Hochschule RheinMain

Dissertation

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften

am Promotionszentrum Angewandte Informatik
der hessischen Hochschulen für Angewandte Wissenschaften:
Hochschule RheinMain, Hochschule Darmstadt,
Hochschule Fulda und Frankfurt University of Applied Sciences

AR Nuggets: Pattern-Based Components for Using and Authoring Augmented Reality Applications

vorgelegt von	**Linda RAU**, M.Sc.
Betreuer:	Prof. Dr. Ralf DÖRNER, Hochschule RheinMain Prof. Dr. Klaus BÖHM, Hochschule Mainz
Gutachter:	Prof. Dr. Paul GRIMM, Hochschule Darmstadt Prof. Dr. Wolfgang BROLL, Technische Universität Ilmenau
Einreichungstermin:	13.12.2023
Prüfungstermin:	19.04.2024
Erscheinungsort:	Wiesbaden
Erscheinungsjahr:	2024

Abstract

Augmented Reality (AR) can be applied to various applications to enhance physical environments or objects virtually. Although there is a growing interest in AR applications, challenges in authoring and using AR remain. Creating AR applications typically requires programming and domain-specific knowledge and thus is a time- and resource-intensive task. Similarly, using AR applications involves challenges, e.g., unfamiliar interactions.

This thesis explores how pattern-based software components can target such challenges and introduces *AR nuggets*. AR nuggets draw from the educational concept of microlearning, where a learning unit consists of multiple elementary self-contained learning nuggets. This thesis transfers the idea of self-contained nuggets to AR in the form of stand-alone and self-contained AR applications. We propose to base these AR applications on patterns and to provide interactive, ready-to-use applications. For example, an application based on a quiz pattern can provide all required scripts, interactions, and virtual 3D objects. Together with domain experts, we identified eight patterns with different variations. We call our ready-to-use, pattern-based, stand-alone, and self-contained AR applications AR nuggets.

AR nuggets target to support users by utilizing tangible interactions and providing proactive user assistance. Furthermore, this thesis develops concepts to combine multiple AR nuggets into one larger AR experience or a mixed experience that includes Virtual Reality (VR) nuggets. AR nuggets also support location-specific experiences, where they can guide users from one AR nugget to another. In five user studies, we show that AR nuggets can add value to continuing medical education as well as contribute to complex location-specific and non-linear AR experiences and experiences with VR.

We introduce three approaches for authoring with AR nuggets that we also implement and evaluate. In three further user studies, we show that persons without programming knowledge or experience with AR can use AR nuggets to create their own AR applications.

Finally, this thesis describes lessons learned from the user studies that can serve as guidelines for future researchers. It contributes to lowering barriers to authoring AR by supporting domain experts with little or no programming and AR experience in creating individual and interactive AR experiences on their own. Overall, this contributes to a) opening up using AR for a larger audience, b) reducing barriers to using and authoring AR, and c) exploiting the advantages of AR in many use cases.

Keywords:

Augmented Reality; Learning Nuggets; Patterns; Tangible Augmented Reality; Authoring Tools; Location-Specific; Human-Computer Interaction

Kurzzusammenfassung

Augmented Reality (AR) kann in verschiedensten Anwendungen physische Umgebungen oder Objekte virtuell erweitern. Obwohl das Interesse an AR Anwendungen wächst, gibt es nach wie vor Herausforderungen bei der Erstellung und Nutzung von AR. Die Erstellung von AR Anwendungen erfordert in der Regel Programmier- und Fachkenntnisse und ist daher eine zeit- und ressourcenintensive Aufgabe. Auch die Nutzung von AR Anwendungen ist mit Herausforderungen verbunden, z. B. mit ungewohnten Interaktionen.

Diese Arbeit untersucht, wie musterbasierte Softwarekomponenten solche Herausforderungen adressieren können und stellt dazu *AR nuggets* vor. AR nuggets basieren auf dem pädagogischen Konzept des Mikrolernens, bei dem eine Lerneinheit aus mehreren elementaren, in sich abgeschlossenen Lernnuggets besteht. Wir übertragen dies in Form von eigenständigen und in sich abgeschlossenen AR Anwendungen auf AR und stellen interaktive, gebrauchsfertige AR Anwendungen auf der Basis von Anwendungsmustern zur Verfügung. Zum Beispiel kann eine auf einem Quiz-Muster basierende Anwendung alle erforderlichen Skripte, Interaktionen und virtuellen 3D-Objekte bereitstellen. Mit Domänenexperten haben wir acht Muster und verschiedene Variationen identifiziert. Diese gebrauchsfertigen, musterbasierten, eigenständigen und in sich geschlossenen AR Anwendungen nennen wir AR nuggets.

AR nuggets bieten auch Interaktionen mit Tangibles (greifbaren Objekten) und proaktive Unterstützung für Personen, die AR nuggets nutzen, an. Darüber hinaus entwickeln wir Konzepte, die mehrere AR nuggets zu einem größeren AR Erlebnis oder einem gemischten AR und Virtual Reality (VR) Erlebnis mit VR nuggets kombinieren können. AR nuggets unterstützen auch ortsspezifische Erlebnisse, bei denen sie Personen von einem AR nugget zum anderen führen können. In fünf Nutzerstudien zeigen wir, dass AR nuggets einen Mehrwert für die medizinische Fortbildung bieten sowie zu komplexen ortsspezifischen, nicht-linearen AR-Erlebnissen und zu Erfahrungen mit AR sowie VR beitragen können.

Wir präsentieren, implementieren und evaluieren drei Vorgehensweisen für das Erstellen eigener AR Anwendungen mit AR nuggets. In drei weiteren Nutzerstudien zeigen wir, dass dies auch für Personen ohne Programmierkenntnisse oder Erfahrung mit AR möglich ist.

Abschließend beschreibt diese Arbeit die aus den Nutzerstudien gewonnenen Erkenntnisse, die als Leitfaden für zukünftige Forschende dienen können. Sie trägt dazu bei, die Hürden für die Erstellung von AR-Anwendungen zu senken, indem sie Domänenexperten mit wenig oder keinen Programmierkenntnissen und Erfahrung mit AR dabei unterstützt, individuelle und interaktive AR Erlebnisse selbst zu erstellen. Insgesamt trägt dies dazu bei, a) die Nutzung von AR für ein größeres Publikum zu öffnen, b) die Barrieren für die Nutzung und Erstellung von AR zu verringern und c) die Vorteile von AR in vielen Anwendungsfällen zu nutzen.

Schlagwörter:

Erweiterte Realität/Augmented Reality; Learning Nuggets; Patterns; Tangible Augmented Reality; Autorenwerkzeuge; Location-Specific; Mensch-Computer Interaktion

Erklärung zur Autorenschaft

Ich, Linda Rau, erkläre hiermit, dass

- die Dissertation selbständig und ohne unerlaubte fremde Hilfe und nur mit den angegebenen Hilfen angefertigt wurde;
- alle wörtlich oder sinngemäß aus veröffentlichten Schriften entnommene Text- stellen und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht sind;
- die Grundsätze guter wissenschaftlicher Praxis eingehalten sind.

Teile der vorliegenden Thesis sind Teil der Arbeiten, die in den folgenden Artikeln und Tagungsbändern vorgestellt und veröffentlicht wurden:

Rau, Linda; Horst, Robin; Liu, Yu; Dörner, Ralf; Spierling, Ulrike: A Tangible Object for General Purposes in Mobile Augmented Reality Applications. In: Reussner, R. H., Koziolek, A. & Heinrich, R. (Hg.): INFORMATIK 2020. Bonn, S. 947–954.

Rau, Linda; Horst, Robin; Liu, Yu; Dorner, Ralf (2021): A Nugget-Based Concept for Creating Augmented Reality. In: 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). Bari, Italy, 10/4/2021 - 10/8/2021: IEEE, S. 212–217.

Rau, Linda; Döring, Dagny C.; Horst, Robin; Dörner, Ralf (2022): Pattern-Based Augmented Reality Authoring Using Different Degrees of Immersion: A Learning Nugget Approach. In: Front. virtual real. 3, Artikel 841066. DOI: 10.3389/frvir.2022.841066.

Bitter, Jessica L.; Dörner, Ralf; Liu, Yu; Rau, Linda; Spierling, Ulrike (2022): Follow the Blue Butterfly – An Immersive Augmented Reality Museum Guide. In: Constantine Stephanidis, Margherita Antona und Stavroula Ntoa (Hg.): HCI International 2022 Posters. 24th International Conference on Human-Computer Interaction, HCII 2022, Virtual Event, June 26 – July 1, 2022, Proceedings, Part III, Bd. 1582. 1st ed. 2022. Cham: Springer International Publishing; Imprint Springer (Springer eBook Collection, 1582), S. 171–178.

Rau, Linda; Bitter, Jessica L.; Liu, Yu; Spierling, Ulrike; Dörner, Ralf (2022): Supporting the creation of non-linear everyday AR experiences in exhibitions and museums: An authoring process based on self-contained building blocks. In: Front. virtual real. 3, Artikel 955437. DOI: 10.3389/frvir.2022.955437.

(Für Publikation angenommen): Rau, Linda; Horst, Robin; Feller, Manuel; Dörner, Ralf (2023): Bridging Realities: Bidirectional Transitions from and to Augmented and Virtual Reality. GI VR / AR Workshop 2023. Gesellschaft für Informatik e.V.. Köln. 19. - 20. September 2023

(Für Publikation angenommen): Rau, Linda; Döring, Dagny C.; Horst, Robin; Dörner, Ralf (2023): A Perspective on Interface Techniques in Tangible Augmented Reality for Mobile Devices. Journal of Virtual Reality and Broadcasting

Eine vollständige Liste meiner Veröffentlichungen ist in Appendix B aufgeführt.

Datum Unterschrift

Acknowledgements

Here, I want to acknowledge the people who supported, inspired, and accompanied me through the journey of working on this thesis. First and foremost, I am thankful to my advisor, Ralf Dörner, who motivated and inspired me. Thank you, Ralf, for genuinely sharing your experiences, always constructive feedback, and patience and support in all situations, whether academic or work-related. I want to thank my second advisor, Klaus Böhm, who brought the idea of working on this thesis to me. Thank you, Klaus, for encouraging me, sharing contacts and insights, and providing valuable feedback on my research. This thesis would not have been possible without the support of my two great advisors.

I also owe a great deal to my colleagues for their camaraderie, fruitful discussions, and helpful feedback. Especially, I want to thank Robin Horst. Robin, you inspired me academically and personally, and I am so grateful that I had the pleasure of working with you. Thank you for sharing your ideas and insights and for your encouragement. I also extend my thanks to my colleagues Jessica L. Bitter, Dennis Fenchel, Yu Liu, Wilhelm Retz, Reimond Retz, and Ulrike Spierling for their collaborative efforts and support. Thank you all for accompanying me in my research and constructive work together.

My work was supported by the German Federal Ministry of Education and Research (BMBF), funding program Forschung an Fachhochschulen, contract number 13FH181PX8, and LOEWE-Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz, Förderlinie 3: KMU-Verbundvorhaben (HA project no. 690/19-10).

