 725–746. https://doi.org/10.1007/s11042-010-0592-1

24. Phillip S. Dunston, Xiangyu Wang, Mark Billinghurst, B. H. (2003). Mixed Reality Benefits For Design Perception by, 1–6.

25. Purdy, T. G., & Choi, Y. M. (2014). Enhancing augmented reality for use in product design. *Conference on Human Factors in Computing Systems - Proceedings*, 1303–1308. https://doi.org/10.1145/2559206.2581251

26. Qureshi, M. R. J., & Hussain, S. A. (2008). A reusable software component-based development process model. *Advances in Engineering Software*, *39*(2), 88–94. https://doi.org/10.1016/j.advengsoft.2007.01.021

27. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., … Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, *52*(11), 60–67. https://doi.org/10.1145/1592761.1592779

28. Seichter, H., Looser, J., & Billinghurst, M. (2017). ComposAR. *Talanta*, *175*, 250–255. https://doi.org/10.1016/j.talanta.2017.07.006

29. Sauro, J. S. (2011, February 11). *MeasuringU: Measuring Usability with the System Usability Scale (SUS)*. Measuring U. https://measuringu.com/sus/

30. Seichter, H., Looser, J., & Billinghurst, M. (2008, September). ComposAR: An intuitive tool for authoring AR applications. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality* (pp. 177-178). IEEE.

31. Unity Real-Time Development Platform | 3D, 2D VR & AR Visualizations. (n.d.). Retrieved December 23, 2019, from https://unity.com/

32. Ury, B., McManus, E., Ho, M. P. Y., & Littman, M. L. (2014). Practical trigger-action programming in the smart home. *Conference on Human Factors in Computing Systems - Proceedings*, 803–812. https://doi.org/10.1145/2556288.2557420

33. Wikitude Studio-Augmented Reality Creation & Management Tool. (n.d.). Retrieved December 23, 2019, from https://www.wikitude.com/products/studio/