Story Authoring in Augmented Reality

Marie Kegeleers and Rafael Bidarra
Computer Graphics and Visualization Group, Delft University of Technology, Delft, The Netherlands
Marie@kegeleers.be, R.Bidarra@tudelft.nl

Keywords: Story Authoring, Augmented Reality, Interaction Design, Content Creation, Mixed-initiative

Abstract: Most content creation applications currently in use are conventional PC tools with visualisation on a 2D screen and indirect interaction, e.g. through mouse and keyboard. Augmented Reality (AR) is a medium that can provide actual 3D visualisation and more hands-on interaction for these purposes. This paper explores how AR can be used for story authoring, a peculiar type of content creation, and investigates how both types of existing AR interfaces, tangible and touch-less, can be combined in a useful way in that context. The Story ARtist application was developed to evaluate the designed interactions and AR visualisation for story authoring. It features a tabletop environment to dynamically visualise the story authoring elements, augmented by the 3D space that AR provides. Story authoring is kept simple, with a plot point structure focused on core story elements like actions, characters and objects. A user study was done with the concept application to evaluate the integration of AR interaction and visualisation for story authoring. The results indicate that an AR interface combining tangible and touch-less interactions is feasible and advantageous, and show that AR has considerable potential for story authoring.

1 Introduction

Content creation applications commonly used today are mostly conventional PC applications with visualisation on a 2D display and interaction using a keyboard and mouse. This includes applications for 3D content creation, e.g. 3D modeling. Graphical User Interfaces (GUI) have been the standard for years. However, a new kind of user interface gradually appears in more research and applications, the Natural User Interface (NUI). An NUI is an interface that enables users to interact with virtual elements in a way similar to how people interact with the real world. It makes use of everyday actions like gestures, touching and picking up objects, and speech for the user to control the application (Preece et al., 2015). Therefore, an NUI can have a more direct way of interacting with virtual content, which can be perceived as more natural compared to a GUI.

A fitting medium for an NUI is Augmented Reality (AR), due to its technology adding virtual elements to a real-world environment. Because of this integration with reality, an AR interface is not limited to a 2D screen but can extend to the 3D space around the user. The same holds for visualisation of virtual content, which can be displayed in front of the user as a 3D hologram.

Although AR is not yet commonly used, it could be helpful in many different fields that benefit from spatial 3D visualisations and a more hands-on and natural interaction. One of the fields in which it is worth exploring the possibilities of AR is interactive content creation. Because the author's environment and body are visible, creative AR tools can display content in a real-world context, while their NUI supports content modification through actions the user is already familiar with.

In this paper, we focus on a particular type of content creation: *story authoring*. In the past years, research has been done on ways to facilitate story authoring using technology (Castano et al., 2016), and some story authoring tools have been developed for desktop PC environments (Kybartas and Bidarra, 2016). However, little has been investigated on the potential of AR for these purposes. Therefore, this research aims at exploring how AR can be used for story authoring (Kegeleers, 2020). To do this, both types of existing AR interfaces, tangible and touch-less, are examined with the goal of integrating a fitting combination of both in a single NUI for an AR interface for story authoring. More specifically, interaction with markers and hand tracking are explored and combined to best accommodate various kinds of interactions.

A prototype application, Story ARtist, was developed that combines new interaction and visualisation concepts to evaluate their suitability for story authoring. To represent and manage the story content in the application, a simple story framework was developed that focuses on the core elements of a narrative. Partly inspired by how children tell stories manipulating puppets with their hands, Story ARtist's NUI allows the author to use their hands to directly interact with markers or with virtual elements. Therefore, to enable optimal hand interaction, a head-mounted AR device was chosen. To have all elements within reach, the application was developed for a tabletop environment, as this is a natural and convenient workspace for this type of content creation.

2 Related Work

Interaction is a crucial element when researching AR for story authoring. We, therefore, briefly discuss current AR interfaces for a variety of applications, and review some examples of research on content creation applications in AR.

2.1 AR Interfaces

Current types of augmented reality interfaces can be classified into two categories. The first category is tangible interfaces, which require the user to interact with physical elements like blocks or cards to control virtual elements. The second category is touch-less interfaces, in which the user interacts only with virtual elements like menu panels with buttons or holograms, e.g. using gestures and hand interaction. Both types of interfaces provide very different experiences and fit different applications based on the AR equipment used, the environment and the application purpose.

2.1.1 Tangible Interfaces

Tangible interfaces combine the overlaid virtual elements with physical objects. The user can manipulate and interact with these physical objects to control virtual elements. The power of tangible interfaces is that the interactive elements have physical properties and constraints that the user is familiar with. This restricts how the objects can be manipulated and therefore makes the controls easy and intuitive (Kato et al., 2000) (Zhou et al., 2008).

The most common implementation of tangible AR interfaces uses cards with markers, each marker representing a single virtual element in AR. Kato et al. (Kato et al., 2000) designed an interface like this for a collaborative and interactive AR application where cards had to be matched based on their AR content. When a user brings a card into view, the corresponding AR object is displayed, making it visible for the user when looking through the AR device. When the card is moved or rotated, the virtual object follows.