I also had the pleasure of supervising and working with students who helped me realize some of my ideas. Thank you, Fabian Birk, Dagny C. Döring, and Manuel Feller, for your dedication, reliability, new ideas, and work. Special thanks go to all the persons who supported or encouraged me during the last years: Thank you, Bastian Plaß, Alexander Rolwes, Isabella Buck, and Kira Kastell.

Finally, I want to thank my dear friends and family. Thank you for your unwavering encouragement, compassion, and understanding, which have been a source of strength throughout this journey. Thank you to my parents, who always pushed me to new achievements without exerting pressure. My deepest appreciation goes to my husband, Tim. Thank you for being there for me, even when the journey was challenging.

Contents

List of Abbreviations xiii

List of Figures xiv

List of Tables xvi

1 Introduction **17**
- 1.1 Motivation . . . 17
- 1.2 Research Questions . . . 19
- 1.3 Methodology . . . 20
- 1.4 Contributions . . . 22
- 1.5 Thesis Structure . . . 23

2 Related Work **25**
- 2.1 Augmented Reality . . . 25
- 2.2 Applied Patterns . . . 26
- 2.3 Connection to the Real World . . . 27
  - 2.3.1 Tangible Augmented Reality . . . 27
  - 2.3.2 Location-Specific AR Applications . . . 29
- 2.4 User Assistance . . . 30
- 2.5 Transitioning to and from Augmented Reality . . . 32
- 2.6 Creating Augmented Reality Applications . . . 34
  - 2.6.1 Challenges in the Authoring Process . . . 34
  - 2.6.2 Authoring Tools . . . 36
- 2.7 Discussion . . . 44

3 Concepts for AR Nuggets **47**
- 3.1 Definition . . . 47
  - 3.1.1 AR Nuggets . . . 47
  - 3.1.2 Involved Actors for Creating and Using AR Nuggets . . . 48
- 3.2 Examples . . . 49
- 3.3 Utilization of Tangible Interactions in AR Nuggets . . . 59
  - 3.3.1 Realistically Shaped Tangible . . . 60
  - 3.3.2 Generic Tangible . . . 62
  - 3.3.3 Plug System Combination of Realistically Shaped and Generic Tangible . . 63

3.4 Integration of User Assistance in AR Nuggets 64
3.5 Usage of Multiple AR Nuggets in Complex AR Setups 67
3.6 Combination of AR Nuggets and VR Nuggets 72

4 Implementation of AR Nuggets 75
4.1 AR Nuggets 75
4.2 Utilization of Tangible Interactions 79
4.3 Integration of User Assistance in AR Nuggets 81
4.4 Usage of Multiple AR Nuggets in Complex AR Setups 83
4.5 Combination of AR Nuggets and VR Nuggets 86

5 Authoring With AR Nuggets 89
5.1 Using Different Degrees of Immersion 89
  5.1.1 Concept 89
  5.1.2 Implementation 91
5.2 Integrated AR Nugget Authoring Tools 96
  5.2.1 Concept 96
  5.2.2 Implementation 102
5.3 Constraint-based Authoring with AR Nuggets 105
  5.3.1 Concept 105
  5.3.2 Implementation 107
5.4 Content Delivery 109

6 Evaluation 111
6.1 AR Nuggets 111
  6.1.1 Expert User Study 111
  6.1.2 Expert User Study Analysis 112
  6.1.3 Expert User Study Discussion 114
6.2 Utilization of Tangible Interactions 115
  6.2.1 User Study 115
  6.2.2 User Study Analysis 117
  6.2.3 User Study Discussion 120
6.3 Integration of User Assistance in AR Nuggets 121
  6.3.1 User Study 122
  6.3.2 User Study Analysis 123
  6.3.3 User Study Discussion 124
6.4 Usage of Multiple AR Nuggets in Complex AR Setups 124
6.5 Combination of AR Nuggets and VR Nuggets 127
  6.5.1 User Study 127
  6.5.2 User Study Results 128
6.6 AR Nugget Authoring Using Different Degrees of Immersion 130
  6.6.1 User Study 130

6.6.2 User Study Analysis 132
6.6.3 User Study Discussion 135
6.7 Integrated AR Nugget Authoring Tools 136
6.7.1 Authoring Workshop 136
6.7.2 User Study Analysis 138
6.7.3 User Study Discussion 142
6.8 Constraint-based Authoring with AR Nuggets 143
6.8.1 User Study 143
6.8.2 User Study Analysis 144
6.8.3 User Study Discussion 145

7 Discussion **147**

8 Conclusion and Future Work **154**

A References **157**

B List of Own Publications **176**

C Awards **179**

D Authoring Workshop Manual **180**

E Research Data **188**

List of Abbreviations

AR	Augmented Reality
CME	Continuing Medical Education
HHD	Handheld Device
HMD	Head-Mounted Device
MRTK	Mixed Reality Toolkit
OMG	Object Management Group
PoI	Point of Interest
RFISR	Research Framework for Information System Research
RQ	Research Question
SLAM	Simultaneous Localization and Mapping
SPES	Spatial Presence Experience Scale
UWP	Universal Windows Platform
VR	Virtual Reality

List of Figures

3.1 Example of a default and an adapted AR nugget 49
3.2 Use case diagram of involved actors for creating and using AR nuggets. 50
3.3 Two 3D-printed realistically shaped tangibles 61
3.4 Images of our generic tangible 64
3.5 A combined tangible of realistically shaped and generic tangibles 64
3.6 Activity diagram of the virtual assistant 66
3.7 Activity diagram of the AR nugget manager and AR nuggets 68
3.8 State diagram of the agent's guiding state machine 70
3.9 Leverage effect on virtual elements that are not placed close to their real-world anchor 71
3.10 A bridge connecting an AR nugget with a VR nugget 73

4.1 Process of importing, exporting, and updating virtual objects' positions and their spatial anchors 76
4.2 Screenshots of a Continuing Medical Education (CME) course about actinic keratosis 78
4.3 Screenshot of a CME course about endometriosis 79
4.4 AR nuggets in the Senckenberg Museum 80
4.5 Prototype with different types of tangibles 81
4.6 Screenshot of the virtual assistant's hints 82
4.7 Screenshot of an AR nugget taken through the HoloLens 2 84
4.8 Pathway calculations based on a 3D scan that has been pre-processed 85
4.9 The three AR experiences 86
4.10 Screenshots of five transitions from/to AR/VR 87

5.1 Conceptual AR nugget authoring workflow. 90
5.2 Screenshot of the AR Nugget authoring tool ARNAUDDI on the desktop device . . 92
5.3 Screenshot of the AR Nugget authoring tool ARNAUDDI on the desktop device . . 93
5.4 An adapted show & tell AR nugget 94
5.5 An adapted semantic zoom AR nugget 95
5.6 An adapted quiz AR nugget 96
5.7 Flowchart of the twofold authoring process 97
5.8 Overview of integrated AR nugget authoring tools. 98
5.9 Influence of the AR device's position and rotation on application start. 99
5.10 Placing a location-specific virtual object with the *grabbable* tool. 103

5.11 Screenshot of the *mode switcher* tool in the Game Engine Unity. 104
5.12 Screenshot of the *rotate towards* tool in the Game Engine Unity. 105
5.13 Screenshot of the *condition manager* tool in the Game Engine Unity. 106
5.14 Automatic calculation of surface and distance constraints based on rankings. . . . 108
5.15 Toolchain for on-demand AR content delivery to a ready-built application. 109

6.1 Results for Q1 - Q6 and Q12. 113
6.2 AR nuggets that participants in the user study explored. 116
6.3 Procedure of the user study 117
6.4 Results for Q1, Q2, and Q4 from our questionnaire 119
6.5 Results from the UEQ-S for each tangible type. 119
6.6 The media designers’ draft for the museum visitor journey 125
6.7 Design of the user study for AR and VR nugget combinations. 128
6.8 Results from the AttrakDiff and Spatial Presence Experience Scale (SPES) 129
6.9 Outcomes from the AttrakDiff questionnaire 133

List of Tables

1.1 Research phases classified based on the Research Framework for Information System Research (RFISR) [NCP90], categorized by Research Question (RQ)s and sections where they were applied. 24

6.1 Expert user study results. 113
6.2 Questions and answer possibilities from the questionnaire 118
6.3 Results from the questionnaire to the 7-point scale questions 120
6.4 Results from the user study about the virtual assistant. 123
6.5 Results from our user study. 129
6.6 Outcomes from the questionnaire for each type of AR nugget 134
6.7 Outcomes for Q1 and Q2 139
6.8 Outcomes for questions asked for each type of AR nugget. 140

Chapter 1

Introduction

Augmented Reality (AR) allows augmenting physical objects or environments with virtual content, e.g., augmenting an object with a virtual 3D model. AR is used in many applications, for example, in education [Den+21; LTK18; Fit+13; RG13; BD12], medicine [Mue+19a; SVN19; Kam+14; CCJ10], cultural heritage [MT+21; SSL21; Ham+20; Bob+19; Jun+16; Ghi+09], navigation [Ham+20; BRA19; GM19; RP18; VZL17], or industrial contexts [KP21; Spe+15]. While there is a great interest in AR applications, challenges in their authoring process and usage remain. This work aims to contribute to easing the use of AR for everyone and the authoring of AR experiences for persons without programming knowledge.

For this, we draw from the educational concepts of microlearning [Hug05] and Virtual Reality (VR) nuggets [HD19b]. In microlearning, learning units are divided into small, independent learning nuggets. VR nuggets apply the idea of small, independent learning units to VR. However, AR incorporates challenges that VR nuggets cannot meet. For example, AR requires information about the users' physical environment but VR nuggets cannot gather and process the required information accordingly. AR applications need to detect and track physical objects and environments and deal with changing environments. Additionally, AR applications need to support interactions that differ from VR interactions. This work applies a nugget concept to AR by introducing self-contained pattern-based AR applications called AR nuggets.

1.1 Motivation

AR can versatilely be used in numerous use cases to add value. Yet, creating AR applications requires special expertise and knowledge, which creates high barriers to getting started [Mac+04a; GM14; NS18; Ash+20; Kra+21]. The authoring of AR applications has been researched, and several AR authoring tools have been developed over the last decades (e.g., [Sch+02; Mac+04a; Ha+10; WTS10; Ram+13; PLP18; MT+21; Lav+21]). Still, many barriers in the AR authoring process remain, as pointed out in recent work, e.g., [Spe+18; NS18; Ash+20; Kra+21]. In addition, using AR applications can also involve several challenges [WLF07; PRD09; WCL19], especially if users are not familiar with the interactions that the AR application supports. Users need personalized guidance and active support to get to know how to interact with the AR application.

In the following, we describe two example scenarios that illustrate challenges that people without or with little knowledge of programming and AR face when creating or using AR applications. Over the course of this thesis, we address these examples by applying our approaches to them.

Example scenario 1: AR in a museum

A natural history museum wants to use AR glasses to enhance its whale exhibition. The museum curator is an expert in whale anatomy but has never worked with AR. Anyway, she decides to try authoring the AR application herself with an authoring tool for non-programmers. She spends a lot of time working with the authoring tool. When she is ready to try her authored application on her AR device, she realizes that the authoring tool does not support location-specific AR. Therefore, she would need to re-create her application with another authoring tool. Frustrated, she hires external developers to create the AR application instead. Still, she needs to invest time and resources in regular meetings with the developers because they need her knowledge about the whales to develop the application properly from a scientific and educational point of view. Finally, the AR application is developed professionally and thus offers qualified opportunities for museum visitors. However, when the visitors try the application, they have difficulties interacting with it using hand gestures or voice commands. The museum curator needs to stand next to the visitor and explain how to open menus, click buttons, and much more. At a later point, she decides to implement a similar AR application for another exhibit. Again, they need to consult AR experts and invest resources, although the functionality of the AR application remains the same.

Example scenario 2: AR for CME

In Germany, physicians must participate in continuing medical education as medical quality assurance. One option for participation is learning courses. A company that develops learning courses for CME uses mainly text, images, and videos to communicate the latest medical findings and knowledge to their learners. They find that some information, e.g., anatomic structures, are difficult to explain on a 2D screen and would like to visualize these in 3D. The company contacts AR experts to discuss how AR can help in their case. The AR experts suggest a couple of AR applications, both parties agree on one idea, and the AR experts implement it. Then, the enterprise adds the educational course with one AR element to its learning platform. Most persons using the course have never used AR before and, therefore, struggle using the application. Consequently, the company decides to include a tutorial in its application. Researchers also find new medical information that must be updated within the course. Again, the enterprise needs to hire external developers, although only one small part of the AR application needs to be updated. The company publishes the updated course with the tutorial, and now users have fewer difficulties. However, some users still face challenges in using the application. When they face a challenge, they do not remember what they learned in the tutorial at the beginning.

Both examples show that an AR application's authoring process is complex and requires expertise in multiple fields and domain-specific knowledge. In our example, the AR authors faced the following challenges: a) They started with an authoring tool that does not support their idea entirely and only found out so at a later point, b) Even for a slight change in the AR application, they needed to consult AR and programming experts, c) AR applications cannot be reused for similar scenarios, d) They do not know where AR can add value or how to use AR's strengths and possibilities of AR.

This thesis's goal is to target the challenges pointed out in the two example scenarios to contribute to making AR more accessible to authors and users. By this, we aim that authors can use the strengths of AR. This thesis approaches the following challenges.

1. VR nuggets targeted some similar authoring issues. However, it is unclear if and how a nugget concept can also be applied to AR. To transfer the idea of learning nuggets and VR nuggets to AR, it is also a prerequisite to find patterns that can serve as a basis for AR nuggets. Yet, it is open if such patterns exist.

2. Furthermore, it is unclear if and how AR applications can include additional features that can contribute to making them sophisticated. For example, AR could extend a digital learning course or replace one or more lessons from the course while other learning lessons remain. Then, it should be possible to combine multiple AR applications with each other or with other interactive digital applications, e.g., with VR nuggets. Additionally, AR applications could provide personalized guidance and active support to users. However, it remains open if and how AR nuggets can be conceived in a way that allows this.

3. In AR, location-specific content and tangible interaction are relevant. Yet, both are challenging for authors to implement and can also involve challenges for users. It is unclear if and how AR nuggets can target location-specific content and tangible interactions.

4. Finally, it is unclear if AR nuggets can support authors in meeting the authoring challenges described above. Accordingly, the challenge lies in conceiving AR nuggets so that they can provide a workflow and tools to adapt, deliver, and execute AR nuggets. This should not require programming or AR experience and be accessible to persons without such experience.

1.2 Research Questions

This thesis explores how pattern-based components can support the authoring process and usage of AR. In this thesis, we address the following four RQs and describe them further with sub-questions.

RQ1 **How can a nugget concept be applied to AR?**

  (a) How can AR nuggets be defined?

  - (b) What universal patterns for developing an AR application can be identified (and applied)?

RQ2 **What features are included in AR nuggets?**

  - (a) How can AR nuggets support users in their AR experience?
  - (b) How can AR nuggets be combined with VR nuggets?
  - (c) How can multiple AR nuggets comprise one larger AR experience?

RQ3 **How can AR nuggets address location-specific content and tangible interactions?**

  - (a) What tangibles are suitable for being used in AR nuggets?
  - (b) How can AR nuggets include location-specific content?

RQ4 **How can AR nuggets support authors in developing qualitative valuable AR applications?**

  - (a) What can the workflow for the authors look like?
  - (b) What tools can support authors in adapting AR nuggets?
  - (c) How can AR nuggets be delivered and executed on an AR device?

1.3 Methodology

Our RQs can be classified into knowledge questions and design questions after the empirical research model for software engineering research by Easterbrook et al. [Eas+08]. While knowledge questions aim to gather knowledge about a current state, design questions target developing new procedures and tools. To solve design questions, associated knowledge questions about the design problem need to be solved. Our first RQ is an exploratory knowledge question and followed by design questions (RQ2 - RQ4). We address our design questions with a user-centered approach based on the RFISR by Nunamaker et al. [NCP90]. The RFISR defines the research phases as Observation, Theory Building, Systems Development, and Experimentation. Table 1.1 describes how this work applies these research phases to its individual sections and the RQs.

For each of the research phases and our RQs, we define one research goal, resulting in four research goals per RQ. In the following, we list these research goals with regard to their RQs.

RQ1: How can a nugget concept be applied to AR?

1. Observation: With literature research, we investigate how nuggets are defined in other application fields, e.g., learning nuggets in e-learning and VR nuggets. Furthermore, we discuss with domain experts from a museum, the event organization domain, and the medical domain what use cases their domains have and how to apply AR there. By this,

we identify patterns from the application domain that could be suitable for AR and AR Nuggets.
2. Theory Building: We transfer concepts from learning nuggets in microlearning and VR nuggets to AR. Additionally, we develop concepts on how to target challenges for applying nuggets and patterns to AR.
3. Systems Development: We implement AR nuggets that reflect our concept for exemplary patterns.
4. Experimentation: We evaluate the concept and implemented AR nuggets with expert interviews, user studies, as well as a real learning course and a real museum application as proof of concept applications.

RQ2: What features are included in AR nuggets?

1. Observation: We observe users of AR applications to identify challenges they face and situations where they need assistance. Additionally, we identify where AR nuggets could incorporate features to further enhance the user experience.
2. Theory Building: We extend our concepts for AR nuggets to support users in the identified situations and to enhance the user experience.
3. Systems Development: Within our AR nuggets, we implement exemplary functions that target to enhance the user experience and assist users.
4. Experimentation: We evaluate the prototypes with a user study.

RQ3: How can AR nuggets address location-specific content and tangible interactions?

1. Observation: We research how tangibles are designed for AR applications and identify challenges with these tangibles for specific use cases. Additionally, we highlight challenges that authors face when positioning location-specific AR elements in the real world.
2. Theory Building: We develop concepts that apply tangible interactions and location-specific positioning to AR nuggets. For tangible interactions, we also explore what kind of tangible is useful in different types of AR nuggets. Regarding the authoring of location-specific content, we conceptualize tools to support authoring on-site and off-site.
3. Systems Development: We create the tangibles and implement the tools in a prototype.
4. Experimentation: We evaluate the prototype with a user study.

RQ4: How can AR nuggets support authors to develop qualitative valuable AR applications?

1. Observation: We research the state of the art regarding current authoring tools and authoring workflows. Hereby, we identify challenges that still exist in the authoring process.
2. Theory Building: We develop concepts on how AR nuggets can approach the identified challenges in the authoring process of AR applications. We also create an authoring workflow for how non-programmers can author AR nuggets.

3. Systems Development: We develop an authoring tool for adapting AR nuggets to create custom AR applications without programming. Furthermore, we develop prototype tools that target individual authoring steps.
4. Experimentation: We evaluate our authoring tool in a user study with non-programmers. For our further prototype tools, we carry out an authoring workshop where we introduce the tools to the participants. Then, the participants use the tools to create their own AR applications based on AR nuggets.

1.4 Contributions

This thesis introduces AR nuggets by transferring the idea of VR nuggets to AR (RQ1a). Hereby, we identify and target AR-specific challenges as well as challenges that are not targeted by VR nuggets. Additionally, we identify and introduce ten application patterns and implement corresponding AR nuggets (RQ1b). In an expert user study, we showed exemplary AR nuggets to experts from the medical domain. These experts see added value from AR nuggets to CME and can imagine using AR nuggets themselves.

This thesis further integrates complex features into AR nuggets that target to support authors and users of AR nuggets (RQ2). For example, it conceptualizes and implements an AR nugget manager based on pre- and postconditions to support the creation and usage of multiple non-linear AR nuggets together (RQ2a). Additionally, we identify and implement patterns for spatial connections between AR nuggets. We show how persons without programming knowledge create a non-linear AR application for a museum based on AR nuggets using the AR nugget manager. Furthermore, we introduce transitions between AR and VR nuggets (RQ2b). Finally, we introduce a novel virtual assistant that proactively supports users of AR nuggets (RQ2c). In a user study, participants understood how to interact with the AR application after seeing the hints from the virtual assistant, including participants who never used AR before.

Regarding RQ3, we include functions for tangible interactions within AR nuggets. Here, we design a universal tangible that can universally be used for multiple use cases and compare it with realistically shaped tangibles. The latter are designed for specific use cases and are similarly shaped as the virtual 3D model that augments them in the AR application. We conduct a user study that indicates that both tangible shapes have specific benefits relating to their objectives. A realistically shaped tangible has more advantages when users focus on moving the tangible to view it or its augmentations from all sides. The universal tangible is more advantageous when the users mainly view one side of the tangible, e.g., because they watch an animation that is augmented to this side. For location-specific content, we introduce tools to place the location-specific content on-site and to save these locations to the AR nugget (RQ3c).

In RQ4, we focus on authoring challenges and propose an authoring workflow for AR nuggets that can be applied to our authoring tools (RQ4a). Furthermore, we develop an AR nugget authoring tool that is accessible without programming knowledge and applies our proposed authoring workflow (RQ4c). It uses a smartphone as an immersive device and a desktop computer as a non-immersive device. Using immersive as well as non-immersive devices allows authors to test AR nuggets immersively while also being able to adapt them on a desktop computer with traditional input methods, e.g., a keyboard. In a user study, authors perceived the AR nugget approach in the authoring tool as helpful to get started with AR applications. Additionally, we develop one authoring tool based on placement constraints to support authoring for yet-unknown environments. Placement constraints can be used to anchor AR nuggets to surfaces that meet certain constraints. For example, instead of placing an AR nugget at a specific Point of Interest (PoI), an AR nugget could be constrained to be anchored to a wall with a specific surface area. Instead of requiring authors to define placement constraints, our tool calculates the constraints automatically. For this, the author places all AR nuggets in one room that serves as a template. Then, the AR nuggets can automatically calculate their placement constraints. Besides stand-alone authoring tools, we also introduce additional AR nugget authoring tools that are integrated into a Game Engine. Finally, we contribute the functionality to export and import AR nuggets in an AR nugget exchange file (RQ4c). To integrate exported AR nuggets into other applications, we develop a content delivery system (RQ4c).

1.5 Thesis Structure

This thesis is structured as follows. The next chapter presents related work regarding AR, patterns, tangibles, and AR authoring. We also discuss the related work to point out research gaps. Based on this, we introduce AR nuggets in Chapter 3. Here, we also describe how AR nuggets can facilitate tangible interactions, support users, be combined with VR nuggets, and be used in complex environments like large buildings with PoIs distributed across many rooms. We implement these concepts in Chapter 4. Next, Chapter 5 focuses on authoring AR applications with AR nuggets by introducing multiple authoring concepts and tools based on AR nuggets. We evaluate AR nuggets from a user's and an author's point of view in Chapter 6. In Chapter 7, we discuss our overall findings by reflecting on the RQs and discussing our contributions to the RQs. Finally, Chapter 8 concludes and points to directions for possible future work.

Table 1.1: Research phases classified based on the RFISR [NCP90], categorized by RQs and sections where they were applied.

Research Phase	**RQ**	**Section**	
Observation	RQ1-4	1.1	Motivation
Observation	RQ1	2.2	Applied Patterns
	RQ3	2.3	Connection to the Real World
	RQ2	2.4	User Assistance
	RQ2	2.5	Transitioning to and from Augmented Reality
	RQ4	2.6	Creating Augmented Reality Applications
Theory Building	RQ1	3.1	Definition
	RQ1	3.2	Examples
	RQ3	3.3	Utilization of Tangible Interactions in AR Nuggets
	RQ2	3.4	Integration of User Assistance in AR Nuggets
	RQ2	3.5	Usage of Multiple AR Nuggets in Complex AR Setups
	RQ2	3.6	Combination of AR Nuggets and VR Nuggets
System Development	RQ1	4.1	AR Nuggets
	RQ3	4.2	Utilization of Tangible Interactions
	RQ2	4.3	Integration of User Assistance in AR Nuggets
	RQ2	4.4	Usage of Multiple AR Nuggets in Complex AR Setups
	RQ2	4.5	Combination of AR Nuggets and VR Nuggets
Theory Building	RQ3, RQ4	5.2.1	Integrated AR Nugget Authoring Tools
System Development	RQ3, RQ4	5.2.2	
Theory Building	RQ4	5.1.1	Using Different Degrees of Immersion
System Development	RQ4	5.1.2	
Theory Building	RQ3, RQ4	5.3.1	Constraint-based Authoring with AR Nuggets
System Development	RQ3, RQ4	5.3.2	
Theory Building & System Development	RQ4	5.4	Content Delivery
Experimentation	RQ1	6.1	AR Nuggets
	RQ3	6.2	Utilization of Tangible Interactions
	RQ2	6.3	Integration of User Assistance in AR Nuggets
	RQ2	6.5	Combination of AR Nuggets and VR Nuggets
	RQ2	6.4	Usage of Multiple AR Nuggets in Complex AR Setups
	RQ3, RQ4	6.7	Integrated AR Nugget Authoring Tools
	RQ4	6.6	AR Nugget Authoring Using Different Degrees of Immersion
	RQ3, RQ4	6.8	Constraint-based Authoring with AR Nuggets
Experimentation	RQ1-4	7	Discussion

Chapter 2

Related Work

This chapter analyzes and discusses related work that addresses the application of patterns, tangible and location-specific AR, and the authoring process when creating an AR application. Each section in this chapter analyzes literature to discuss the contributions from related work and where research gaps remain. The last section summarizes and discusses the findings and research gaps.

2.1 Augmented Reality

AR enhances a user's perception of the real world by augmenting the physical environment with digital information. Additionally, some definitions of AR describe the technology further. One definition that prevails in the scientific community is based on work by Azuma [Azu97]. In contrast to VR, users in AR can see their real environment while parts of it are augmented with virtual elements. Azuma characterizes AR with the following three aspects: 1) AR augments a physical world with virtual information, 2) AR is interactive in real-time, and 3) AR registers the physical environment to correctly place virtual elements in 3D. Azuma's definition contrasts AR with VR. While the second aspect applies to both, AR and VR, the other two factors apply to AR only and distinguish AR from VR.

In contrast to VR, the goal of AR is not to shield users in a completely virtual environment. One goal of AR can be to fuse the virtual and physical elements to the extent that users feel that the virtual elements are part of the real world [RS02]. The user's feeling of virtual elements from the AR application being present in the real world is called presence [RS02]. Lombard and Ditton [LD97] explore the concepts of presence and introduce different types of presence. Their presence type "it is here" can be applied to AR and is described as follows: "Instead of transporting the user to a different place, a sense of presence may bring the objects and people from another place to the media user's environment." [LD97]. Applied to AR, if users feel that virtual objects are in the same place as themselves, they feel presence [RS02]. In the context of AR, it is often referred to as "spatial presence" [KB97; RS02; Sla02] Several questionnaires to measure presence in VR exist [SUS94; WS98; SFR01], and more recently, questionnaires that specifically target AR have been developed [Har+16; GK17].

2.2 Applied Patterns

This work utilizes patterns and applies them to AR. Patterns describe reusable solutions for specific recurring problems. Pioneering work about patterns by Alexander et al. [AIS77] uses patterns to describe best practices from the domain of architecture. The idea of patterns was transferred and applied to several other domains, e.g., pedagogic ([Sal99; Ber+12]), linguistics, or psychology. Additionally, patterns are commonly used in software engineering [Gam+95]. Gamma et al. [Gam+95] distinguished three types of patterns for object-oriented design: creational patterns, structural patterns, and behavioral patterns. Creational patterns offer solutions to create, assemble, or represent objects. Structural patterns organize objects and their classes to build a larger system, e.g., utilize inheritance mechanisms. They can also combine objects with each other or integrate an object into another system. Behavioral patterns control how objects communicate with each other and assign responsibilities to the objects.

To allow other persons to reuse one's patterns, patterns can be documented in a structured form. How to structure the patterns' documentation depends on the application domain. For example, a structured pattern documentation form for architecture [AIS77] includes some similar, but also some other elements than one for software engineering ([Gam+95; Cop98]). However, most pattern description templates are based on Gamma et al.'s [Gam+95] work [PM20].

Reicher et al. [Rei+03] and MacWiliams et al. [Mac+04b] proposed to use patterns to develop and document AR systems. Both identified patterns for system architectures in AR systems and collected them in a catalog. Based on work by Gamma et al. [Gam+95], MacWiliams et al. [Mac+04b] described each pattern in a scheme with its name, goal, motivation, description, and usability. They systematically order their nearly 50 identified patterns to show dependencies between them. Besides patterns for system architectures, there are also patterns for interactions in AR. For example, work by Lages and Bowman [LB19] identified interaction patterns, focussing on interactions for Handheld Device (HHD) while walking. Emmerich et al. [EKH17] identified interaction patterns that can be used and combined to create AR games. While there are patterns that describe AR systems or interactions in AR, there are currently no patterns that describe complete AR applications with all 3D objects and interactions.

Horst and Dörner [HD19b] apply patterns to VR and use them to describe complete VR scenes. They identified patterns from the application domain as a basis for VR applications for recurring scenarios. Based on these, they introduced VR nuggets. The term nuggets is based on learning nuggets in microlearning. Microlearning is an educational approach that divides learning content into so-called learning nuggets. Learning nuggets are small, self-contained learning units, each with a single learning goal that can be accomplished in a short amount of time [Hug05; Bai+06]. Various types of media can be applied to microlearning, e.g., books [LCN19], videos [GW17], audio recordings [Bea+07], mobile applications [Bea+07; SA14; MR17], or VR [HD19b]. Similar to learning nuggets, each VR nugget has one learning goal that can again contribute to one higher learning goal. Additionally, VR nuggets are independent of each other and can be customized by authors.

In microlearning, Hug [Hug05] distinguished three different levels in microlearning nuggets by their complexity. On a micro level, nuggets have the lowest complexity and size. Examples are single letters or vocables regarding a specific topic. Nuggets on a meso level are more complex and can be reflected by a whole topic or lesson. The highest complexity have nuggets on a macro level, e.g., a full course or curriculum. Similarly, Horst and Dörner [HD19b] distinguished micro, meso, and macro VR nuggets. Multiple VR micro-nuggets can form a VR learning lesson in the form of a VR meso nugget, and multiple VR meso-nuggets can form a larger educational course in the form of a VR macro-nugget.

Horst et al. also introduced a pattern description scheme for VR nuggets [Hor21]. It describes each pattern with (1) a name, (2) its concern, (3) an illustration in the form of a conceptual image, (4) an example image or screenshot, (5) a few sentences that describe when to use the pattern, (6) a short paragraph that explains the pattern's visual components, (7) interactions that the pattern includes, (8) an example scenario where the pattern can be applied, and (9) what is needed to adapt the VR nugget. In contrast to other structured forms of pattern documentation, patterns in the description by Horst et al. [Hor21] include no references to other patterns. Instead, the underlying patterns are stand-alone and independent from each other, similar to VR nuggets. However, this structured form cannot document patterns for complete AR scenes because it needs to include information about the requirements of the physical environment.

2.3 Connection to the Real World

AR integrates virtual elements into the real world. It registers the physical environment in 3D to gain information about the real world. Using this information, virtual objects and other virtual elements can be augmented to physical surfaces, images, 3D objects, or anywhere in the environment. For example, an AR application could allow users to place a virtual object on any surface in their environment. Such AR applications can be experienced anywhere and are not location-specific.

In contrast, location-specific AR applications are linked to one specific location. Examples are AR applications that augment specific exhibits in a museum, which cannot be experienced from another place than the museum (e.g., [Ghi+09; Ham+20]). Subsection 2.3.2 describes related work for location-specific AR applications.

AR applications can also be linked with tangible objects. A virtual object can augment a physical one, and interactions can be mapped to them [BKP08]. Subsection 2.3.1 elaborates on this link between real and virtual objects called tangible AR [BKP08].

2.3.1 Tangible Augmented Reality

Tangible Augmented Reality is one way to design intuitive interaction techniques. Work by Billinghurst et al. [BKP08] states that traditional tangible user interfaces (UI) support controlling data through buttons or sliders but have limited support to view virtual 3D objects. The authors applied design principles from such tangible UIs to tangibles in AR. They defined tangible AR interfaces as interfaces where all virtual objects are anchored to physical objects, and users can manipulate the tangibles to interact with the virtual objects. Users can, e.g., move and spin a physical object to move and rotate virtual elements in the same way. Studies [LRS10; DLB15; Ssi+19] showed that manipulating physical tangibles can give users a more natural feeling than using freehand interactions. For example, cubes are used as tangible objects in several application fields [Jim+15].

Tangibles with a shape similar to the virtual object that augments them (realistically shaped tangibles) can merge visual and kinesthetic perception, increasing realism and simplifying manipulation [KKL09; SVG15]. One way to create realistically shaped tangibles is by incorporating 3D printing. For instance, the VIVATOP project [Lüc+20] used 3D-printed organ phantoms to improve surgical results by providing realistic haptic feedback. It created 3D prints of organs to support surgeons in practicing operations and measured the in vivo organs' physical hardness, touch, and palpation to apply them to the 3D-printed organ phantoms. Also in a medical context, Münder et al. [Mue+19a] presented an idea of how to use tangible AR in the three phases of an operation (preoperative planning phase, during the operation, and for training).

However, creating realistically shaped tangibles can be time- and resource-consuming. Münder et al. [Mue+19b] explored how haptic fidelity affects immersion, performance, and intuitive interaction. They created tangibles with three different levels of haptic fidelity: realistically shaped but longer creation time, universal disc-shaped tangibles, and lego-built tangibles. In a user study, they showed that the realistically shaped 3D printed tangibles perform best, but the Lego tangibles are a good trade-off, allowing fast creation of the tangibles while having sufficient fidelity.

Henderson et al. [HF08] used physical objects that are already present in a user's environment to provide haptic feedback. Their application, called 'Opportunistic Controls,' uses present props (e.g., screws or bolts) as buttons to create affordances. Follow-up work [HF10], described examples where 'Opportunistic Controls' were applied and tested in a user study. Similar to the work by Datcu et al. [DLB15], users preferred the tangible interface.

Work by Hettiarachchi et al. [HW16] similarly integrated the users' environment. The application, called 'Annexing Reality,' searches the user's environment for physical objects that match virtual objects in shape and augments them on the matching physical objects. Then, the virtual objects are scaled to fit the physical objects' sizes. This way, the users do not need to carry a tangible, and instead, the present environment and its props are used while the best available haptic feedback can be provided. In a user study, content creators found Annexing Reality useful.

From a technical point of view, there are several toolkits and SDKs that implement computer vision techniques to support detecting and tracking 3D objects for AR, e.g., Vuforia or ARToolKit. These distinguish between detecting and tracking. First, the computer vision algorithms aim to detect the tangible in at least one of the camera frames. Once a 3D object is detected, the tracking is started.

2.3.2 Location-Specific AR Applications

With location-specific AR applications, authors are faced with the challenge of placing and anchoring virtual content at the correct PoI in the physical world. This can become a tedious and time-consuming task, especially if there are a lot of objects that are anchored on individual PoIs in the real world [BRA19]. Options to anchor virtual objects are by detecting and tracking a known image, object, or area or using Simultaneous Localization and Mapping (SLAM) in an unknown environment [LM97] (e.g., with Mixed Reality Toolkit (MRTK)'s spatial anchors or ARToolKit).

For example, Kampa and Spierling [KS17] used images of the physical environment, like an image of an ancient building, to overlay it with cut-out video scenes on an AR HHD. Their two-phased authoring process distinguishes between working at an office desk to create a storyline and a locative part where authors match AR content with the environment at the target location using mobile tools. For the locative part, Kampa and Spierling [KS17] introduced a toolset that supports authors in capturing pre- and postconditions and creating placeholder content on site. Using these pre- and postconditions, multiple AR scenes can be connected to one non-linear story.

Bachras et al. [BRA19] used specific points (called spatial points) in the real world to calculate pathways for navigation purposes. They state that development effort increases with more spatial points, but it can increase user performance and enhance the user experience. Thus, they suggested using fewer spatial points for short paths and more for longer paths.

Moreover, it can also be challenging to describe pathways from one PoI to another. If the PoIs are not located in the same room, guiding users from one PoI to the next one can be necessary. In contrast to VR, in AR, it is not possible to control the whole environment that the user sees, but only the virtual elements. Thus, authors of location-specific AR applications typically cannot rearrange PoIs and, therefore, need to have users guided to the PoIs at their location. For example, a human guide can navigate visitors in a museum to different exhibits. One alternative to human guides or signs can be a virtual guide [Ghi+09; Ham+20]. Virtual guides are available on demand so that users do not have to wait for a human guide to be available. Additionally, they can tailor the guidance to the users' individual needs. For example, a virtual guide in a museum exhibition can navigate visitors on pre-planned pathways, thus reducing the need for human museum guides [Ghi+09; Ham+20]. Moreover, utilizing a virtual guide for navigation purposes is more efficient compared to alternative methods of virtual guidance [Cam+14; Nat+20].

Hammady et al. [Ham+20] introduced a museum application with a virtual guide. The authors observed museum visitors and then categorized them into four groups with different behaviors. From these, they identified 12 features to implement in 'MuseumEye,' their digital museum guide. Within the museum, the authors strategically positioned virtual guides at various PoI. These guides explain the exhibits to visitors and provide supplementary information directly on-site. Yet, the work did not implement an agent that guides visitors from one PoI to another. Similarly, Martí-Testón et al. [MT+21] developed an AR-based museum application that incorporates a virtual guide, but the guide only appears on the PoIs when the user is close enough and does not support users in finding the way from one PoI to another.

If an AR application is going to be experienced in a currently inaccessible, unknown, or constantly changing environment, then it is not only time-consuming to place virtual objects but could also be not possible at all. Targeting this challenge, Singh et al. [Sin+21] implemented a story authoring tool with a graphical UI for non-technical authors called ’Story CreatAR.’ Applications created with Story CreatAR can either be experienced on-site in AR or simulated in VR. Authors can define ’placement rules’ to specify where virtual elements should be placed in the physical world. The ’placement rules’ are based on the spatial characteristics of visual complexity, openness, and visual integration of a room. In an iterative design process, concerns were raised about whether authors would be able to understand and use the spatial characteristics. Thus, Singh et al. [Sin+21] implemented spatial ’attributes’ that combine spatial characteristics. For example, virtual objects that the author wants to be difficult to find can be placed using the attribute ’hidden’ with the spatial characteristics of low visual complexity and low visual integration. In contrast, a virtual object that should be easy to find can be placed in an area with high visual integration using the ’easy to find’ attribute. The authoring tool focuses on stories where the user interacts with virtual agents. Agents can be placed relative to each other using ’formation rules.’ The agent moves based on the storyline and on ’traversal rules’ that the author can decide on. The ’traversal rules’ set the agent’s start and goal as well as its walking speed. It calculates pathways using the A* algorithm [HNR68], a common pathfinding algorithm that finds the shortest path to one goal. To find points in the physical room that match the rules, ’Story CreatAR’ can analyze a floor plan and calculate the spatial characteristics. However, Singh et al. [Sin+21] left open how such a tool can react to environments without a floor plan or changing environments.

The application FLARE (Fast Layout for Augmented Reality) [Gal+14] also uses constraints to place virtual objects in the users’ physical environment. It allows an AR application to adapt to unknown or changing environments on the spot by calculating layouts for the AR application in real-time. When FLARE is started, it creates a layout that fulfills the constraints and defines where virtual objects are placed. FLARE allows authors to define constraints regarding the relation between multiple virtual objects, the relationship between virtual elements and users, and between virtual objects and the users’ physical environment. Constraints can include whether to place a virtual object on a vertical or horizontal surface or floating in the room, a virtual object’s position and rotation relative to a surface, and a range for an allowed scale. However, it is not possible to use semantic object recognition, i.e., to define specific categories of horizontal or vertical surfaces like desks, floors, ceilings, or walls.

2.4 User Assistance

Interactions with virtual objects, physical tangible objects, AR devices, or interactions in AR in general can also challenge users, especially if the users have no or little experience with interactions in AR. AR applications and their authors can support users in facing such challenges. One typical approach to help users interact with AR applications is to provide a tutorial. A tutorial can, e.g., be realized with textual hints [Gra+09; MGF11; Kim+14], images [Gra+09], videos [MGF11; Pon+11; Kim+14], or as an interactive tutorial [GMF10; FGF11].

While tutorials can support users, using them can also include challenges. First, users need to browse for and find a tutorial, which can be a tedious task [Kim+14]. Here, one can argue that users who struggle with operating the target application are not always able to realize their struggle and start a tutorial on their own. However, always going through a tutorial at the start of the target application may be helpful for only a few of the users. There are different types of users that have different levels of experience using AR applications [SP10; Che+17]. A novice user might need more assistance, while an experienced user might feel disrupted by tutorials.

One idea to accommodate the different needs of individual users could be to provide degrees of assistance, i.e., give novice users more assistance while not disturbing experienced users with hints. Second, following a tutorial while simultaneously using the target application [Pon+11] can be difficult. Users either need to switch back and forth between the target application and tutorial [Pon+11], or to memorize everything they learned in the tutorial in order to be able to operate the target application afterward.

One approach is to provide a tutorial or hint only at specific points in time. For example, Wu et al. [WCL19] proposed a method to improve user guidance in the process of AR-based object scanning. In their prototype, users interact with an AR HHD to scan a physical object, with the goal of creating a 3D model. The scan quality, and thus the 3D model's quality, are reduced if users shake or move their AR device too quickly or do not hold the AR device within the correct distance to the scanned object. Therefore, the authors aim to support the users in handling the AR device as their application requires. They aimed to improve the scan results by showing visual hints on the AR device and used icons that are displayed on the left and right edge of the user's field of view as visual hints. The hints warn in case of overspeeding while rotating, correct the capture distance, or instruct to rotate the object. They are only shown if the system detects that a problem in the scanning process is occurring. For example, when the scanned object is held too far away, the corresponding hint is shown. In a user study [WCL19], the authors showed that their system performs better in usability, understandability, and satisfaction with the reconstructed scanned model than three other state-of-the-art approaches. This shows that warnings and instructions on demand can support users. Similarly, the reconstruction application 'ProFORMA' [PRD09] aims to guide users during the model reconstruction process. It augments the 3D model with arrows that direct the user to the faces of the 3D model that were not scanned yet. While both works support the users during the scanning process, they still presume a certain amount of experience with the application. The applications focus on the specific use case of scanning and reconstruction but do not support users with instructions or hints about how to interact with UI elements like the menu. The displayed hints are all shown as icons without any textual or audible description and require the users to understand the icons on their own.

Work by White et al. [WLF07] aimed to support users interacting with tangibles. It distinguished five ways to represent visual hints: textual, diagrammatic, ghosted, animated, and composite hints. Similar to works by Wu et al. [WCL19] and Pan et al. [PRD09], it does not show all hints at all times. Instead, it implemented ways to activate and deactivate the hints.

For example, if users do not move the tangible, a hint can be triggered. However, this can be triggered inadvertently if the users hold the tangible still to look closer at and study its virtual augmentation. In White et al.'s work [WLF07], hints can also be triggered by an activation gesture (e.g., shaking) or with a button press. The authors combined their different representations of visual hints and activation methods in a prototype they evaluated in a user study. They found that combining different methods is a successful form of user guidance. However, the visual space around a tangible is limited. Therefore, it is impossible to show hints from each category at once. For example, there was only space for one ghosting or animated hint at a time to explain one gesture, while diagrammatic and textual hints can more easily be placed next to each other.

2.5 Transitioning to and from Augmented Reality

Transitions are one way for users to smoothly start or end an AR experience. There are numerous studies showing that transitions from the physical world to VR can enhance VR experiences, e.g., by improving distance perception [Ste+09; Ste+10], increasing presence [JWH18], user awareness [VF17], or task performance [Fel+19].

Jung et al. [JWH18] explored the effects of gradual transitions between the physical world and VR, where the user sees a virtual replica of the physical world in VR. They distinguished between three physical-mental transition stages: putting the HHD on, making the mental shift, and fully entering the VR environment. The authors called one stage during the mental shift the 'Limbo.' During the Limbo stage, the user adjusts to the gap between the physical and virtual space while making assumptions about the emerging virtual environment. Jung et al. [JWH18] compared how gradual transitions affect the users' perception of presence compared to instantaneous transitions. They state that the Limbo phase has a significant influence on the perceived presence. Similarly, work by Steinicke et al. [Ste+09; Ste+10] demonstrated that transitions from the physical environment to VR can enhance the users' perceived presence and ability to accurately perceive distances within the VR environment. It also used a virtual replica of the users' physical environment to transition. In a user study conducted by Feld et al. [Fel+19], users performed a task while wearing a VR headset. Contrasting to prior work that did not include tasks, the users in this study valued the transition's efficiency over its interactivity.

Horst et al. [Hor+21b] applied transitions to short, consecutive VR experiences, where users typically put on and take off HHDs frequently. For example, in a presentation with text, images, and videos on slides, presenters could replace single slides with VR applications [HD19a] while leaving the rest of the presentation as it is. Then, only the parts of the presentation where VR adds value are experienced in VR, and the audience puts the HHDs on the experience one VR part and then takes the HHDs off again. Horst et al. [Hor+21b] referred to transitions from the physical world to VR as 'intro transitions.' To transition from a VR experience back to the physical world, the authors introduced 'outro transitions.' They argue that outro transitions are similarly important as intro transitions, especially for short, consecutive VR experiences. From a conceptual point of view, the authors divide their outro transitions into three phases: initiation, interlude, and exit. The initiation triggers the transition. For example, a transition could be initiated by pressing a button. It can be performed by the user or an external person, e.g., a presenter. The interlude is an optional state that aims to make the transition less abrupt. Here, visualizations or interactions that are not part of the original VR scene can become the user's center of attention. Common transition metaphors here are teleporters or doors. In the exit phase, the transition tells the user to take the HHD off and end the VR experience. Furthermore, Horst et al. [Hor+21b] used the outro transitions as a signal for users to take their HHDs off without the need for a third person to tell them to do so. Additionally, Horst et al. [Hor+21b] implemented eight different outro transitions and evaluated them in a user study. The study showed that the outro transitions can support presenters in signaling to the users to end a VR session without having a negative impact on presence or the overall experience.

While Horst et al. [Hor+21b] researched transitions between VR experiences and the users' physical environment, other researchers combined transitioning techniques between AR and VR [ESE06; CES22; Poi+22]. For example, Eissele et al. [ESE06] implemented one VR prototype that makes use of different degrees of virtuality. They stated that both, AR and VR, can help to approach challenges in production environments. Their prototype uses VR glasses and incorporates the VR glasses' camera stream to include the real world, resulting in a single application with AR and VR parts.

Cools et al. [CES22] presented four techniques for transferring objects between various augmented and virtual environments. They conducted a user study that showed that users can manipulate and transition objects significantly faster if one of their transition techniques is used. Based on the findings of their study, they implemented a fifth transition technique and carried out a second user study. With the results from both studies, the authors stated that it depends on the user's task how efficient a transition is. Finally, the authors developed guidelines for designing transitions between AR and VR. However, the transitioning techniques cannot be directly applied to larger environments, like museums, as the authors' focus is on transitioning small handheld objects. The authors also did not evaluate the transitions' impact on presence.

Pointecker et al. [Poi+22] implemented four transition techniques with the aim of achieving a seamless transition between AR and VR. The authors conducted an evaluation of their transitions through a user study, wherein participants analyzed a logistics network while immersed in VR. One result of their user research was that users unfamiliar with switching between realities benefit from prominent transitions. If users must be aware of their surroundings, prominent transitions are essential. The authors noted that their outcomes might vary from methods that employ transitions to and from reality.

These studies that combine AR and VR with transitions used the same HHD for the VR part and the AR part. Thus, users did not have to take the HHD off to switch environments. They realized this using current VR glasses that can run AR applications by incorporating a live stream from the HHD's camera and augmenting the live stream. However, a camera live stream does not resemble reality in quality, frames per second, delays, colors, or resolution. Therefore, AR glasses could be favorable over VR glasses for AR applications, and it could be useful to investigate transitions to or from AR and VR where HHDs are switched.

2.6 Creating Augmented Reality Applications

Creating AR experiences is a process that often involves multiple steps, persons, and challenges. Regarding the steps in the creation process, we distinguish content creation and authoring. Content creation is the creation of virtual elements that the AR application should include, i.e., creating assets like videos, images, textures, sounds, or 3D models with animations [Kra+21]. It often involves a variety of different tools, e.g., Adobe Photoshop to create a texture, and Blender; or 3ds Max to create a 3D model. Authoring, on the other hand, includes the implementation of tracking functionalities, physics simulation, testing, storytelling, and all other required steps to create AR applications. Subsection 2.6.1 describes the roles of persons who are involved in creating AR applications and what challenges they typically face. Several authoring tools have been developed that target to meet some of these challenges. Subsection 2.6.2 presents such authoring tools and analyzes how they target authoring challenges and where research gaps remain.

2.6.1 Challenges in the Authoring Process

Not only using AR can be challenging, but it can also be difficult to create AR experiences in the first place. This process is not only a challenge for persons without a technical background, but also professional AR/VR designers and developers face similar difficulties in the authoring process [Ash+20]. The authoring process involves a lot of different tools [NS18] that authors need to be able to work with. Authors need a variety of skills and knowledge for this [Bro+19]. For example, skills, knowledge, and experience that are necessary to create an AR application for a museum include domain-specific knowledge from a museum expert, design and 3D modeling skills to create 3D models or other assets, usability, and HCI knowledge to create a satisfying user experience, programming knowledge to implement the application, and many more.

However, a single person can hardly have all the experience necessary to fulfill all of these roles. As one participant in a user study by Krauß et al. puts it: "you cannot be an expert in everything. It requires collaboration. ... The need for collaboration in AR/VR is greater because of the complexity and the need for a number of different assets." [Kra+21]. In these interdisciplinary teams, communicating and using early prototypes is especially important. Therefore, AR authoring tools should support communication concepts and approaches to create a more usable toolchain.

Krauß et al. [Kra+21] interviewed 26 AR/VR designers and developers to identify how collaborative AR and VR applications are developed in practice. Their interviews highlighted that multiple persons with individual skills need to work together to create AR applications. They found that creating AR/VR experiences involves four roles: concept developer, interaction designer, content author, and technical author. Often, one person fulfills more than one of these roles.

Similarly, Myers [Mye95] described that one person can fulfill more than one role and that one role can be fulfilled by more than one person. Myers [Mye95] defined four roles of persons who are involved with UI software: 1) The end-user uses the resulting UI, 2) the UI designer creates the UI, 3) the tool creator creates tools that the UI designer can use, and 4) the application programmer writes the software.

For VR, Horst et al. [Hor+21c] identified and distinguished the following three author roles to create pattern-based applications: 1) system authors who implement fundamental basic runtime functionalities like rendering, a UI, or collision detection within an authoring tool, 2) pattern authors who identify design patterns (e.g., in interviews with domain experts) and integrate them into the VR authoring tool, and 3) content authors who use the authoring tool and design patterns to create their own VR applications, for example, by exchanging 3D models and inserting images or videos.

Ashtari et al. [Ash+20] interviewed 21 AR/VR creators to identify challenges and opportunities in current practices of creating AR/VR experiences. Their participants were hobbyists, domain experts, as well as professional designers from a variety of application fields. Based on the interviews, the authors identified eight key barriers:

1. participants do not understand the landscape of available tools and, therefore, do not know where and how to start,
2. it is difficult to find learning resources because participants do not understand the nomenclature,
3. there are no design guidelines or examples available that participants could not draw from,
4. it is difficult to design experiences that feel natural in motion, gesture, and audio,
5. participants cannot simulate or forecast how the AR/VR application's user moves and interacts,
6. it is difficult to design the AR/VR experience in an immersive, story-driven way,
7. participants cannot keep up with the constant changes in tools and technology and have no viable debugging tools,
8. participants face challenges in understanding how to test and evaluate their application or have no access to the AR/VR device.

Complementing the research from Ashtari, Krauß et al. [Kra+21] identified three key challenges for creating AR/VR experiences in interdisciplinary teams: 1) "misconceptions about the medium," 2) "lack of tool support," and 3) "no common language and shared concepts." One example of "misconceptions about the medium" is that the participants do not know what is possible with the AR/VR devices, what interactions can be used, and what the software is capable of. One AR designer from the study explains that because there are no standardized patterns, they download multiple applications to observe interactions with the goal of finding interactions that can also work in their own application. The participants also describe how they put a lot of effort into creating artifacts or prototypes and, therefore, do not want to discard them at a later state. In several cases, they delivered artifacts developed as prototypes as the final product, making maintenance, code readability, and code reusability problematic. For the three identified challenges, Krauß et al. [Kra+21] summarized how the participants' interdisciplinary teams try to handle them. To approach the first challenge, the teams try to create awareness for the hardware with demonstrations and experience sessions. For the second challenge, the participants and their colleagues teach their tools to each other, do joint sessions, or create physical prototypes where they mimic interactions and positions to understand their applications overall. Joint sessions are also used to approach the third challenge, e.g., in live coding sessions.

MacIntyre et al. [Mac+04a] focused on challenges designers faced when creating AR applications. The authors described that there are no simple environments, and most tools require using programming languages. Additionally, the multitude of technologies is difficult to understand and work with. Finally, it is a challenge to manage the relationship between virtual and physical elements and to test these in the real world. MacIntyre et al. [Mac+04a] addressed these challenges with their authoring tool called 'Designer's Augmented Reality Toolkit' (DART). For example, DART allows designers to capture and replay videos, including sensor data, to work on their location-specific AR applications while off-site. DART is built on an established multimedia development tool (Macromedia Director), and the authors highlighted that it is important to integrate research tools within commercial tools.

Ten years later, Gandy et al. [GM14] interviewed users of DART to reflect on how they used the authoring tool and what challenges they faced. Gandy et al. [GM14] stated that, instead of the small templates provided by DART, the users would have preferred more sophisticated examples. This especially applied to persons who were not familiar with AR.

Nebeling and Speicher [NS18] reviewed existing AR and VR authoring tools and grouped them into five classes. Based on these, they identified three main challenges: 1) The variety of available and required tools is massive, 2) authors need to use tools from different classes, and 3) tools from different classes do not work well with each other.

The challenges pointed out in the related works can be summarized as follows. 1) Authors often spend a lot of time and effort getting to know and working with a tool before knowing if it is the right tool [NS18; Ash+20] and the right way [Kra+21]. 2) There are no or insufficient examples provided to the authors [GM14; Kra+21]. Instead, authors have to download and view other apps to get an impression of what options they might have [Ash+20; Kra+21]. 3) It is difficult to understand the connection between real and virtual worlds [Mac+04a], especially when not working with an immersive device.

2.6.2 Authoring Tools

We can distinguish between programming toolkits that target programming experts and content design tools that target authors who have no or little programming knowledge [Ham+06]. Several programming tools to author AR applications exist, but their usage is up to experts because programming knowledge is necessary [Jee+14]. A popular example of such a tool is ARToolKit, which is based on C/C++. The product family of ARToolKit includes several libraries to target desktop, web application, and mobile application development. Other AR libraries or frameworks are MRTK, DWARF [BK], or ARTag [Fia05]. In contrast, content design tools allow authors to develop AR applications without coding but rather with a graphical or tangible user interface. They help authors focus on the application instead of low-level functionalities [Ham+06]. Because these tools simplify the authoring process, they leverage the widespread adoption of AR and target a larger amount of users than programming tools [Rob+16]. Examples of content design authoring tools with a graphical UI are DART [Mac+03] and AMIRE [Dör+03]. Content design tools featuring a tangible UI and an immersive approach for the authoring process are, e.g., ARtalet [Ha+10], MagicCup [Kat+03] and an approach described by Lee et al. [Lee+04]. These or comparable content design tools fail when authors aim to create more complex AR applications [WTS10]. They limit authors if they, e.g., choose to change computer vision aspects [Ham+06]. In most cases, they are limited to augmenting a real marker or a user-defined place in the real world with a virtual image or model.

Authoring tools can additionally be categorized into platform-independent or -specific tools and stand-alone or plug-in software, as concluded by Roberto et al. [Rob+16]. The authors compared authoring paradigms and distribution strategies with each other to find advantages and disadvantages for each. Based on their findings, they classified existing authoring tools and elaborated four dataflow models. Nebeling et al. [NS18] distinguished authoring tools further into five classes:

1. basic interaction design tools (e.g., Adobe XD, InVision, or Sketch),
2. tools that support basic AR/VR scenes and interactions (e.g., DART [Mac+03], ProtoAR [Neb+18], or HoloBuilder [Spe+15]),
3. tools focussing on AR/VR interactions (e.g., ARToolKit, Tiles [Iva+01], Studierstube [Sch+02], or ComposAR [SLB08]),
4. 3D content creation tools (e.g., Teddy [IMT06], Lift-Off [JK16], SketchUp, Google Blocks, Autodesk 3ds Max, or Maya),
5. 3D games and application development platforms (e.g., Unity, Unreal, or A-Frame).

These categories of and examples for AR/VR authoring tools emphasize that the authoring process often involves more than one authoring tool.

Depending on the targeted authors, authoring tools can also implement functionalities from more than one of these categories. Thus, authoring tools can target to support general purposes to create any kind of AR/VR experience (e.g., [Mac+04a]), or can focus on specific use cases, e.g., authoring tools for museum context [Ger+20; Efs+20; MT+21], education [WTS10; Den+21; Zhe+21], maintenance [Ram+13], or storytelling [Sin+21].

For example, the work by Dengel et al. [Den+22] focused on AR authoring tools for education. The authors identified 69 different AR authoring tools and classified these based on their accessibility, degree of required programming knowledge, and interactivity. Most of these are designed for experts and require programming skills, which limits their use in the classroom. The authors stated that teachers need easily accessible authoring tools with a graphical UI and interactive content. Only five of the 68 authoring tools address the needs of teachers to design educational AR experiences.

There are also authoring tools that implement novel and creative workflows and UIs. Gasques et al. [Gas+02] introduced an AR authoring tool that they call PintAR. It allows designers to sketch using a digital pen and then view their sketches in 3D using an HHD. By this, it separates the 2D sketching task (carried out using the digital pen) from tasks that require 3D interactions (carried out using an HHD). The AR/VR authoring tool 360proto [NM19] uses paper prototyping and a Wizard of Oz authoring approach. Authors create mockups of components from paper and take a picture of each. These captures are organized, and interactions are defined within the authoring tool. Hence, no programming knowledge is required. To test the authored AR/VR experiences, one person acts as a computer by reacting to the user interactions and moving the mockups while another person films this and live streams the video to the AR/VR device.

In the following subsections, we identify categories of authoring approaches and give examples for each.

Immersive Authoring Tools

Immersive AR authoring tools provide an AR environment where authors can create content or specify behavior, while immersive VR authoring tools provide a VR environment for authoring tasks. The use of 3D manipulation to construct a virtual scene is a more natural way than using traditional 2D input with mouse and keyboard [But+92; Min95]. Hence, some authoring tools for immersive 3D modeling were developed [But+92; WS01] and there are currently a number of commercial available VR 3D modeling and painting tools (e.g., Google Tilt Brush, Google Blocks, Adobe Substance 3D Painter, and Adobe Medium). Immersive authoring tools for AR and VR can also support modifying the position, rotation, and scale of virtual objects in the physical or virtual world, e.g., [Min95].

Besides the virtual objects' placement, their behavior also plays an important role in AR and also in VR applications. Steed and Slater [SS96] presented a system that allows manipulating the virtual objects' behaviors while immersed in VR. Lee et al. [LKP02] described that evaluating VR systems is tedious because developers need to switch from their development environment to the VR environment in order to evaluate their VR system. The authors stated that while there are authoring tools that allow the immersive placing of virtual content, modeling the VR system's behavior is still done using traditional 2D interfaces and programming environments. Here, their work introduces 'PiP (Programming virtual object behavior in virtual Program)', a system with immersive interfaces that allows the modeling of object behavior in VR. In a user study, the authors showed that their system saves time because developers do not need to switch environments.

Lee et al. [Lee+04] also applied immersive authoring techniques to tangible AR applications. The authors conducted a user study to assess their authoring methodology among non-programming persons. The study's findings demonstrated that all participants were capable of generating an AR scene using the authoring process, implying that the process is appropriate for users without programming expertise. The majority of participants, comprising 42% of the study group, expressed a preference for the immersive interface as opposed to a non-immersive interface. However, a notable proportion of users, constituting one-third of the study group, expressed a preference for a hybrid interface that combines both immersive and non-immersive elements. This interface would allow users to switch between conventional input methods such as keyboard and mouse, and tangible augmented reality input.

In follow-up work, Lee et al. [LKB05; LK09] described the concept of immersive authoring as similar to What-You-See-Is-What-You-Get (WYSIWYG). WYSIWYG authoring displays text, graphics, or other elements on a screen exactly as they will appear in the result after completing the authoring process. Based on WYSIWYG, Lee et al. [LKB05; LK09] coined the term "What-You-Experience-Is-What-You-Get" (WYXIWYG) to describe how users can immersively experience their AR applications while authoring it in the same environment.

Furthermore, Lee et al. [LK09] proposed design principles for immersive AR authoring tools and implemented one immersive authoring tool based on these. In a user study, the authors demonstrated that their immersive authoring tool is significantly more efficient in specifying behaviors and spatial relations. However, they also found that their immersive authoring tool has weaknesses in supporting authors for abstract tasks like logical programming.

Another immersive authoring approach is distinguishing between an editor and an experience mode or interface. Wang et al. [WTS10] developed an AR authoring tool featuring a graphical UI to facilitate the creation of AR-enhanced digital examination applications by educators and learners who do not have programming expertise. Additionally, they implemented an AR viewing interface that can be used to view the authored applications. The authors made a clear differentiation between authoring and viewing modes. 3D models or text files created in the authoring tool can be exported and then imported into the viewing interface or vice versa. This facilitated efficient and quick alteration of both the questions for the examination and 3D models. However, their authoring tool was only suitable for creating exam-related AR applications.

Similar distinctions were made between editor and viewer applications by the content design framework SimpleAR [AYPVCD42]. The two applications interact with each other using a Firebase real-time database. The editor software is based on various components, for example, an "augment a marker" component. Authors can add 3D models or other assets using the editor tool on a web development platform as well as choose and adjust the various components. In a user study, the work showed that all participants were capable of accomplishing the designated task and successfully developing an AR application within a time frame of less than five minutes. Nonetheless, the findings of the study indicated that the users encountered difficulties in comprehending the underlying concept of the various components.

Component-Based Authoring

According to the Object Management Group (OMG) Unified Modeling Language Specification, a component "represents a modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces." [Obj]. Similarly, Pressman et al. described a component as a "modular building block for computer software" [PM20]. These properties allow components to be independent of each other and to be reused in different software systems. This makes components attractive to use for AR authoring. For example, the authoring tools AMIRE [Dör+03], Studierstube [Sch+02], and a tool by Jee et al. [Jee+14] use components that provide basic system functionalities like rendering, tracking, or viewing.

Lee et al. developed a "component based application model" [Lee+04] that uses an immersive and tangible authoring approach to create tangible AR applications (see also Section 2.6.2). They defined a tangible AR application with multiple components and connections between their properties. Their application model implements three component types, where each component has properties that represent its state. First, physical objects have properties regarding their current tracking state, e.g., if they are visible or their current position. Next, virtual objects have similar properties to physical objects, but their properties can be modified by the AR application. The third component type is a logic box that can perform calculations or logical operations. The components can be linked with each other to create an AR application. For example, a virtual object can be connected and anchored to a physical one by linking the transformation properties. Authors can use this application model to create their AR applications by creating, destroying, modifying, and linking components.

Kotis proposed his research idea ARTIST [Kot19] with the goal of providing code-free methods to create reusable and optimized AR experiences. ARTIST uses components based on a novel concept of an "Experience as a Trajectory" (EaaT). An EaaT maps representations of the movement of entities, augmented with annotations, to AR experiences. Another goal of ARTIST is to monitor and analyze user behavior to optimize the AR experiences for the user in real-time. While Kotis [Kot19] described how ARTIST could be implemented, ARTIST is currently still a research idea and not implemented or evaluated.

Speicher et al. [Spe+15] implemented holobuilder, targeting to support layperson authors in creating AR-based instruction applications for industrial contexts. The authoring tool implements an editor and a player mode as components. It builds on core principles from Microsoft PowerPoint. Creating 2D slides is a familiar task for the targeted authors. Using holobuilder, authors can create "3D projects", similar to creating 2D slides using Microsoft PowerPoint. Holobuilder was turned into a commercial authoring tool.

There are more commercial authoring tools that aim to simplify the AR authoring process by using components or patterns. For example, blippAR, Microsoft Power Apps, room, XRTY App, and Zapworks provide tracking patterns like placing a 3D model or video on an image, marker, surface, or face. These tools offer a restricted range of functionalities yet allow the creation of AR applications without requiring programming knowledge.

However, the process of creating an executable AR application using any of the described tools and commercial tools still requires several sequential steps. In this way, authors can only preliminary test their AR applications after investing a lot of time in the authoring process. Also, the described related work uses components to encapsulate basic system functionalities or single blocks of content. For all, multiple components are required to create a single AR application. It could also be possible to design and implement higher level components that each represent a whole 3D scene, i.e., AR application. Each component could then be classified by its intention, e.g., components to compare objects or to realize a quiz. This idea is reflected in VR nuggets [HD19b] and nugget-based authoring, which we describe in the following section.

Nugget-Based Authoring

Horst et al. introduced a total of four authoring tools that support the creation of pattern-based VR experiences called VR nuggets [HD19c; Hor+20; Hor+22a]. Horst and Dörner [HD19c] divided their nugget-based VR systems into three parts: a *VR nugget*, an optional *functional coating*, and a *VR nugget platform*. The *VR nugget* reflects one design pattern from the educational domain, for example, a compare pattern. Depending on the pattern, a VR nugget has different parameters and virtual objects. In the case of the compare pattern, the VR nugget provides two virtual objects that serve as placeholders and can be replaced during the authoring process. The VR nugget ensures that the mandatory number of virtual objects is always included so that the VR application is always executable. Additionally, the VR nugget holds spatial information about where in the room virtual objects are located. In microlearning, the self-contained learning nuggets are studied one after another. Based on this, there is always only one VR nugget active in the VR system. The optional *functional coatings* in Horst and Dörner's VR system [HD19c] can extend VR nuggets by adding supplemental functionalities. However, they do not change the VR nugget's learning goal or utility. For example, a *functional coating* can enable end-users to grab and rotate a virtual object or allow an educator to guide a learner by highlighting specific virtual elements. Multiple *functional coatings* may be applied to one VR nugget. The VR nuggets are executed in a runtime environment that Horst and Dörner call *VR nugget platform*. It holds the VR nuggets' placeholder 3D objects and other assets and stores information about the VR nuggets. Additionally, it manages which VR nugget is active and handles aspects like memory management, data processing, and programming interfaces.

To make the VR nuggets and functional coatings accessible to persons without programming knowledge, Horst and Dörner introduced the authoring tool 'VR Forge' [HD19c]. Its UI is inspired by slideshow presentation software. Each VR nugget is represented by one slide in a timeline and the VR nuggets are connected in a linear, chronological order. Using VR Forge, authors can select a VR nugget from the timeline, adapt it, and view the changes live.

Horst et al. [Hor+20] also introduced 'IN Tiles (Immersive Nugget Tiles'), an immersive authoring tool to create pattern-based VR experiences called VR nuggets. IN Tiles represents VR nuggets with four different tile-like virtual 3D objects: A hexagonal tile frame represents the VR nugget's pattern type. The frame is filled by a hexagonal tile-shaped 3D object representing the virtual objects from the VR nugget. Optional coating tiles can be attached to each of the tile's sides. They represent the VR nugget's functional coatings. Finally, a pattern-function tile controls pattern-specific functionalities. The tile shapes offer affordances to the authors using IN Tiles. The hexagonal tile frame offers the affordance to fill it with the hexagonal tile, similar to putting a puzzle together. IN Tiles implements three VR rooms. The VR authoring room is used to assemble the tiles. Each VR nugget can be edited in a separate VR editing room, for example, to adjust 3D positions. Finally, the authors can test each VR nugget in a VR demo room. Authors can switch between these rooms as they like.

Both VR nugget authoring tools, VR Forge and IN Tiles were evaluated in user studies with laypersons in VR authoring [HD19c; Hor+20]. For VR Forge, participants found that similar educational concepts and implementations built the base for the VR nuggets and explained that this was beneficial for learning. With IN Tiles, participants liked the non-linear workflow and that they were able to switch between authoring and testing from an end-user's point of view. However, the participants perceived certain actions, like text input, as tedious in VR. Other interactions, like placing 3D content, were perceived positively to be carried out immersively in VR. Horst et al. [Hor+20] stated that this suggests exploring authoring approaches that combine traditional and immersive tools. Additionally, Host et al. [Hor+20] compared the VR nugget authoring tools IN Tiles [Hor+20] and VR Forge [HD19c]. IN Tiles and VR Forge were rated similarly on the AttrakDiff [HBK03] questionnaire. Both VR nugget authoring tools were perceived positively regarding their hedonic and pragmatic quality.

Two years later, Horst et al. introduced two further VR nugget authoring tools, called 'CoNMoD' and 'ViNS Tiles' [Hor+22a]. Using CoNMoD (Context-Related Nugget Modules with Direct Content Interaction), authors can adapt VR nuggets using context-related direct interactions. It is based on modules that reflect the VR nuggets' parameters. Depending on which module is selected, different authoring interactions are available. Authors can view their changes to the VR nuggets from a VR HHD. ViNS Tiles (Visual Nugget Scripting Tiles) transfers the authoring approach with tiles from IN Tiles [Hor+20] to a setup with a desktop computer.

Both VR nugget authoring tools, CoNMoD and ViNS Tiles, were rated positively by participants in two separate user studies. Some deviations in single aspects of the four authoring tools VR Forge, IN Tiles, CoNMoD, and ViNS Tiles suggest that participants have individual preferences for these aspects. For example, CoNMoD and VR Forge were rated higher in their pragmatic quality than ViNS Tiles and IN Tiles, and vice-versa regarding the hedonic quality. Therefore, authors who prefer a pragmatic approach might prefer to work with CoNMoD or VR Forge, while hedonic-motivated authors might want to work with ViNS Tiles or IN Tiles. Therefore, Horst et al. [Hor+20] introduced an exchange format for VR nuggets that allows authors to exchange VR nuggets with each other. Each VR nugget is saved in one archive file that includes the 3D models, images, a thumbnail image that serves as a preview for the authors, and a JSON file that stores semantic relations and positions of the VR nugget.

The four VR nugget authoring tools and their evaluations show that nugget-based authoring can support persons without experience in VR or programming in creating their own, small VR experiences. Although it was shown that the concept of nuggets can be transferred to effective VR authoring ([HD19c; Hor+20; Hor+22b; Hor+22a]), nuggets have not yet been transferred to AR authoring. VR nuggets do not implement functionalities to register the user's physical environment in 3D or to detect and track physical images or tangible objects. Therefore, combining real-world with virtual elements cannot be targeted with VR nuggets. To create AR applications, authors need to be able to define where in the real world virtual elements are anchored. For example, an author could anchor a virtual 3D object to a specific point of interest in a room or to a tangible object that the user can grab and move. New authoring paradigms for nuggets are needed that not only allow authors to do so but also support authors in choosing suitable anchors in their environments.

Moreover, AR applications need to be able to adjust to changes in the users' physical environment. For location-specific AR applications, the environment is known. However, there can still be unexpected changes, e.g., a person standing on the point where virtual elements should appear and blocking the view. VR nuggets do not continuously scan and register the environment, so they cannot deal with such challenges. Also, for non-location-specific AR applications, authors may want to define where virtual elements are placed, e.g., on a desk or a wall. Again, VR nuggets do not support authors with this challenge.

Besides supporting authors, supporting AR application users also includes challenges that VR nuggets cannot target. For instance, the interaction paradigms in VR and AR are distinct. VR is often used with a fully immersive HHD, and users can interact with the virtuality using controllers. In AR, it is common to use either HHD with interactions similar to using a smartphone or HHDs with support for gestures and hand tracking. These differences in interaction models require different design considerations and approaches for authoring content in VR and AR. Different interactions come with different challenges for the user interacting with the applications. VR nuggets cannot support users in AR if they encounter challenges. For example, if the lighting conditions are too dark to register the user's environment, VR nuggets cannot detect why the environment is not registered and cannot support the user in finding a solution.

Further challenges in AR that cannot be targeted using VR nuggets are related to the nuggets' underlying patterns. Not all patterns from VR nuggets might be suitable or useful to be applied to AR. Similarly, there could be patterns that are useful for AR but have not been introduced for VR nuggets or that may not be suitable for VR nuggets. For instance, However, also patterns that can be applied to both, AR and VR, need to be transferred and adapted in their implementation for AR. For example, a VR nugget that annotates a virtual object with labels has its labels attached to the virtual object. In AR, the virtual labels could similarly be attached to a virtual object, but it could also be useful to be able to attach virtual labels to a physical object.

Game Engines

Game Engines provide authoring environments to develop interactive virtual environments, e.g., computer games. They can control the gameplay, render the game's graphics, and provide the physics engines and simulation tools for achieving realistic behavior and interaction [MJ02]. Examples of common game engines are Unity, Unreal Engine, Cryengine, or Frostbite Engine.

Current game engines also provide functionalities that support the creation of AR and VR experiences. For example, they support many different devices and target platforms: smartphones and tablets operating on Android as well as on iOs, AR HHDs based on Windows (HoloLens 2) or other operating systems, and numerous VR HHDs operating on different systems. Furthermore, there are multiple toolkits and SDKs available that offer AR- or VR-specific functionalities. For example, Microsoft's MRTK implements the functionality to anchor virtual objects in a physical room with so-called spatial anchors. Moreover, it provides functions and virtual UI elements for interactions. With a scene understanding SDK, Microsoft also supports detecting and categorizing surfaces in the users' physical environment by type, e.g., wall or floor. The tracking toolkit Vuforia implements image, object, and area detection and tracking. All of the named toolkits and SDKs are available as plugins for the game engine Unity. Additionally, game engines have built-in support for spatial sound [Spe+18].

However, to utilize game engines for the development of AR/VR experiences, programming and AR/VR knowledge are required [NS18]. Game engines do not target the challenges pointed out in Subsection 2.6.1. There are approaches to lower the barriers of using game engines for non-programmers by utilizing visual scripting [Sew15; Ber18]. Visual scripting refers to a programming paradigm that utilizes graphical elements and symbols to represent code logic and control flow. By arranging and connecting the elements, authors can manipulate the virtual objects and their behaviors. Nonetheless, authors need to have experience with AR/VR to create an AR/VR experience using visual scripting and game engines. Additionally, this can become complex when interactions or mathematical transformations from computer graphics are involved.

Game engines can also be used to create other novel authoring tools. As game engines have become a "standard for AR/VR development" [NS18], it is also common to use them to create AR/VR authoring tools. This is also underlined by the number of scientific works using game engines to create AR/VR authoring tools. Several of the authoring tools described above and more were developed using game engines, e.g., [Spe+18; AYPVCD42; dTC19; Gas+02; Ger+20; Efs+20; Kİ20; MT+21; TLR22] were all developed based on Unity.

2.7 Discussion

The related work presented and discussed in this chapter shows that several challenges exist in using and authoring AR applications. This section summarizes the related work and points out where research gaps remain. We see these research gaps as a motivation to develop and research novel ideas. Based on these, we formulated RQs (Section 1.2) and our motivation (Section 1.1). The following paragraphs map the research gaps identified in this chapter to our RQs.

RQ1: How can a nugget concept be applied to AR?

The concept of nuggets from microlearning was transferred to other domains, e.g., to VR [HD19b]. Although there are similarities between AR and VR, there are additional challenges in AR that VR nuggets cannot target. This is mainly because virtual elements in AR are anchored in and connected with the real world. For example, VR nuggets cannot support authors to find suitable points in the real world to anchor their virtual elements. Additionally, there could be patterns from VR nuggets that cannot be transferred to AR, and there could be new patterns for AR that are not suitable or not identified yet for VR nuggets. Finding a concept of how to apply nuggets to AR is a research gap that this work targets (RQ1a). This also includes finding universal patterns for AR applications that can serve as a basis for AR nuggets (RQ1b).

RQ2: What features are included in AR nuggets?

In VR, it is possible to control the whole environment of the user. In contrast, authors of AR applications cannot control the user's physical environment. