Journal of Universal Computer Science, vol. 26, no. 8 (2020), 947-971
submitted: 11/1/20, accepted: 8/7/20, appeared: 28/8/20

Bite-Sized Virtual Reality Learning Applications: A Pattern-Based Immersive Authoring Environment

Robin Horst
(RheinMain University of Applied Sciences, Germany
Robin.Horst@hs-rm.de)

Ramtin Naraghi-Taghi-Off
(RheinMain University of Applied Sciences, Germany
Ramtin.Naraghi-Taghi-Off@student.hs-rm.de)

Linda Rau
(RheinMain University of Applied Sciences, Germany
Linda.Rau@hs-rm.de)

Ralf Dörner
(RheinMain University of Applied Sciences, Germany
Ralf.Dörner@hs-rm.de)

Abstract: Bite-sized learning is a current educational trend in which educators divide content into relatively small, easily comprehensible chunks, called nuggets. In this paper, we introduce an authoring toolkit that relies on VR implementation of nuggets and show that a nugget-based approach is also facilitating the authoring of VR learning content. In particular, we present *Immersive Nugget Tiles* (IN-Tiles), a novel authoring toolkit aimed at authors who are not experts in VR. With IN-Tiles, manipulating VR nuggets and authoring VR learning content can be directly accomplished within a virtual environment allowing authors to immediately experience the results of their authoring efforts in VR. We discuss the underlying concepts of IN-Tiles, specifically how to visualize VR nuggets in a virtual environment and how to present affordances that support authoring and manipulating VR nuggets. We report the results of a user study where we evaluated the IN-Tiles toolkit and compared it to a conventional 2D authoring environment that also relies on component-based VR. The results support the hypothesis that nugget-based immersive authoring tools are suitable to create bite-sized VR applications successfully and that authoring directly in VR has an added value particularly for authors who are no IT specialists.
Key Words: Virtual Reality, Pattern-Based Authoring, Bite-Sized Learning, Reusable Software, VR Nuggets, E-Learning, Game Engineering Software
Category: H.5.2, D.1.7, D.2.13, L.3.0, L.2

1 Introduction

Microlearning [Hug 2005] is a novel paradigm that has been successfully used in the e-learning domain [Polsani 2003]. The fundamental idea behind the microlearning methodology is to break down complex educational structures (e.g., lessons, modules, courses) into basic and easily comprehensible chunks – called *learning nuggets*. Educators can convey the content to learners piece by piece rather than presenting them bulky content at once and therefore reduce the cognitive load. This method is also referred to as bite-sized learning.

The microlearning methodology can also reflect more complex educational structures. Authors create their lessons by arranging nuggets so that each lesson consists entirely of its underlying nuggets. Authors can easily adapt existing lessons by rearranging nuggets, for example, to adapt to the prior knowledge of the audience. A learning nugget can be realized using different media. While much modern media have already been used [Hug 2005], Virtual Reality (VR) is not established in this context, yet. Nugget-based VR would offer a solution for seamlessly introducing VR to those parts of a lesson where it offers an added value justifying the overhead and challenges associated with VR.

A challenge for VR authoring toolkits, in general, is that authors may not have a feeling for the interactive 3D application that they are creating – especially for authors that are not experts in the fields of VR or IT (laymen authors). What-you-see-is-what-you-get (WYSIWYG) interfaces can help to overcome this challenge. Here, the immediate experience of the environment that is created is necessary. Authors can frequently switch back and forth between the current state of the application and the authoring process to experience the application through the eyes of the VR users (i.e., the learners). Immersive authoring technologies (e.g., [Lee et al. 2004, Takala 2014, Dunk, Haffegee, and Alexandrov 2010, Jee et al. 2014]) can be utilized for seamlessly supporting a WYSIWYG authoring approach. However, such immersive authoring has not been explored in the context of creating bite-sized content and learning nuggets yet. One challenge for accomplishing this is that an immersive authoring toolkit cannot be easily derived from conventional 2D authoring toolkits. In our case, nuggets and application structures must be visualized within a virtual environment. This environment must provide affordances to authors that allow authoring actions such as adapting nuggets or connecting them to form larger application structures.

In this paper, we make the following contributions:

- We introduce Immersive Nugget Tiles (IN-Tiles) – a novel authoring toolkit aimed at authors who are inexperienced in VR. It relies on a pattern-based VR implementation of nuggets. It allows VR authoring without the need to step outside a virtual environment during the content creation process. We discuss the underlying concepts of IN-Tiles, specifically our concepts on how to visually represent learning nuggets and application structures. We show how to facilitate authoring activities by providing according affordances.

- Based on the results of our user study, we show that an immersive nugget-based approach is facilitating the authoring of VR learning content for laymen authors. Furthermore, our IN-Tiles toolkit is compared to a conventional 2D authoring environment that also relies on component-based VR.

This paper is organized as follows. The next section discusses related work. In Section 3, we present our pattern-based authoring concept that dovetails the learning nugget approach and enables authors to create small VR applications within an immersive environment. In the fourth section, we describe the implementation of the toolkit. Next, we report and discuss the evaluation of IN-Tiles. Section 6 provides a conclusion and points out directions for future work.

2 Related Work

We present related work about microlearning, authoring, and specifically VR nuggets and a conventional 2D authoring application that is nugget-based.

2.1 VR and Microlearning

It has already been demonstrated over the last decades that VR can effectively support educational purposes in general (e.g., [Bailenson et al. 2008]). For example, while exclusively text-based learning can lead to boredom comparatively fast [Zhang et al. 2004], VR has the potential to communicate learning content in a vividly and interestingly way [Monahan, McArdle, and Bertolotto 2008]. Regarding the field of microlearning, related research focuses on mobile learning in virtual environments and augmented reality (AR) applications. Current work by Beutner et al. [Beutner and Pechuel 2017] and Souza et al. [Souza and do Amaral 2014] discusses the use of micro-content and micro-units respectively in mobile learning scenarios. They outline next-generation mobile devices that deliver content to learners in a small scope, closely related to learning nuggets. Beaudin et al. [Beaudin et al. 2007] explore AR as a medium to transport small pieces of content but do not mention microlearning.

Work by Davies et al. [Davies, Crohn, and Treadgold 2019] examines how the medium of VR can be integrated into short and group-based learning sessions at universities. They describe a learning scenario and a small VR application that can be utilized similar to nuggets. Horst and Dörner [Horst and Dörner 2019a] describe a bite-sized authoring process that makes use of patterns to create VR applications for what they call knowledge demonstration – a direct teaching scenario using small VR experiences. They point out a general model for this process but do not address immersive authoring that would enable laymen authors to create the VR content themselves. Their paper still identifies benefits that can be obtained by bringing together VR and microlearning methodology. Their paper also motivates the exploration of pattern-based authoring interfaces for creating bite-sized VR applications.

2.2 VR Authoring and Pattern-Based Approaches

VR authoring is a complex process and still involves many challenges throughout the last decades of VR research [Dörner, Kallmann, and Huang 2015, Steed 2008]. Pioneering work by Bierbaum et al. [Bierbaum et al. 2001] presents VRJuggler – a framework for VR development. It offers a runtime environment that includes an abstraction layer for devices and scene graphs. It also allows configuration changes at runtime.

Schmalstieg et al. [Schmalstieg et al. 2002] propose the Studierstube system. It is a presentation system that also serves as an authoring system for custom-made AR applications. As Studierstube uses AR technology, it blends virtual content with the real world so that presenters and the audience may see each other during presentations and communicate in a natural way. Rather than building upon methodologies from application domains (e.g., microlearning in educational applications), it serves as a basic technology for general application scenarios.

The AMIRE framework by Dörner [Dörner et al. 2003] describes an approach to include domain experts within the authoring workflow. The contribution does not target VR as a medium but focuses on mixed reality application development. Time- or size-related constraints for VR experiences which are common for bite-sized learning are not targeted in this work.

The authors of InTml [Figueroa et al. 2008, Figueroa et al. 2012] describe a higher-level programming language for rapid prototyping of virtual environments. InTml is an XML-based application that includes geometry in X3D. It also provides the possibility to hide unnecessary information during the authoring process. It facilitates the authoring for designers, so they can concentrate on the actual task that the VR application should fulfill rather than the technical details. They provide a visual editor for the development that includes visual programming. The concept does not expect the authors to be experts in programming, however, it still expects them to have profound knowledge of VR development in general.

All the above-mentioned systems use non-immersive authoring tools that are based on 2D desktop PCs. Work by Lee et al. [Lee et al. 2004] proposes the term 'immersive authoring' as a category of their AR authoring application. It carries out authoring tasks within an AR application itself. They describe that the development and the testing of the application can be done concurrently throughout the entire development. It is stated that this is a major advantage of using immersive authoring. Jee et al. [Jee et al. 2014] focus on an immersive authoring toolkit to develop specifically educational AR applications, whereas Dunk et al. [Dunk, Haffegee, and Alexandrov 2010] describe a set of immersive authoring interactions that transfer this approach to CAVE-like VR systems. Takala [Takala 2014] investigates immersive authoring user interfaces that rely on low-budget hardware and that can be used by laymen authors. The Unity game engine is used to let users develop VR applications with predefined building blocks. These blocks do not rely on domain-specific application patterns, but the author rather reflects certain technical VR aspects for creating spatial interfaces.

Work by Klinker et al. [Klinker et al. 2002] proposes a component-based approach for creating AR applications. In expert interviews, they pointed out recurring scenarios in the domain of automotive design presentation and implemented them in an AR application. By using these domain-specific scenarios as a basis, they indicate that experts can familiarly use the AR technology. The presented 'Fata Morgana' system is the only one of the mentioned systems that considers certain patterns as the foundation for application development. Current work indicates that pattern-based authoring approaches can be beneficial for usage by domain experts.

2.3 Virtual Reality Nuggets

The concept of VR nuggets was recently introduced by Horst and Dörner [Horst and Dörner 2019b, Horst and Dörner 2019c]. It is an authoring concept that dovetails the nugget paradigm from the educational domain in small and self-contained VR systems that are based on patterns from the educational domain. Each of these VR systems is divided into three major parts: (1) *VR nuggets* reflect the pattern-specific content, (2) *functional coatings* can customize VR nuggets by adding optional functionality to it, and (3) the *nugget platform* combines runtime functionality for VR nuggets. The latter handles functionality such as data handling, graphics rendering and display, hardware control, etc.

VR nuggets provide authors with VR software that implement the underlying pattern completely. Relying on defaults that can be changed later, authors can customize the VR software while the software is always in an executable state. Authors simply must replace placeholder objects during the authoring process (e.g., the small orange cube in Fig. 1 with a meaningful 3D model).

Each VR nugget represents a single pattern. This representation allows domain experts, such as educators, to use VR in a nugget-like way when structuring their learning content, e.g., in a lesson. Existing nuggets that may be implemented in another medium, such as a video or a graphic, can easily be exchanged. There exist several basic microlearning patterns suitable for VR identified by Horst and Dörner [Horst and Dörner 2019a]. For example, a *compare* pattern visually compares two objects of interest, a *chronological sequence* illustrates a temporal relation and a *setting* pattern shows an object of interest within a typical surrounding. A *puzzle* pattern divides an object into several sub-objects and lets learners put them together to learn about the structure of an object. In the following, we choose the *show and tell* pattern as an example

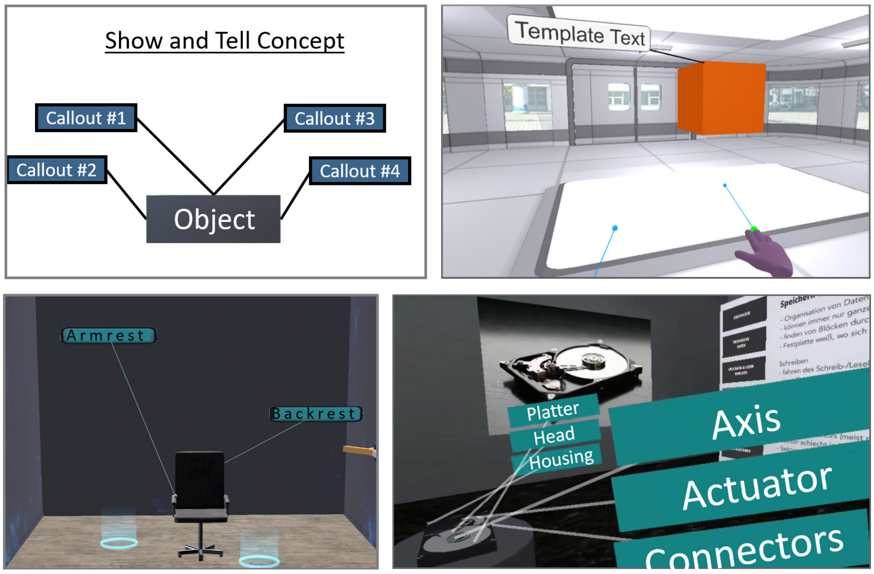

Figure 1: *The show and tell pattern (top left) and three implementations of it.*

throughout the paper (Fig. 1). It is a pattern that annotates a 3D object with callouts – short text strings that are connected to the 3D object with a line.

Generally, VR nuggets provide two mechanisms to customize them. (1) Each VR nugget has a set of pattern-specific parameters that are specified by the creator of the VR nugget. In the case of the show and tell pattern, the work by Horst and Dörner [Horst and Dörner 2019b] establishes mandatory parameters as exactly one main object to annotate and at least one callout. All parameters have a default value (placeholder). During the authoring process, the authors can decide to either keep a default value of a parameter or to customize it. For instance, in the case of the show and tell VR nugget, authors can replace the main object with their own choice, alter the number of callouts, or change the text on the labels. Other VR nuggets can have different parameters. (2) The concept of functional coatings offers further possibilities for authors to customize a VR nugget. It brings optional functionalities to a VR nugget. These predefined functionalities are optional in the way that the VR nuggets can successfully be used without any coatings, but they may be beneficial in certain situations. In the case of the show and tell nugget, a coating may add atmospheric sounds/music, but it may also enable the learners to grab the main object and rotate it. The latter can be helpful when the annotated object is rather small, but counterproductive if the main object is large. At last, a nugget system is restricted to only have one nugget in an active state to simplify their usage for laymen. Authors only have to focus on one nugget during authoring or later usage.

2.4 VR Forge

The VR Forge [Horst and Dörner 2019b] is inspired by conventional slide-based presentation programs such as PowerPoint. The interface consists of six menus, illustrated in Fig. 2. The *main menu* (top left) provides an interface for general functionality. Here, authors can add new nuggets or delete existing ones and start the presentation or close the application. Upon adding a new nugget, they are prompted to choose from initial VR nuggets that the system provides. The *demo timeline* (left) shows previews of selected VR nuggets and brings them in a linear order of execution. The *current nugget menu* (middle) displays the currently edited VR nugget. *Functional coatings* can be added to the nuggets from the top right corner. The bottom menu provides *navigation* aspects within the virtual room during the authoring process. The *nugget menu* (right) provides the authors functionalities to replace parameters of the VR nuggets, move them, and edit pattern specific content (e.g., the callouts). VR Forge was evaluated using the abbreviated version of the AttrakDiff questionnaire [Hassenzahl et al. 2000].

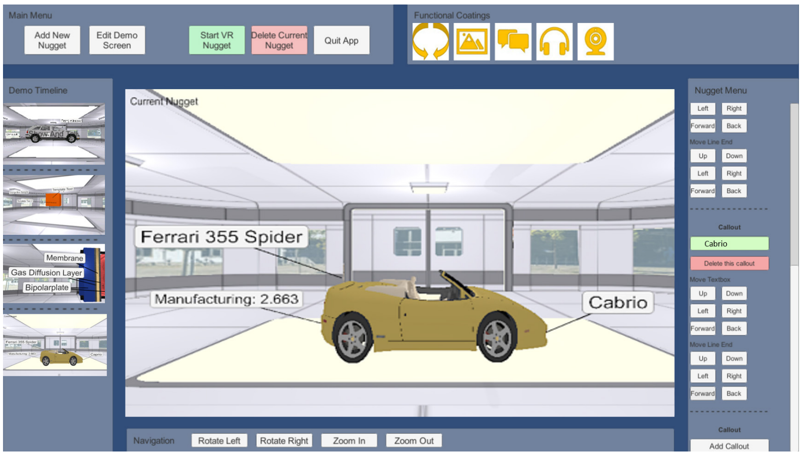

Figure 2: *The VR Forge authoring interface with an active show and tell nugget.*

3 IN-Tiles – A Pattern-Based Immersive Authoring Toolkit

In this section, we introduce our novel IN-Tiles toolkit, which uses immersive technologies to let authors experience their creations under the same circumstances as the end-users. The toolkit enables users to author VR nuggets at runtime within a virtual environment and therefore pursue a WYSIWYG approach for the authoring process. We discuss the underlying concepts of IN-Tiles. Specifically, we focus on our concepts to visualize VR nuggets within a virtual environment and to present affordances that allow manipulating VR nuggets.

The IN-Tiles system provides authors visual programming actions without code (visual scripting) within the immersive virtual environment. We represent a VR nugget as a 3D widget (Fig. 3 e) to make the abstract concept of VR nuggets graspable for authors. This widget is assembled with four tile-like shapes, illustrated in Fig. 3 a) to d). We provide authors these 3D tiles to customize a VR nugget. Fig. 3 a) shows the *pattern tile*. It represents the underlying educational pattern that the VR nugget is implementing. The authors can choose between provided patterns by selecting the specific pattern tile. In the following, we will stick with the example of the show and tell pattern presented in Section 2.

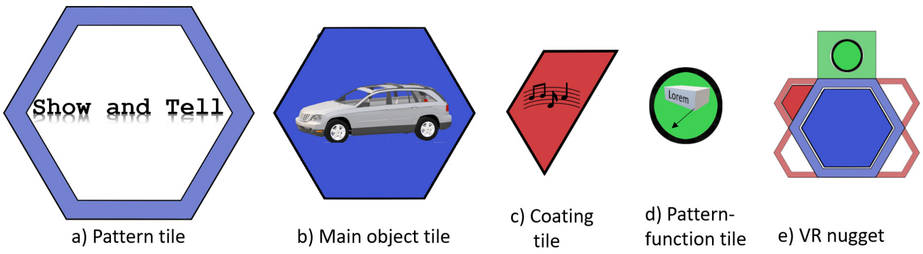

Figure 3: *a) to d): The tile-like shapes that we use to visualize the concept of VR nuggets within an immersive virtual environment. Authors use them to assemble a VR nugget structure (e).*

The second shape is the *main object tile* (Fig. 3 b). This tile represents 3D models that can be used in the VR nugget. It is inserted in the pattern tile. Authors can choose between 3D models that were loaded into the system previously in the authoring process. Depending on the number of main objects that a certain VR nugget requires, a pattern tile can consist of two or more shapes attached to each other on one of the hexagonal sides, shown in Fig. 3 a). Therefore, more than one main object tile can be inserted.

The *coating tile* (Fig. 3 c) can be added at one of the sides of the pattern tile to add supplemental functionalities for customizing VR nuggets. It can add optional functionalities, as described in Section 2. The *pattern-function tile* (Fig. 3 d) represents functionalities that are predefined in the VR nugget. Compared to the optional coating functionalities, these pattern-functions are mandatory. Therefore, such tiles are already attached to the VR nugget widget upon creating it. It is attached to the specific slot that appears at the top of the pattern tile. In the case of the show and tell VR nugget, this would be a callout. An example implementation of a show and tell VR nugget with one main object, one coating, and one callout is illustrated in Fig. 3 e).

The shape of the tiles follows a general design rationale inspired by naive physics. We utilize holes that offer the affordance to be filled. We use this affordance throughout the representations of VR nuggets and let authors assemble the general VR nugget system with them. The pattern tile, for example, represents solely the abstract pattern, but already gives an idea about the content of a VR nugget implementation (e.g., if it needs two main objects, it has two holes). At this stage, the actual content is still the default content that comes with the VR nugget. Holes of the pattern tile must be filled with tiles of the fitting form to customize the VR nugget to the authors' contentment.

Three virtual rooms reflect the IN-Tiles authoring process and structure it into three task areas. They are labeled according to their tasks as the *assembly room*, the *editing room*, and the *preview room*. Figure 4 illustrates the relation between each of the rooms. It is a 'has access to'-relation. There exists only one assembly room for a specific project to create a resulting learning application. One assembly room has access to n editing rooms, where n is the number of VR nuggets that are assembled by the author in this project. One editing room has access to only one preview room. In the following, we describe the connections between the three rooms in more detail.

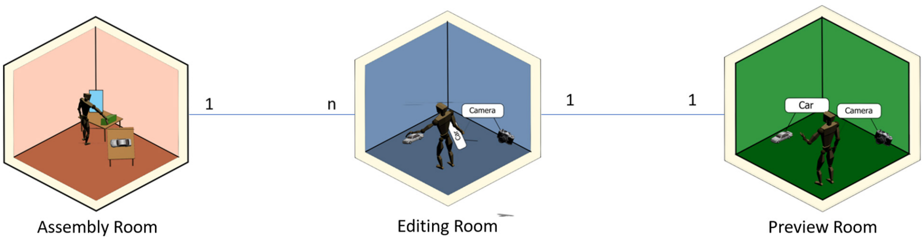

Figure 4: *The quantitative relationship between the three room entities – one assembly room has access to several editing rooms, while one editing room has access to exactly one corresponding preview room.*

In our authoring approach, the process starts when authors have loaded the 3D models that they want to use in the system and start using the immersive authoring toolkit. Basic authoring interactions such as the selection and assembly of VR nugget tiles are performed in the *assembly room*. We illustrate the process of creating a first VR nugget in Fig. 5. The author chooses a pattern for the VR nugget, selects the main object(s), adds additional functionality with the coatings, and adjusts the pattern-specific parameters. The tiles are exclusively used in this room. Here, authors make fundamental decisions about customizing a VR nugget and have the ability to outline the general structure of the resulting learning application. The authors can arrange the VR nugget widgets freely within the space of the assembly room.

We equip the authors with five *meta authoring tools* in the assembly room: 1) a duplicator, 2) a mover, 3) an eraser, 4) a connector and 5) a compiler tool. These tools are provided to facilitate the structuring and the handling of multiple VR nuggets. The duplicator allows authors to duplicate an existing VR nugget implementation and to create a reference from the original to the duplicate one. If the original VR nugget is changed, then the duplicate will adopt these changes as well. If the authors make changes to the duplicate, then the original will be unaffected. This tool is provided to create alternative implementations that can be used to adjust the learning content to an audience without having to create the whole VR nugget from scratch. An application example would be the use of another language for texts in show and tell callouts to adjust the learning application to another audience. The eraser tool lets authors erase existing VR nuggets permanently from the assembly room. The third tool allows authors to select and move one or more VR nugget structures in the assembly room. It provides authors the possibility to gain an overview of the implemented VR nuggets and sort or group them spatially in the room after their own order. This order does not affect the resulting learning application and is for supporting the authors in their workflow. However, to form a complete learning application at the end of the authoring approach, VR nuggets must also be brought in a sequential relation. Chronological order is necessary to bring the bite-sized VR applications in the right succession of execution. The connector ensures that authors can form more complex curricular structures from them, for example, a whole lesson built from nuggets. At last, the compiler can be used to finalize the authoring process and deploy the VR nuggets. When the compiler is used, a standalone application is exported that can be deployed in an operational VR learning environment. Our toolkit can create one application for each VR nugget separately, or if VR nuggets are connected to each other with the connection tool, then an application reflects such an association of VR nuggets.

When the authors have finished the assembly of the tiles, they enter the *editing room* in which they spatially adjust the 3D content of the individual nuggets. In the example of a show and tell VR nugget, they place the main object within an initially empty room, adjust the object's scale, add text to the callouts and place the callout labels in adjacent places to the 3D model. Then authors move existing connection lines to the correct position. The number of

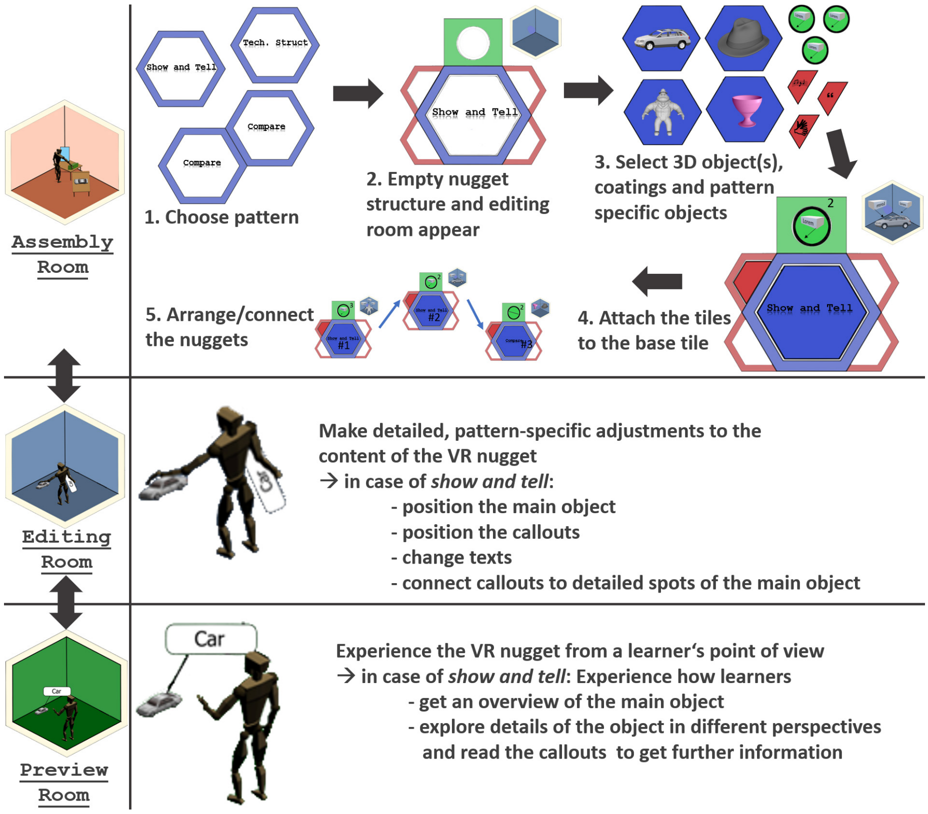

Figure 5: *The IN-Tiles authoring workflow. Authors are not limited to a linear workflow as illustrated and can jump to whatever step they need to.*

lines that exist is defined in the assembly room.

After fine-grained adjustments have been performed in the editing room, the author moves to the associated *preview room*. Inside the preview room, the authoring interface is hidden and only an additional exit switch is visible. Apart from that, both rooms look alike. Only VR interactions that were defined in the VR nugget implementation and that will be available for the subsequent learners are activated in this room. The authors can move back and forth between the preview room and the corresponding editing room. The state of the preview room (e.g., with an animation) is reset to the initial state. This assures that the authors can perform frequent and lightweight testing through the eyes of the learners, which is emphasized as an advantage of immersive authoring [Lee et al. 2004].

After all, the number of VR nuggets that are used by authors in one learning application has a bijective relation to the number of preview rooms. We use this bijection as a natural constraint to fulfill the nugget requirement that only one nugget can be active at a time – as a person can only be in one room at a time. The authors can also switch back to the assembly room from the other rooms as depicted in Fig. 5. Reasons for this may be to change crucial information at the current VR nugget or to put the current VR nugget assembly aside to start a new VR nugget from scratch.

4 Prototype

We implemented the IN-Tiles toolkit to show the feasibility of the concept. We based the prototype of our immersive visual scripting concepts on the Unity game engine and the Virtual Reality Tool-kit (VRTK) [Extend Reality Ltd 2020]. The VRTK was used to implement VR interactions. It supports several VR-related SDKs such as SteamVR or Oculus. For our prototype, we used an HTC Vive.

For the assembly room, we used a sci-fi environment where the proposed tools were placed as 3D widgets in the room (illustrated in Fig. 6). However, the sci-fi environment from the assembly room can be changed, similar to the skin of an application. Different environment skins can be provided, such as laboratory environments or school environments. These can either be included by the initial authors of IN-Tiles or defined as a parameter of the system so that laymen authors can input their own environment skin (e.g., as 360° image).

The skin of the assembly room can differ from the environment used in editing and preview rooms. Learners will perceive the environment of the editing/preview rooms in the deployed application, whereas the skin of the assembly room is only relevant during the authoring process. A preview of the editing and preview rooms is displayed in the assembly room as a world in miniature [Stoakley, Conway, and Pausch 1970]. It is updated during run-time to realize the WYSIWYG paradigm. The initial skin for editing and preview rooms is a common room environment with wooden floor tiles and concrete walls. This supports that authors can distinguish clearly between the futuristic assembly room skin and the traditional editing and preview room skin.

To allow the authors a glance on what content is in which implemented VR nugget, we chose to attach the miniatures of the editing room adjacent to each individual VR nugget structure, as shown in Fig. 7. This insight into the content of the editing room from the assembly room is relevant for authors to keep an overview of all implemented VR nuggets. It also enables them to select the right nuggets without transitioning into the editing room every single time they need to know the content of a specific nugget. While negligible in the case of a few nuggets, it becomes more crucial with a rising number of VR nugget structures positioned in the assembly room. We implemented the relation that one assembly

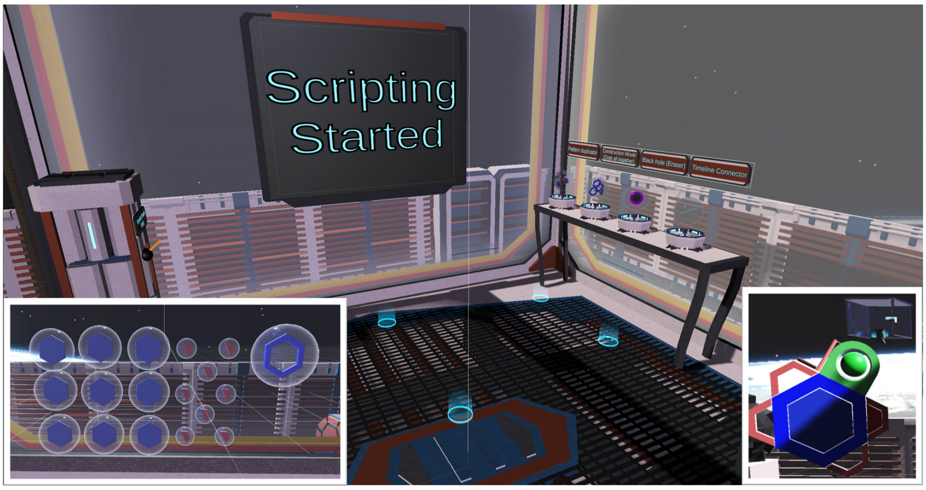

Figure 6: *Three renderings of the assembly room tools. Bottom left inlay: An illustration of different tile affordances. Bottom right inlay: An illustration of an implemented VR nugget tile. Main image: A part of the assembly room and the five additional meta authoring tools.*

room is connected to several editing rooms by making an editing room a child object to each individual pattern tile in Unity. Upon spawning such an initial pattern tile, a prefab of an empty editing room is instantiated as a child of it.

The transition from the assembly room to the editing room was implemented as a shrinking mechanism. Similar to Alice in Wonderland, the users will gradually shrink as they trigger the entry procedure. Authors can access the world in miniature model and experience it in full scale to make changes to the content. As shown in Fig. 7, the editing room consists of semi-transparent walls so that authors can still orient themselves in the larger-than-life assembly room. Therefore, authors are never exclusively in the editing room. However, using the switch on the wall, authors transition from the editing room to the preview room and back, where the assembly room is hidden, and the walls are opaque. From an implementation perspective, the preview room and the editing room remain the same room. The switch deactivates the authoring functions and hides all authoring-related tools to let authors experience the VR nugget from the learner's point of view. An open door in the editing room allows passing through to the assembly room again and activates the unshrinking mechanism.

The connection tool is placed in the assembly room. To use the proposed tools, authors grab them using their VR controllers. Fig. 8 shows its implementation. Three nuggets are connected in a series. By touching the VR nuggets in

Figure 7: *Left: The IN-Tiles toolkit with an adjacent miniature preview of the editing room. Right: A close-up view of the miniature.*

the room with the connection tool, numbers appear on them to indicate their sequence. Furthermore, we implemented a time constraint that determines how long a VR nugget will be active before it switches to the next VR nugget. This time constraint is an optional aspect. Authors thereby can restrict a single VR nugget's time of use in case authors plan a lesson with a tight schedule. To facilitate the selection tiles, we used hologram overlays to give insights on the content it provides. For example, a main object tile shows a hologram of the underlying 3D model. This allows authors to quickly distinguish and select the right tiles.

Concerning the deployment, our toolkit builds self-contained standalone applications using our compiler tool. Conceptually, one application is built from one or more learning nuggets. Technically, it links the preview rooms of the connected VR nuggets. As the learners do not need them, authoring functions (such as the editing- or assembly room tools) are not included in these applications. We discard Unity GameObjects and assets that are associated with authoring functionalities to optimize deployments (e.g., reduce the size of the application).

At last, educators can integrate our deployed VR nugget applications similar to learning nuggets that were realized with other media. For example, they can distribute the applications over a learning management system to learners (e.g., for home learning). They can also use the small and self-contained applications within attendance lessons to supplement their existing learning material or replace parts of it with VR-based material.

We presented the show and tell pattern as an example and to illustrate how patterns serve as a foundation in our concept. To provide authors with more VR nuggets in our system, new patterns must be identified first. For example, such patterns can be derived from interviews with experts from a specific domain. Deriving patterns from interviews or observation is usually a creative process. It

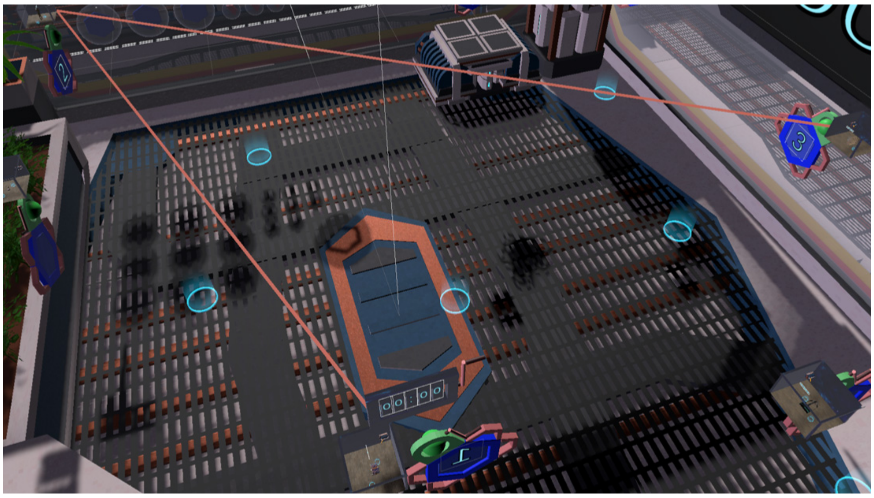

Figure 8: *Three VR nuggets are sequentially connected by red lines to illustrate the timeline of the bite-sized VR learning content within the assembly room.*

starts with the devising of hypotheses and confirming them in subsequent applications of the pattern – where one would analyze their re-usability or usefulness. After the identification of suitable patterns, they must be made available in the form of VR software and then parametrized and initially implemented with placeholder objects. Then, these new VR nuggets can be provided to authors within our authoring toolkit, similar as we provide the show and tell VR nugget.

5 Evaluation

We conducted a user study utilizing our prototype for evaluating our approach. Furthermore, we compare IN-Tiles to the VR Forge authoring system to contrast immersive and non-immersive authoring toolkits. To draw conclusions about the authoring concept and its usability for laymen, we considered four aspects:

[A1] **Ease of use**: Is the handling of the VR nugget tiles manageable for laymen?

[A2] **Workflow**: Do the authors know which interactions to perform at any time?

[A3] **Clarity**: Can the authors handle the creation of multiple VR nuggets in one assembly room or are there issues relating to the overview?

[A4] **Efficiency**: Are the authors satisfied with their results in relation to the time spent during authoring?

5.1 User Study

The user study involved 21 unpaid, voluntary participants (aged between 20 and 56 years with Ø 31.14 and SD 12.09, 6 females). They rated their experience on a 0-3 scale with Ø 1.24 and SD 1.02, where 0 means they never have used VR at all, and 3 means they use VR regularly. Therefore, we classify our participants as VR laymen authors. Our participants had mixed professional backgrounds. Nine participants had no profession of technical nature and our participants indicated that they have only a little experience with authoring technology. However, 15 expressed during the interview that they regularly create and conduct pedagogical presentations for their work.

The procedure of the study was as follows. Participants were welcomed and informed about the VR nugget concept in a brief oral presentation. They were introduced to the HTC Vive controllers and then guided towards four tasks they should accomplish. For each task, we built a separate and ready-built version of our prototype, where some content was already predefined. Therefore, the participants were introduced to the application step by step. The chronological sequence of the tasks is illustrated in Fig. 9.

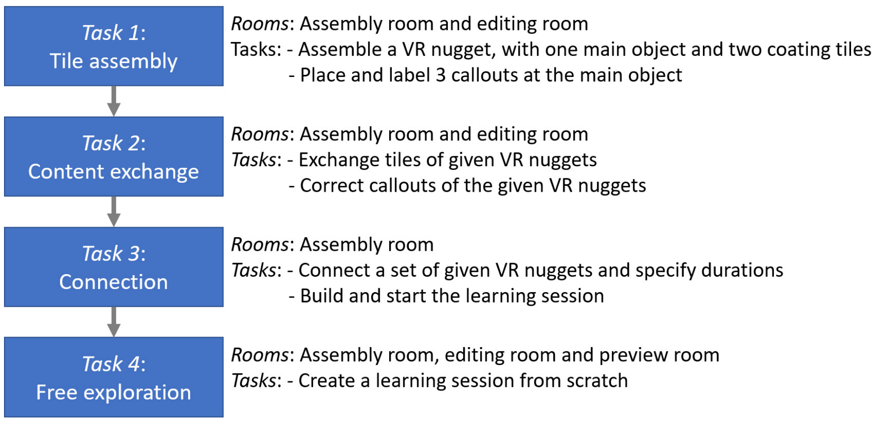

Figure 9: *Sequence of the user study tasks.*

(1) The first task asked the participants to implement one VR nugget with 3D objects of their choice and a given coating. In this scenario, we provided the participants five ready-made VR nuggets in the scene to give them an idea of what a final VR nugget could look like. (2) In the second task, they had to adjust aspects of existing nuggets. In a show and tell nugget, for example, they had to adjust callouts that annotate a human body so that the callouts would refer to the correct parts. (3) In task 3, the participants had to apply the

additional five meta-authoring tools to given VR nuggets during the third task. They should specifically focus on connecting and deploying the VR nuggets with distinct time-constraints. (4) In task four, the participants were asked to create a whole learning session using at least six VR nuggets with given 3D models. The scene for this task was the empty default scene of the authoring tool so that the participants had to create the VR nuggets from scratch.

After the study, the participants were asked to fill out a questionnaire that consisted of 15 custom questions, each relating to a certain evaluation aspect A1-A4. We also included the abbreviated version of the AttrakDiff questionnaire to draw general conclusions about the hedonic and pragmatic qualities [Hassenzahl et al. 2000] of the immersive authoring application.

In addition to our questionnaire, the thinking-aloud method [Lewis 1982] was used during all four tasks. We also provided free space on the questionnaire forms, for any comments that participants wanted to give. An experimenter took notes based on observations during the study, for example, when users made errors.

Our participants used a version of the questionnaire which was translated into their native language: Q1) How well were you able to distinguish the three different rooms/modes from each other? Q2) Was the functionality of the individual rooms clearly recognizable? Q3) How did the change between the three rooms affect you? Q4) How easy did the interaction with the tiles work? Q5) How well did the interaction with the tools (delete, move, connect, copy) work? Q6) How helpful were the tools (delete, move, connect, copy) for solving your tasks? Q7) How much did the shapes of the tiles help you to understand the structure of the VR nuggets? Q8) Did you feel you had enough space to create your VR nuggets? Q9) How easy did you find it to link the callouts to the desired points on the object? Q10) How well did you get an overview of the assembled construct (VR nugget) in the virtual world? Q11) How understandable was the workflow for the creation of the VR nuggets for you? Q12) How satisfied were you with the results of your VR learning application? Q13) In your judgment, took the creation of the VR application an acceptable amount of time? Q14) How well can you replicate the same results with different software? Q15) Would you have needed more time to create the learning application with other tools?

We used a semantic differential scale [Osgood, Suci, and Tannenbaum 1957] (7-point) to capture the data from the questionnaire items stated above. It is an established methodology for evaluation [Preece, Rogers, and Sharp 2015]. It was chosen for its ease of use for participants not familiar with an evaluation methodology and the flexibility to provide individual semantic word pairs for each item. A 7-point version was used to provide our participants enough gradations to express their feedback and to comply with the standardized AttrakDiff items that were included in our questionnaire and are based on a 7-point semantic differential scale, as well. Again, the items were translated into our participants' native language. The pairs for each item were the following: Indistinguishable–clear (Q1); not recognizable–clear (Q2); disturbing–pleasant (Q3); difficult–easy (Q4); difficult–easy (Q5); not helpful–helpful (Q6); not helpful–helpful (Q7); too little space–enough space (Q8); difficult–easy (Q9); it was confusing–it was clear (Q10); incomprehensible–understandable (Q11); dissatisfied–satisfied (Q12); too long–adequate (Q13); much better–not at all (Q14); quicker–longer (Q15). The study was performed within a time frame of one hour per participant.

5.2 Analysis of the Results

We carried out Wilcoxon signed-rank tests [Wilcoxon 1964] on the individual items Q1-Q15 to analyze how the IN-Tiles toolkit was rated by our participants. We ran a test for each item compared to the hypothetical neutral rating of 3 points. The outcome is used to make a statement about the significance of the difference between this neutral and our participants' ratings. Furthermore, we averaged the item ratings clustered by the aspect they are evaluating: Ease of use: Q5 and Q9; Workflow: Q3, Q4, Q6, Q7, and Q11; Clarity: Q1, Q2, Q8, and Q10; Efficiency: Q12, Q13, Q14, and Q15. On the mean aggregated values, we carried out further Wilcoxon signed-rank tests.

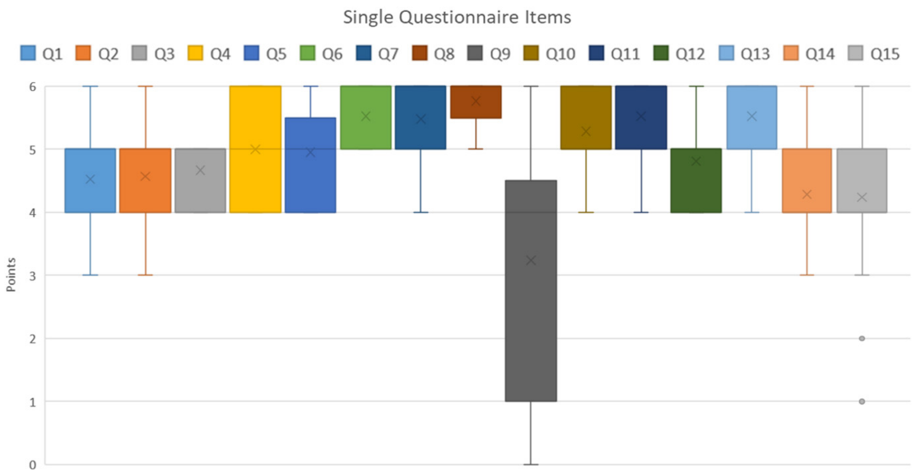

Figure 10: *Box-and-whisker plots for the single items Q1-Q15.*

The value distributions of the single items and the clustered aspects are depicted in the descriptive statistics in Fig. 10 and 11 respectively. They show that the medians of the box-whisker plots of all single questions lie above the value of 4 except for the question Q9, which shows a large span ranging from

0 to 6 points. The only outliers that the graphic shows concern Q15, where participants gave ratings of 2 and 1 points. In Fig. 11 we can see that again all median values lie above 4 points. The aspects 'clarity', 'workflow', and 'efficiency' show small deviation spans, whereas 'ease of use' shows clearly more variance ranging from 2 to approx. 5.5 points.

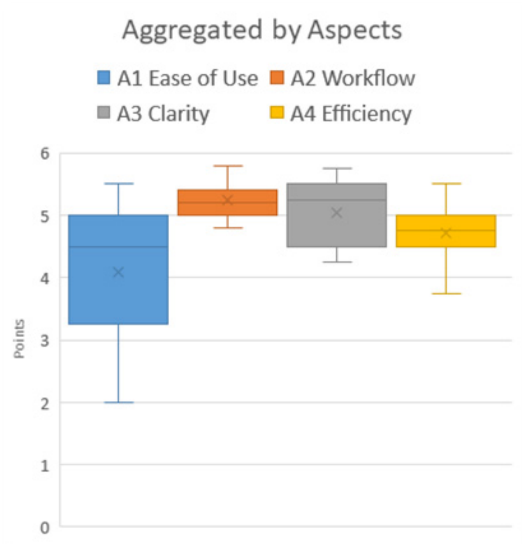

Figure 11: *Descriptive statistics of the item values aggregated by the four aspects A1-A4 using box-whisker plots.*

With a threshold for statistical significance of 5%, the Wilcoxon signed-rank tests show that for all single questions except Q9 a significant difference was detected. The respective p-values are as follow: Q1, Q2 and Q14: $p = 0.0002$; Q3, Q4, Q5, Q6, Q7, Q8, Q10, Q11, Q12 and Q13: $p \leq 0.00001$; Q15: $p = 0.00194$. No significance could be detected for Q9 with a p-value of $p = 0.64552$. Relating to the four aspects, the tests confirmed a significant difference compared to the neutral value with respective p-values: Ease of use: $p = 0.00194$; workflow, clarity and efficiency $p \leq 0.00001$.

We used the free-form comments from the questionnaire together with observations and statements from the thinking-aloud protocol to capture additional data about the editing process and to identify user errors and recoveries. We utilized affinity diagrams [Beyer and Holtzblatt 1998] as proposed by Preece, Rogers, and Sharp [Preece, Rogers, and Sharp 2015] to create overarching topics from comments and statements. We could assign these results again to the aspects *ease of use*, *clarity*, *workflow*, and *efficiency*.

Concerning the ease of use, the participants mentioned that the text input using VR hardware was perceived very tedious, whereas the detailed interaction with the 3D content within the editing room was judged positively by our participants. One participant specifically mentioned that he 'would not want to click with the mouse to do all the detailed interactions' such as positioning the callouts and adjusting the connection to the annotated 3D object (task 2). Concerning the interaction with the four types of tiles, one participant mentioned that 'it is really easy to recognize the right place for the tile' and the affordances were generally well accepted. We observed that multiple participants were uncertain about the correct sequence of putting together the tiles. In particular, this occurred when the participants wanted to start with a tile that was not the pattern tile. However, it is a prerequisite for using other tiles and for providing according affordances. Furthermore, we could observe impatience and fatigue when the participants had to type in the text for the callouts using the immersive technology (a virtual keyboard) or had to connect the nuggets to bring them in chronological order. To connect the nuggets, the participants had to walk back and forth between them within the assembly room.

The participants mentioned the clarity of the application and that they knew what they had to do or what they could do in the three specific rooms as soon as they identified in which room they were. They stated that they understood the distribution of the different authoring tasks to the rooms, but two participants indicated that they had difficulties to distinguish between the editing and the preview room. They suggested that more prominent visual cues, such as 'a large room label or pop-up on the walls' would help. The shrinking effect was positively perceived by our participants, as they could 'keep a view to other VR nuggets from within the editing room'. Eight participants explicitly stated that they had problems to keep an overview of connections between VR nuggets when numerous of them were connected. They could solve the issue by using the free 3D space of the assembly room to devise their own ordering systems. They placed the tile constructs that belonged together after their own assessment in similar regions of the assembly room. One participant specifically wished to exchange the sci-fi environment with a 'less exciting skin' such as 'an open meadow or a beach'. The participant mentioned that the sci-fi skin leads to distraction that was perceived as disruptive.

We observed that the general authoring workflow was understandable for our participants again as soon as they identified in which room they were. The non-linear workflow was judged mostly positive by our participants and was used frequently to switch between different rooms. Three participants liked that they could place the resulting VR nugget representations freely in the assembly room. One participant mentioned that she liked that VR nuggets could be structured after '[her] own arrangement system'.

As for the efficiency, our participants mentioned that they could successfully use IN-Tiles to create VR nuggets and one participant perceived the application as a 'fast and funny way to create VR content'. However, we could observe that our participants needed a short familiarization period before they were able to use our IN-Tiles toolkit to its full extent. Five participants indicated that picking up the three-dimensional tools in the assembly room took too much time in the process. One participant suggested placing the set of tools on a menu that is always attached to the authors' hand.

In addition to the above-mentioned questions and aspects, we analyzed the outcome of the AttrakDiff questionnaire items. The results are shown in Fig. 12. The figure also illustrates a comparison of our IN-Tiles toolkit with VR Forge, which was evaluated using the same AttrakDiff questionnaire by Horst and Dörner [Horst and Dörner 2019b]. One participant did not complete this part of the questionnaire and was not considered. The portfolio-presentation (Fig. 12 right) shows that IN-Tiles was rated similar to VR Forge by our participants. IN-Tiles is also located in the 'desired' region of the graph and even lies within the confidence area of VR Forge. The IN-Tiles square is slightly more shifted towards 'self-oriented' than towards 'task-oriented'. The word-pair visualization of the AttrakDiff items (Fig. 12 left) provides further insights into the single items. All points lie above one point on the positive side of the graph. Again, the graph shows similar results as for VR Forge. For most items, the ratings lie slightly below VR Forge. The values for the items 'ugly-attractive' and 'dull-captivating' are higher. The largest gaps between the two evaluations are located at the items 'confusing-clearly structured' and 'unpredictable-predictable'.

We obtained the evaluation data from Horst and Dörner [Horst and Dörner 2019b] and conducted a Mann-Whitney U test [Mann and Whitney 1947] with a threshold for statistical significance of 5% to draw conclusions on the significance of the differences between the AttrakDiff items of IN-Tiles and VR Forge. From 'complicated–simple' as the first item AI1 and 'bad–good' as the last item AI10 the Mann-Whitney U test did show the following p-values: AI1: $p = 0.4902$; AI2: $p = 0.21498$; AI3: $p = 0.24604$; AI4: $p = 0.07186$; AI5: $p = 0.93624$; AI6: $p = 0.24604$; AI7: $p = 0.15854$; AI8: $p = 0.36282$; AI9: $p = 0.69654$; AI10: $p = 0.9124$. None of the tests shows a significant difference.

5.3 Discussion of the Results

The results of our evaluation indicate that, with a short familiarization time, our laymen participants could successfully use IN-Tiles to create VR applications that serve as bite-sized learning nuggets. The statistical analysis supports these observations. Our participants also had fun during the use of IN-Tiles, however,

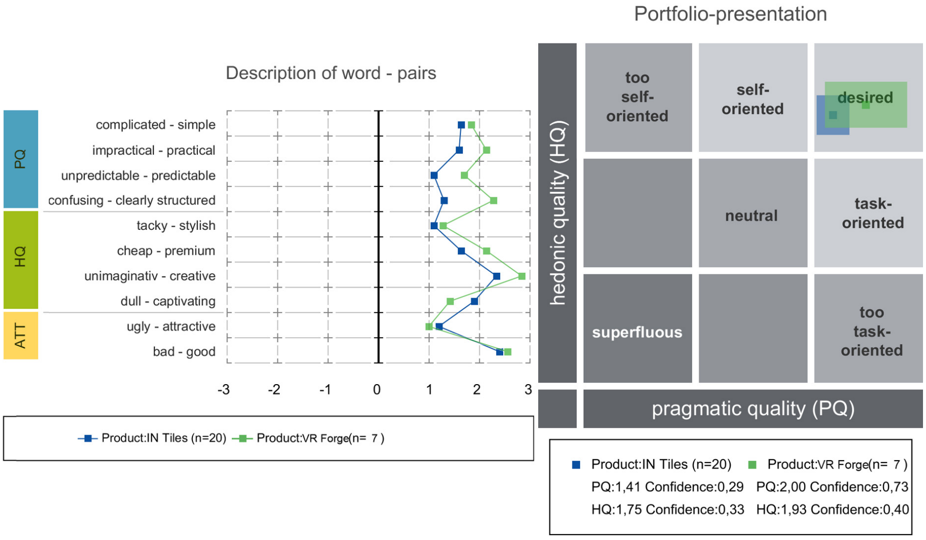

Figure 12: *Word-pair visualization (left) and portfolio-presentation of the AttrakDiff's outcome. It compares IN-Tiles (blue) with VR Forge (green).*

there can be an inherent bias of our participants to give positive feedback due to the novelty of the medium VR. This potential effect should be examined.

We could point out specific challenges and aspects for improvement. Particularly, these aspects relate to the ease of use. For example, the outlier of the test (Q9) indicates that the implementation of linking VR nuggets was challenging for some participants. This claim is further supported by the statements of the participants and is also reflected in the evaluation of the aspect clusters in Fig. 10. It shows a higher variance in the aspect ease of use than in the others. The second item for the ease of use aspect, Q5, was rated more positively. It is an indicator that the variation of the ease of use aspect may entirely be evoked by the challenge of linking VR nuggets.

The acceptance and usability of the prototype are also supported by the AttrakDiff analysis. The AttrakDiff graphs (Fig. 12) indicate that the results of the between-group comparison of IN-Tiles and VR Forge were very similar.

It is interesting that the participants of our study explicitly suggested that specific authoring actions of the application were not suitable to be performed in VR, such as entering text or connecting VR nuggets to more complex structures. Other functionalities, such as making detailed changes to the VR nugget in the editing room, were mentioned as particularly helpful. As some functionality is not implemented in VR Forge, there is no direct comparison between these.

However, it is an indicator that some authoring interactions for creating bite-size VR applications may be better suited for immersive and others may be better suited for non-immersive technologies. This suggests the exploration of combinations of traditional and immersive authoring methodologies.

6 Conclusions and Future Work

In this paper, we introduced the IN-Tiles toolkit. IN-Tiles builds on microlearning methodologies and enables authors to create bite-sized VR applications for learning purposes. We work with implementations of solutions to recurring problems. Once implemented by a programmer and included in IN-Tiles, a VR nugget can be customized to be used in various similar use-cases. In this work, we focused on patterns that lie in the field of education and microlearning, but the general authoring paradigm we utilize could also be transferred to other domains. Some nuggets may even be agnostic to the application domain.

We demonstrated that laymen authors could successfully create VR nuggets in the user test we conducted. We compared our toolkit to the VR Forge [Horst and Dörner 2019b] using the AttrakDiff questionnaire. IN-Tiles performed similarly well as VR Forge. We could show that there were certain points in the VR nugget authoring workflow that were well suited to be performed in our immersive authoring environment. For example, making spatial-related changes to the VR nugget, such as placing the callouts. We identified tasks in our evaluation that could be carried out better on a conventional 2D application. Examples are entering text but also connecting VR nuggets to more complex structures.

The future of research in pattern-based authoring can be divided into two categories. First, we have technically laid the foundation to further explore the fields that we refer to as 'mixed' or 'asymmetric immersive authoring'. We presume that it may be beneficial to use both desktop and immersive technologies within the VR nugget authoring workflow. Being able to switch back and forth between IN-Tiles and conventional authoring applications such as VR Forge could have been useful for our participants during our study. Second, there is still the challenge that a VR expert must create and provide an initial implementation for each VR nugget before less-experienced authors can make use of them. Establishing an exchange format and standardizing the concept of VR nuggets is a next step that can help to make it accessible for a greater community. For example, this could be done comparable to the .unitypackage format for exchanging Unity project files. With such an exchange format, a platform for VR nuggets could be built. For example, authors could easily transfer VR nuggets between IN-Tiles and VR Forge. It would also enable authors to access VR nuggets from 'stores'. VR nugget stores could be inspired by the 'asset store' for Unity plugins or the app store for mobile apps. These stores also support the distribution of applications/plug-ins free of charge. Such a platform would enhance the reusability of VR content to a great extent.

Acknowledgments

The work is supported by the Federal Ministry of Education and Research of Germany in the project Innovative Hochschule (funding number: 03IHS071).

References

[Bailenson et al. 2008] Bailenson, J. N., Yee, N., Blascovich, J., Beall, A. C., Lundblad, N. and Jin, M. (2008). "The use of immersive virtual reality in the learning sciences: Digital transformations of teachers, students, and social context". The Journal of the Learning Sciences, 17(1), 102-141.

[Beaudin et al. 2007] Beaudin, J. S., Intille, S. S., Tapia, E. M., Rockinson, R. and Morris, M. E. (2007, November). "Context-sensitive microlearning of foreign language vocabulary on a mobile device". In European conference on Ambient intelligence (pp. 55-72). Springer, Berlin, Heidelberg.

[Beutner and Pechuel 2017] Beutner, M. and Pechuel, R. (2017, March). "Micro Units-A New Approach to Making Learning Truly Mobile". In Society for Information Technology & Teacher Education International Conference (pp. 744-751). Association for the Advancement of Computing in Education (AACE).

[Beyer and Holtzblatt 1998] Beyer, H. and Holtzblatt, K. (1998). "Contextual design: defining customer-centered systems (Vol. 1)". Morgan Kaufmann.

[Bierbaum et al. 2001] Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A. and Cruz-Neira, C. (2001, March). "VR Juggler: A virtual platform for virtual reality application development". In IEEE Virtual Reality 2001 (pp. 89-96).

[Davies, Crohn, and Treadgold 2019] Davies, A. G., Crohn, N. J. and Treadgold, L. A. (2019). "Can virtual reality really be used within the lecture theatre?". BMJ Simulation and Technology Enhanced Learning, 5(4), 234-235.

[Dunk, Haffegee, and Alexandrov 2010] Dunk, A., Haffegee, A. and Alexandrov, V. (2010). "Selection methods for interactive creation and management of objects in 3D immersive environments". Procedia Computer Science, 1(1), 2609-2617.

[Dörner et al. 2003] Dörner, R., Geiger, C., Haller, M. and Paelke, V. (2003). "Authoring mixed reality - a component and framework-based approach". In Entertainment Computing (pp. 405-413). Springer, Boston, MA.

[Dörner, Kallmann, and Huang 2015] Dörner, R., Kallmann, M. and Huang, Y. (2015). "Content creation and authoring challenges for virtual environments: from user interfaces to autonomous virtual characters". In Virtual Realities (pp. 187-212). Springer.

[Extend Reality Ltd 2020] Extend Reality Ltd. (2020). Virtual Reality Toolkit. Accessed 12.01.2020. https://github.com/ExtendRealityLtd/VRTK

[Figueroa et al. 2008] Figueroa, P., Bischof, W. F., Boulanger, P., Hoover, H. J. and Taylor, R. (2008). "Intml: A dataflow oriented development system for virtual reality applications". Presence: Teleoperators and Virtual Environments, 17(5), 492-511.

[Figueroa et al. 2012] Figueroa, P., Gil, S., Oses, R., Toro, J., Rodriguez, C., Benavides, C. and Correa, E. (2012). "Visual Programming for Virtual Reality Applications Based on InTml". SBC Journal on 3D Interactive Systems, 3(1).

[Hassenzahl et al. 2000] Hassenzahl, M., Platz, A., Burmester, M. and Lehner, K. (2000, April). "Hedonic and ergonomic quality aspects determine a software's appeal". In Proceedings of the SIGCHI conference on Human Factors in Computing Systems (pp. 201-208). ACM.

[Horst and Dörner 2019a] Horst, R. and Dörner, R. (2019). "Integration of Bite-Sized Virtual Reality Applications into Pattern-Based Knowledge Demonstration". In Proceedings of the 16th Workshop Virtual and Augmented Reality of the GI Group VR/AR. Gesellschaft für Informatik. Shaker-Verlag. (pp. 137-148).

[Horst and Dörner 2019b] Horst, R. and Dörner, R. (2019). "Virtual Reality Forge: Pattern-Oriented Authoring of Virtual Reality Nuggets". In 25th ACM Symposium on Virtual Reality Software and Technology. ACM. New York. 1–12.

[Horst and Dörner 2019c] Horst, R. and Dörner, R. (2019). "Mining Virtual Reality Nuggets: A Pattern-Based Approach for Creating Virtual Reality Content Based on Microlearning Methodology". Proceedings of the Intl. Conference on Engineering, Technology and Education (IEEE TALE 2019), IEEE Press. to appear (8 pages).

[Hug 2005] Hug, T. (2005). "Micro learning and narration: exploring possibilities of utilization of narrations and storytelling for the design of "micro units" and didactical micro-learning arrangements". Proceedings of Media in Transition.

[Jee et al. 2014] Jee, H. K., Lim, S., Youn, J. and Lee, J. (2014). "An augmented reality-based authoring tool for E-learning applications". Multimedia Tools and Applications, 68(2), 225-235.

[Klinker et al. 2002] Klinker, G., Dutoit, A. H., Bauer, M., Bayer, J., Novak, V. and Matzke, D. (2002, October). "Fata morgana-a presentation system for product design". International Symposium on Mixed and Augmented Reality (76-85). IEEE.

[Lee et al. 2004] Lee, G. A., Nelles, C., Billinghurst, M. and Kim, G. J. (2004, November). "Immersive authoring of tangible augmented reality applications". In Third International Symposium on Mixed and Augmented Reality (pp. 172-181). IEEE.

[Lewis 1982] Lewis, C. (1982). "Using the 'thinking-aloud' method in cognitive interface design". Yorktown Heights, NY: IBM TJ Watson Research Center.

[Mann and Whitney 1947] Mann, H. B. and Whitney, D. R. (1947). "On a test of whether one of two random variables is stochastically larger than the other". The annals of mathematical statistics, 50-60.

[Monahan, McArdle, and Bertolotto 2008] Monahan, T., McArdle, G. and Bertolotto, M. (2008). "Virtual reality for collaborative e-learning". Computers & Education, 50(4), 1339-1353.

[Osgood, Suci, and Tannenbaum 1957] Osgood, C. E., Suci, G. J., & Tannenbaum, P. H. (1957). The measurement of meaning (No. 47). University of Illinois press.

[Polsani 2003] Polsani, P. R. (2003). "Use and abuse of reusable learning objects". Journal of Digital information, 3(4).

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

[Schmalstieg et al. 2002] Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavári, Z., Encarnaçao, L. M., Gervautz, M. and Purgathofer, W. (2002). "The studierstube augmented reality project". Presence: Teleoperators & Virtual Environments, 33-54.

[Souza and do Amaral 2014] Souza, M. I. F. and do Amaral, S. F. (2014). "Educational microcontent for mobile learning virtual environments". Creative Education 5(9).

[Steed 2008] Steed, A. (2008). "Some useful abstractions for re-usable virtual environment platforms". IEEE VR SEARIS Workshop.

[Stoakley, Conway, and Pausch 1970] Stoakley, R., Conway, M. J. and Pausch, R. (1995, May). "Virtual reality on a WIM: interactive worlds in miniature". In CHI (Vol. 95, pp. 265-272).

[Takala 2014] Takala, T. M. (2014, October). "RUIS: A toolkit for developing virtual reality applications with spatial interaction". In Proceedings of the 2nd ACM symposium on Spatial user interaction (pp. 94-103). ACM.

[Wilcoxon 1964] Wilcoxon, F. and Wilcox, R. A. (1964). "Some rapid approximate statistical procedures". Lederle Laboratories.

[Zhang et al. 2004] Zhang, D., Zhao, J. L., Zhou, L. and Nunamaker Jr, J. F. (2004). "Can e-learning replace classroom learning?". Communications of the ACM, 75-79.