To allow modifications of and interactions with virtual elements, markers do not necessarily all need to correspond to a virtual element but can also represent an action. Poupyrev et al. (Poupyrev et al., 2002) developed a tangible AR interface where marker cards were divided into data tiles, each containing a virtual element, and operation tiles, representing an action. Moving an operation tile next to a data tile causes the operation to be performed on the data tile. Another idea introduced in the same tangible interface is dynamically assigned markers. In most applications, each marker card has a predetermined virtual element assigned to it. However, to improve flexibility and reduce the number of marker cards needed, it is possible to make the user assign objects to cards dynamically. By providing one element that represents a virtual catalog and designing actions to copy objects from the catalog to marker cards, the user can select the object they need and assign those to empty markers.

There are also some disadvantages to tangible interfaces. Physical objects have their natural properties which can be difficult to change and therefore very limiting (Kato et al., 2000). Physical properties make tangible elements and their controls easy to use. However, they restrict the possibilities of AR. Furthermore, markers and other tangible objects always need a fair size for optimal tracking. This can result in a less convenient way of interacting and can be limiting for precise interactions.

2.1.2 Touch-less Interfaces

An alternative to tangible interfaces that can overcome some of the limitations of physical objects is touch-less interfaces. As the name suggests, touch-less interfaces are fully virtual and therefore do not require the user to touch any physical objects.

A common way to interact with a touch-less interface is through hand gestures. For example, a simple and intuitive hand gesture is to use one or two fingers as a pointer, similar to a mouse cursor in a normal desktop setup. An example of fingers being used as a virtual mouse is the AR football game developed by Lv et al. (Lv et al., 2015) who developed a finger and foot tracking method for AR applications, in which the player controls a goalkeeper's glove by moving their fingers. If virtual buttons are used, a selection command can be simulated by bending and extending the fingers, similar to a clicking motion when using a regular mouse. When more complex gestures than pointer motions can be recognized by the system, more possibilities open up for intuitive interaction. Benko et al. (Benko et al., 2004) presented a collaborative mixed reality tool for archaeology that includes a hand tracking glove that allows the user to grab a virtual object. Performing a grabbing motion attaches the object to their hand to be able to move it around to examine.

Because touch-less interfaces are fully virtual, there is no tactile or haptic feedback, which can be a limitation. Without this feedback, the user can only rely on their vision to position their hand to interact with a virtual object. This can be difficult because of the lack of occlusion of virtual elements by physical objects, causing a distortion in depth perception. Most AR devices do not support such occlusion. Furthermore, although hand gesture recognition is considered one of the most natural ways to interact in AR (Malik et al., 2002)(Kim and Dey, 2010), these techniques are relatively new and not yet fully optimised nor easily accessible. Recent developments in computer vision-based hand tracking and the implementation of this feature in AR devices, make hand gesture based interfaces very promising for the future. Currently, however, hand tracking is not robust enough, which poses considerable limitations.

2.2 AR Content Creation

Some research has been done on content creation in AR where new interface concepts were introduced. Shen et al. (Shen et al., 2010) created an AR product design application which includes 3D modeling and collaborative design activities. The main interaction tool for modeling is a virtual stylus that is controlled by two markers placed next to each other. The first marker is used for position tracking, the second marker is used as a selection mechanic. When the second marker is occluded, it registers as a button click, selecting what is currently at the tip of the stylus. Phan and Choo (Phan and Choo, 2010) designed an AR application for interior design with a similar interaction mechanic using markers and occlusion. Furniture can be arranged in a room using single markers representing a piece of furniture to track the position. Strips of markers placed next to each other can be used to change properties of the furniture. The property changes based on which marker is occluded.

Remarkably, much research on content creation in AR uses tangible interfaces, specifically markers. Not much research has been done on AR content creation using touch-less interfaces, even though touch-less interaction could be a better fit for this type of applications. A possible explanation for the lack of AR content creation research using touch-less interfaces is the accessibility and simplicity of marker tracking; it only requires some pieces of paper with printed patterns, a basic camera and an image processing algorithm that is relatively simple. As discussed above, touch-less interfaces typically use more advanced techniques like hand tracking. With markerless tracking techniques becoming more advanced and more common, as they are being integrated in AR devices, new interface tools become more reliable and powerful. This encourages further research on AR content creation using touch-less mechanics instead of only (or together with) tangible marker-based interfaces.

3 Interaction Design for Story Authoring

The advances discussed in the previous section recommend building upon existing techniques, capitalizing on existing research achievements, although they usually involve only one of the two types of AR interfaces, tangible or touch-less. Combining interactions from both types in a useful way may result in improved interface concepts. By carefully deciding what should be done by tangible interactions and what suits best with touch-less based on each type's strengths and weaknesses, each action can be assigned to the most fitting type.

3.1 Tangible Interactions

Tangibles are physical, which gives them the advantage of tactile feedback and familiar physical properties. A virtual element tied to a marker follows the physical movement of that marker. This makes tangibles a good fit for spacial actions. A spatial action is any action that relates to movement of a virtual element where placement in the 3D space is meaningful and has to be easily changeable. For a story authoring application, this translates to spatial placement of characters and objects in the scene. A character or object can be added to a marker and placed on the desired position in the scene by moving the marker to that position. Markers can be moved around freely to change the composition of characters and objects. They can also easily be moved out of, and brought back into, the work space, which is useful for recurring elements.

Other than virtual elements with direct spatial significance, a marker can also represent a more general

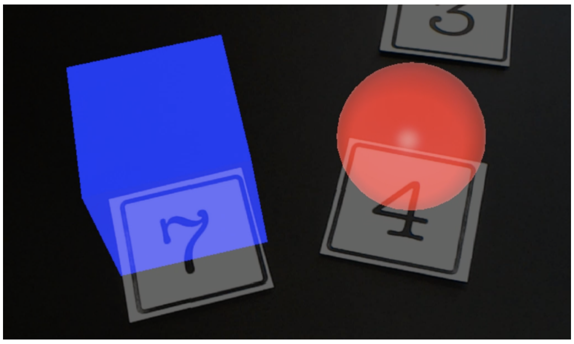

Figure 1: Virtual elements represented by markers

element or even operation that causes a global change without being positioned in the scene. A typical story authoring example is the environment where a plot point takes place: e.g. when authoring a scene that is supposed to take place in the kitchen, the marker representing the kitchen environment can be brought into view; where exactly this marker is shown or placed is irrelevant and, as soon as the marker is registered, the environment of the scene can be adjusted and visualized, after which the marker can be taken out of view.

Using markers for spatial actions and variable recurring elements means that, once each element is added to a marker, these operations only require the user to move markers around. Figure 1 shows an example of virtual elements represented by a marker.

3.2 Touch-less Interactions

When an interface has to offer a large amount of options that is unsuitable for one marker per element, a menu can be used to present all elements, so the user can choose which ones should be assigned to markers. A selection mechanic is then needed to enable the user to choose the desired option. Natural actions to indicate a choice are pointing and touching or pressing buttons. This can be done with touch-less interactions using virtual buttons and hand tracking to enable the user's physical hands to interact with the virtual buttons. By displaying the options as buttons on a virtual menu, the user simply needs to point to what they want and perform a motion similar to pressing a button. With virtual buttons there is, of course, no tactile feedback. To compensate, visual cues can be added to the virtual buttons, like movement to simulate the pressing of a physical button and change of color to show selection.

Story authoring involves selection of various elements like characters, objects, actions, environments, properties and more. It seems, therefore, natural to use a menu to display all such options, with the touch-less selection mechanic described above, allowing the author to quickly select story components from a wide range of choices.

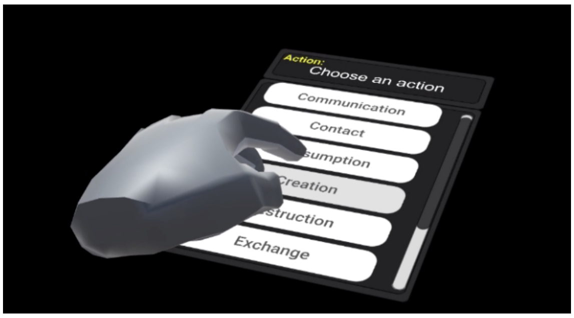

Figure 2: A virtual representation of a physical hand interacting with virtual buttons. Ideally, the user can use their natural hands to interact with buttons, without a virtual representation.

While tangibles are suitable for spatial actions and variable recurring elements, touch-less interactions can be used for interface elements that are static and do not have spatial relevance. This includes general operations that can be translated to a button, a gesture or an interactive visualisation. A good story authoring example is the storyline, a structural element that should be available to the author for browsing and editing the story. It would not make much sense for the author to attach the storyline or plot points to a marker. Instead, operations involving the storyline, like adding a new plot point, browsing through plot points or going back to a previous plot point to modify it, can be implemented using similar concepts to the selection mechanic described above.

Many different operations can be implemented through virtual buttons, keeping interactions consistent by using one single selection mechanic. Buttons can be placed anywhere in the 3D space, or 2D space when placed on the tabletop surface, possibly grouped together by functionality for convenience. An example of a virtual representation of a physical hand interacting with buttons can be seen in Figure 2.

To summarize, for all actions where location in the 3D space matters and all actions that are configurable and recurring, tangible interactions can be used. All actions involving selection and general control operations where location is not important can be done using touch-less interactions.

4 The Story ARtist Application

To assess how AR can be used for story authoring using the interaction design presented in the previous section, we developed Story ARtist, a prototype AR application for creating simple linear stories. To author a story, the author can successively create (and edit) plot points, and (for each plot point) select actions and assign characters, objects and environments to markers.

4.1 Plot Point Structure

In this context, a story line consists of a sequence of plot points, each one representing a single action. When a new plot point is created, an action is chosen and the plot point needs to be filled with information related to that action.

4.1.1 Actions and Arguments

An action is the main verb that represents what happens in the plot point. Authoring a story verb by verb could be tedious and would require the author to define many separate elementary plot points. To avoid this and for the sake of simplicity, the actions chosen for the prototype are descriptive verbs that encompass multiple ’smaller’ verbs, which would be required to complete the action. An example is the verb *give*: in the application, this represents not just the action of a character handing over an object to another character; instead, it includes the first character collecting the object, moving to the second character, handing it over and the second character receiving it.

Once an action is chosen, the interface requests what is needed in the scene to author the chosen action, called its *arguments*. For example, when the author chooses the action *greet*, there need to be two characters in the scene where one character will greet the other. The interface keeps track of which arguments are already specified in the scene, and for which ones an element needs to be added.

4.1.2 Scene

A plot point does not only contain an action and its arguments, but also a *scene*, in which 3D models of the arguments are present and can be arranged at will using their markers, to visualise what the authored plot point should look like. This is a static representation of the action’s arguments that can be seen as a snapshot of the story. When the author goes to a next plot point, i.e. when the plot point is complete and the author is happy with the arrangement, the locations of the (markers representing the) 3D models in the scene are saved in the application together with the action and arguments.

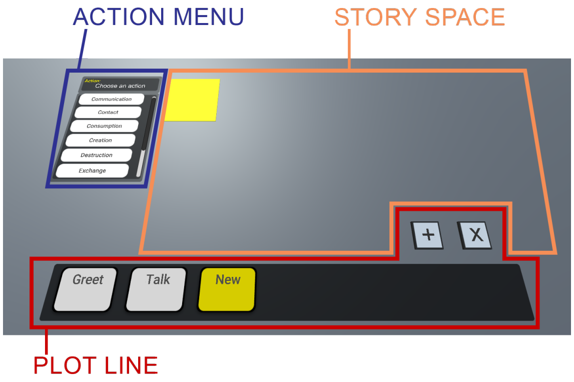

Figure 3: Overview of the Story ARtist work area and interface

4.1.3 Plot Line

To keep the focus of the Story ARtist application on the AR interaction, only linear plot lines can be authored and no narrative consistency constraints are presently handled. The story is displayed as a linear sequence of plot points called the *plot line*. Selecting an existing plot point opens it, which allows the author to look at (and possibly modify) the authored scene with the chosen action, elements and scene composition. A new action can be chosen, the environment can be changed and characters and objects can be moved, added or replaced.

4.2 Interface

Figure 3 displays an overview of the Story ARtist interface, with its three main sectors. In the top left corner, the *action menu* can be found. It can be used to choose an action for each plot point. When an action is chosen, it is assigned to the plot point that is opened at that moment and the menu changes to display the action’s arguments, so the author knows what needs to be added to the scene. This menu implements the touch-less design where the user can use their hands to press virtual buttons for selection.

The *plot line* displayed at the bottom of the application work area contains each authored plot point as a button that can be selected for editing. Above the plot line are some buttons corresponding to different plot line related operations like adding and deleting plot points. All these buttons follow the touch-less design guideline discussed above.

The rest of the application work area is the *story space*. This is where story elements like characters and objects can be placed in the scene and arranged to configure and visualise the plot point. Adding characters or objects to a scene is a spatial action because the placement of the element in the scene matters; and

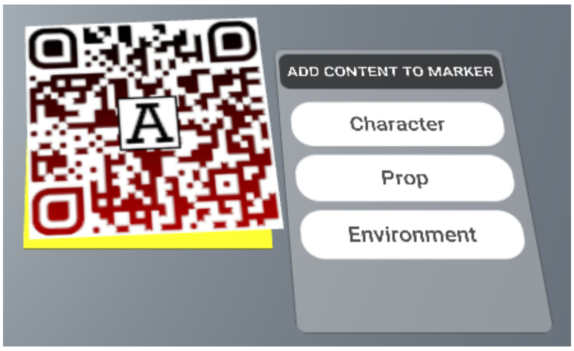

Figure 4: Marker programming space with marker placed on top of it. The menu appeared next to the marker to add content to it.

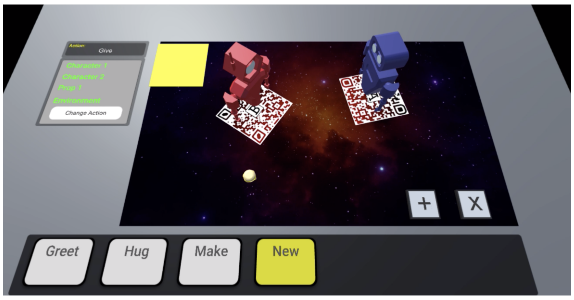

Figure 5: Example of a fully authored scene with a chosen action and elements added to the scene.

choosing the desired environment for a scene sets a recurring element for plot points. Therefore, both types of interactions are done using markers. To enable the author to choose between a wide variety of story elements like characters and objects, all markers that can be used in the application are variable, i.e. do not represent any content when starting the application.

The story space contains a *marker programming space*, visualised as a yellow square, which can be used to assign content to a marker. When a marker is placed on that spot, a menu pops up that can be used to select an element to assign to that marker, as shown in Figure 4. By programming the desired characters and objects onto markers and placing them at desired places in the scene, plot points can be filled with story elements. In contrast, the semantics of assigning a scene environment to a marker is different: it is meant to affect the entire story space, possibly for several plot points. Therefore, it suffices to show an environment marker anywhere in the application work area, as it immediately gets registered as the current plot environment, and that location is visualized accordingly. Figure 5 depicts an example of a plot point being authored, in which a robot is giving an object to another robot. This plot point is part of a larger example story that has been described elsewhere (Kegeleers and Bidarra, 2020).

5 Implementation

After describing the front-end side of the application in the previous section, this section focuses on the back-end side, describing everything needed to run the application and how the application works internally. The setup will be discussed in terms of hardware and software, followed by a description of the framework used to internally represent the story and store it in data files.

5.1 Setup

5.1.1 Hardware

An important hardware requirement for this project was to use a head-mounted AR device, so as to keep the author's hands free and enable better interaction. The available AR device for this research was the Microsoft Hololens, first generation.

The Hololens has a regular camera built in that can be used for marker tracking. However, this was not optimal, as markers had to be held close to the camera for a considerable amount of time in order for them to be recognized. To solve this problem, an external camera was added, a Logitech C920 Full HD Webcam. The camera is placed above the table, facing down.

The interface designed above includes hand tracking. By default, the Hololens device supports very little hand tracking and gestures. To avoid being limited by the lack of touch-less interaction supported by the Hololens, Ultraleap's Leap Motion Controller was added to the setup. This is a small device specifically designed for hand tracking. The Leap Motion was attached to the external camera above the table, also facing down.

Currently, a serious limitation of head-mounted AR devices is the field of view (FOV). The Hololens' FOV is 30 by 17 degrees, which is a considerably small portion of our natural FOV. To counter this, tabletop projection was added to the setup: a projector was placed on the ceiling, facing down to the table so a projection overlay could be displayed on the tabletop environment.

All devices were connected to a PC running the application: an Alienware Aurora 2012 PC, with a GeForce GTX 590 graphics card. The Hololens uses a wireless connection to communicate with the application running on the PC. To improve this connection, an Asus AC3100 WiFi adapter was added.

5.1.2 Software

The application was built in Unity 2018.4 . This is a Unity version with long-term support that many libraries support as well, including the libraries needed for the hardware used in this project.

Ultraleap provides different libraries to integrate their hand tracking software into a Unity project. For this project, the core SDK for Unity was used together with the interaction engine, which enables interaction with custom UI elements.

For tangible interactions, a library was needed to track markers. This functionality was added using the Vuforia Engine, an AR library that tracks markers and objects using image processing. The Vuforia engine was configured to run on the frames provided by the external camera.

Data management for both story assets and authored stories as a result of the application was done using JSON (JavaScript Object Notation). The Unity asset *JSON .NET for Unity* was added to the project for easy integration of JSON parsing into scripts.

5.2 Story Framework

The story framework consists of two modules: (i) the internal representation of the story that is used to store plot points, so that the authored data can be exported in a readable format for external use; and (ii) the representation that is application specific and is mainly used to visualise the story.

5.2.1 Representation of the Story

Story ARtist is a prototype application developed specifically for creating a simple linear narrative that can be used as a baseline for a story. To enable further development of the story and its elements, a common framework was used as inspiration for the story representation in the application and a widely used format was chosen to store the plot line in a file.

Many applications that involve some kind of narrative use actions, or a grouping of actions, as base units for the story. Actions, represented by verbs, are a good identifier for story events, as they are often unique and easily characterized. Therefore, verbs were chosen as a base for each story event, here called a plot point. Each plot point is assigned a singular action that determines which content is required in order for the plot point to make sense related to the action. What elements each action requires is defined by its predicates. In this story representation, predicates are the subdivisions of the main verb/action, each representing a subevent that is part of the main action. When combined, they form the entire main action. For example, the predicates of the verb *give*, where one character gives an object to another character, would be *grabbing the object*, *moving to the other character* and *transferring the object*.

This representation of actions, predicates and arguments was inspired by VerbNet (Schuler, 2005), an English verb lexicon. VerbNet is often used for natural language processing but can also be used for story-related contexts like computational storytelling (Kybartas and Bidarra, 2015) (Kybartas and Bidarra, 2016). Two recently presented updates to VerbNet on generative lexicon event structures (Brown et al., 2018) and subevent semantics of transfer verbs (Brown et al., 2019) describe the use of predicates to divide verbs into more specific fragments, defining its subevents in detail. This representation was adapted to fit the story ARtist framework, as described above.

To enable easy export of the authored story for external use, the JSON format is used, when writing the plot points to a text file. Plot points are stored as a collection of the chosen action, the filled arguments, the environment and other elements in the scene. Using the JSON format together with the VerbNet-inspired structure for defining an action's arguments, enables easy conversion of the authored plot line to other tools, especially tools using the VerbNet representation.

5.2.2 Visualisation of the Story

Apart from enabling use in other tools, predicates can be very useful in an adaptive framework for the animated visualisation of the story. For every predicate, an animation can be programmed. As described in section 4.1, the verbs used in the Story ARtist application are used as global actions where the framework automatically includes smaller subdivisions of the verb, i.e. predicates, to make the action work. Because of this, verbs often contain the same predicates. This way, a list of different predicates can be programmed which can then be used to compose a wide variety of actions to include in the application and allows for easy adaption of the actions, including addition of new options.

6 Evaluation

Using the Story ARtist application and its setup, a user study was conducted to evaluate AR for story authoring with the proposed interaction techniques. A formative evaluation producing mostly qualitative data was chosen because it best suits the exploratory nature of this research and gives a good insight into participant feedback and ideas.

6.1 User Study

A total of 20 participants came by, each for an individual session of 45 minutes. All participants were students, recent graduates or postdocs. The session included an introduction to the application in the form of a narrated tutorial, a task-based interaction phase with the application, an interview and a questionnaire. Each participant interacted with the application for 10 to 20 minutes during the tutorial and interaction phase. During application interaction, the participants were observed and the graphics display of the application on the PC monitor was recorded.

Upon starting the tutorial, the participants were asked to only focus on the application and its authoring functionality, rather than on the 'known issues' caused by technology failures, because those are not related to this research. During the tutorial, participants were guided by the observer through the operations required to author one plot point.

After the tutorial, participants were encouraged to further explore the application by authoring some more plot points by themselves. A list of all operations the participant was required to perform was kept by the observer to make sure every participant tried all significant functionality related to story authoring. Once the participant was done adding plot points, they were asked to try out the functionality they did not explore yet.

When the participant was done interacting with the application, the interview followed. The interview was semi-structured with mostly open questions to explore participants' opinions without being restricted by options. A list of 11 interview questions was composed which focused on the main aspects to be evaluated, i.e. interaction, visualisation and overall impression. The interview questions can be found in the Appendix. Each participant was also asked to describe what kind of previous experience they had with augmented and virtual reality.

Finally, the participants were given a questionnaire in the form of an online survey that could be completed via smartphone. The chosen questionnaire was the System Usability Scale (SUS) (Brooke, 1996), to evaluate the application's usability. More specifically, the updated version by Bangor et al. (Bangor et al., 2008) was used.

6.2 Results

To process the qualitative data from the video recordings and the interviews and draw the appropriate conclusions, the affinity diagram method for evaluating interactive prototypes developed by Lucero was used (Lucero, 2015). Observations from the recordings and interview answers were converted into notes and classified into clusters using a digital infinity diagram to form clusters to identify common data patterns and ideas. Out of a total of 580 notes, 7 clusters emerged. An overview of the affinity diagram can be seen in Figure 6. Below, the clusters and sub-clusters containing notes on interaction and visualisation are discussed along with some common general notions.

6.2.1 Interaction

The largest amount of notes were classified into the Interaction cluster. From these notes, it was clear that a majority, 14 out of 20 participants (14/20), found the combination of tangible and touch-less interaction useful. People expressed that having markers for placement and hand tracking for selection is a good combination, that it belongs together and it does not create any disconnection. Out of these 14 people, 4 noted that if hand tracking would work perfectly, markers would no longer be necessary. The people who did not find the combination favorable, expressed that they would prefer hand interaction only or the addition of speech.

Even though 14 people liked the interaction combination, more participants (17/20) noticed the advantages of markers. Many remarks were made about the ease of use to position and rotate elements, with some participants relating it to the physical aspect of markers. An operation that was unclear to many people was removing an element out of a scene by taking the marker out of the application work space. Only 3 people took markers out of the scene without instruction from the observer and 5 people expressed that it was unclear elements could be removed from the scene by removing the marker. Another disadvantage mentioned by 5 people was that there was no way of knowing what was programmed on which marker when a marker was out of the AR device's view or not being tracked. This made people forget what was on each marker.

Specifically for hand interaction, a majority (16/20) explicitly mentioned liking the hand interaction, describing it as intuitive, natural and easy to use. Some participants, including the 4 not endorsing the hand interaction, mentioned preferring hand interaction in a 3D space or more complex gestures instead of only clicking buttons on the table.

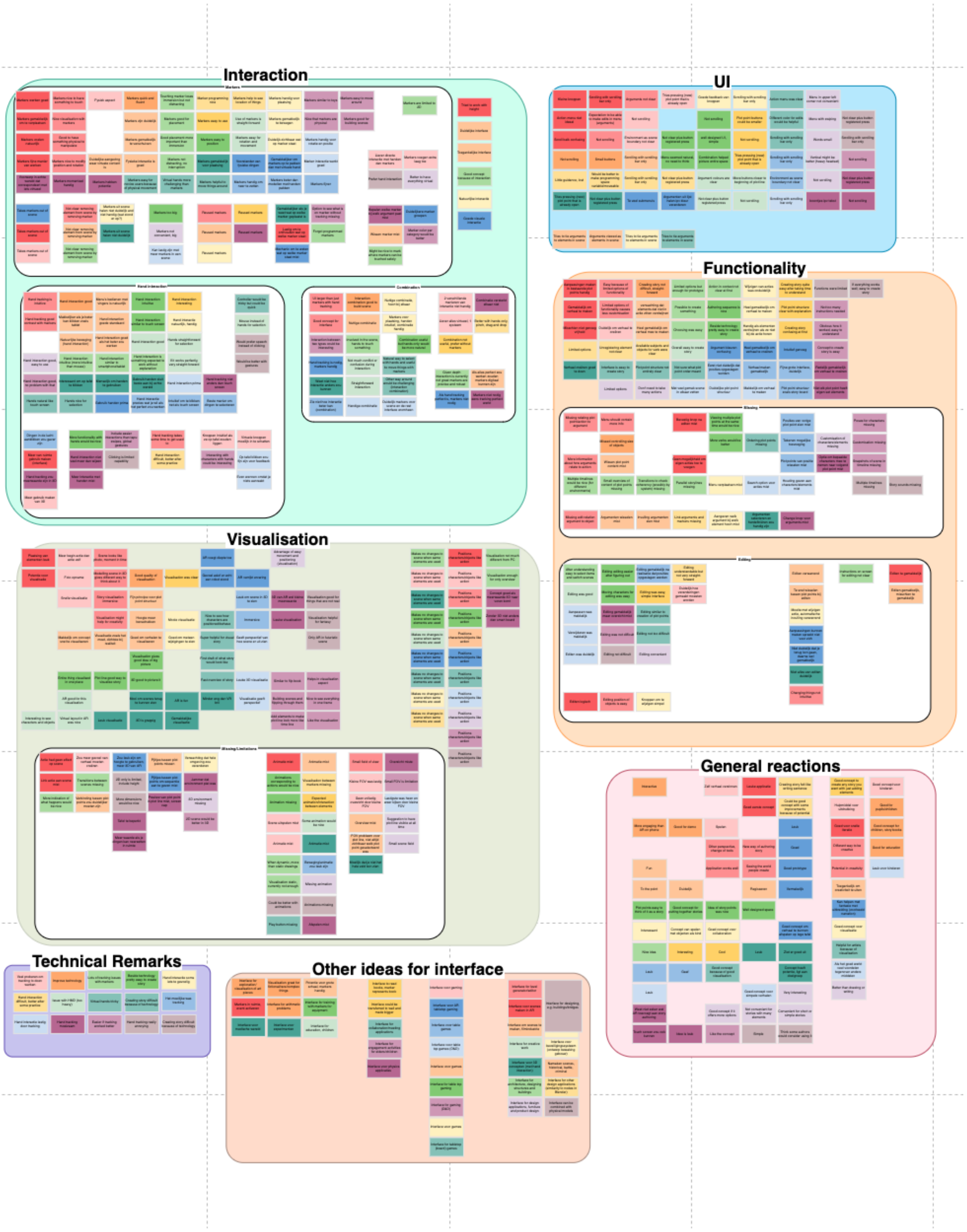

Figure 6: An overview of the final affinity diagram containing all clustered notes from the video recording observations and interview answers.

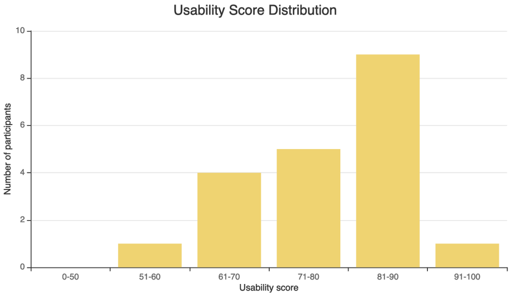

Figure 7: Distribution of the usability scores over participants.

6.2.2 Visualisation

Almost all participants (18/20) expressed positive opinions about the AR visualisation. Many mentioned that the 3D aspect of AR is a good way to picture a scene. However, barely half of the participants (11/20) utilised the scene visualisation as intended. This could be seen in the video footage where only 11 people positioned their characters and objects in a way that made sense related to their chosen actions (e.g. characters facing each other for the action *greet*), while others only put each marker next to the other.

Many participants (12/20) reported missing some animations to watch the story play out. Some other limitations mentioned were the 2D environment and not being able to see transitions between plot points.

6.2.3 General Notions

Regarding the general task of authoring a story using the Story ARtist application, a vast majority (17/20) indicated this to be easy. Some reasons given were the natural and intuitive interface, the visualisation of everything in one place, straightforward interaction and the small amount of actions required. Some mentioned that the application itself is easy but the particular technology used made it harder.

6.2.4 System Usability Scale

Following the SUS processing method, all scores from the questionnaire were processed resulting in a usability score per participant between 0 and 100. The distribution of the usability score over participants can be seen in Figure 7, which shows that most participants gave the application a usability score between 81 and 90. The average usability score is 78.62 with a standard deviation of 9.94.

7 Discussion

General reactions were mostly positive, both when asked for some first impressions and throughout the entire interview. However, despite asking participants to focus on the application only, more than half expressed frustrations about the technology during the interview. During development and testing, it was clear that the available technology did not suffice to make the application work smoothly. This led to the integration of all the hardware mentioned in Subsection 5.1, which in turn, introduced performance and reliability issues.

Because 14 out of 20 participants valued the combination of hand interaction and markers, this can be seen as a favorable and encouraging signal. Many understand the value of the physical aspect of markers, especially for placement. Participants with significant AR and VR experience, who had tried fully touchless interfaces before, even expressed the current difficulty of picking up virtual objects using hand interaction only. Without this experience, it is understandable for people to think that hand interaction alone would be more convenient, especially if marker tracking is not working perfectly, which was the case in this prototype. However, hand tracking was not accurate enough in the setup either, and participants had to use a virtual representation of their hands that did not align perfectly with their physical hands to interact with buttons, as shown in Figure 2. This might have given people a more favorable impression on the use of markers.

The Story ARtist application only includes tapping and sliding as hand interaction which is why a number of participants mentioned a lack of gestures and interaction in the 3D space. The choice of including only basic hand interaction was made for simplicity. In order to include more complex hand interaction, additional research would be needed, to find the right balance between interaction and simplicity. Some participants suggested combining input speech as well, which is another interaction modality deserving to be investigated.

8 Conclusion

Content creation applications have predominantly used conventional methods and devices, often lacking immersion and direct interaction. Augmented Reality has the potential to overcome these limitations.

To evaluate the usefulness of AR for story authoring, we analyzed the pros and cons of existing AR interaction types, tangible and touch-less, and explored the combination of both into a single natural user interface. To assess this proposal, Story ARtist, an AR prototype application was developed, featuring hand gestures as touch-less interaction for selection, and markers as tangible interaction for programming narrative elements and placing them in the scene.

Results from a user study showed that this combination was quite favorable, with participants appreciating the physical aspect of the markers and the intuitiveness of the hand interaction. Moreover, the evaluation results confirm that the interactive 3D visualisation significantly improves the story authoring experience. We can therefore conclude that there is a considerable potential in using AR for story authoring.

Future research should use an AR setup with more powerful and well-integrated devices, to provide more convenient and effective interaction. Furthermore, other interaction techniques could be investigated, including more elaborate hand interaction, as well as its integration with speech input.

REFERENCES

Bangor, A., Kortum, P. T., and Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594.

Benko, H., Ishak, E. W., and Feiner, S. (2004). Collaborative mixed reality visualization of an archaeological excavation. In *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 132–140. IEEE.

Brooke, J. (1996). Sus: a "quick and dirty'usability. *Usability evaluation in industry*, page 189.

Brown, S. W., Bonn, J., Gung, J., Zaenen, A., Pustejovsky, J., and Palmer, M. (2019). Verbnet representations: Subevent semantics for transfer verbs. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 154–163.

Brown, S. W., Pustejovsky, J., Zaenen, A., and Palmer, M. (2018). Integrating generative lexicon event structures into verbnet. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Castano, O., Kybartas, B., and Bidarra, R. (2016). TaleBox - a mobile game for mixed-initiative story creation. In *Proceedings of DiGRA-FDG 2016 - First Joint International Conference of DiGRA and FDG*.

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

Kegeleers, M. (2020). Story ARtist: Story authoring in augmented reality. Master's thesis, Delft University of Technology, The Netherlands.

Kegeleers, M. and Bidarra, R. (2020). Story ARtist. In *Proceedings of IEEE AIVR 2020 - 3rd International Conference on Artificial Intelligence & Virtual Reality*.

Kim, S. and Dey, A. K. (2010). Ar interfacing with prototype 3d applications based on user-centered interactivity. *Computer-Aided Design*, 42(5):373–386.

Kybartas, B. and Bidarra, R. (2015). A semantic foundation for mixed-initiative computational storytelling. In *International Conference on Interactive Digital Storytelling*, pages 162–169. Springer.

Kybartas, B. and Bidarra, R. (2016). A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(3):239–253.

Lucero, A. (2015). Using affinity diagrams to evaluate interactive prototypes. In *IFIP Conference on Human-Computer Interaction*, pages 231–248. Springer.

Lv, Z., Halawani, A., Feng, S., Ur Réhman, S., and Li, H. (2015). Touch-less interactive augmented reality game on vision-based wearable device. *Personal and Ubiquitous Computing*, 19(3-4):551–567.

Malik, S., McDonald, C., and Roth, G. (2002). Hand tracking for interactive pattern-based augmented reality. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, page 117. IEEE Computer Society.

Phan, V. T. and Choo, S. Y. (2010). Interior design in augmented reality environment. *International Journal of Computer Applications*, 5(5):16–21.

Poupyrev, I., Tan, D. S., Billinghurst, M., Kato, H., Regenbrecht, H., and Tetsutani, N. (2002). Developing a generic augmented-reality interface. *Computer*, 35(3):44–50.

Preece, J., Sharp, H., and Rogers, Y. (2015). *Interaction design: beyond human-computer interaction*. John Wiley & Sons.

Schuler, K. K. (2005). *VerbNet: A broad-coverage, comprehensive verb lexicon*. PhD thesis, University of Pennsylvania.

Shen, Y., Ong, S., and Nee, A. Y. (2010). Augmented reality for collaborative product design and development. *Design studies*, 31(2):118–145.

Zhou, F., Duh, H. B.-L., and Billinghurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *Proceedings of the 7th IEEE/ACM international symposium on mixed and augmented reality*, pages 193–202. IEEE Computer Society.

APPENDIX - Interview Questions

List of questions asked to participants during the interview of the user study.

- What did you think of the application?
- What did you think of the use of the markers?
- What did you think of using your hands for selection?
- Did you find the combination of hand tracking and markers useful? Why (not)?
- How easy or difficult was it to use the interface to create a story? Why?
- How easy or difficult was it to edit your created story?
- Was the plot point structure clear? What would you suggest to improve it?
- What are your thoughts on how the story was visualised?
- Do you think this application is a good concept for story authoring? Why?
- Is there any functionality that you missed?
- Do you think this interface could be used for other AR applications with other goals and domains? If so, can you think of an example?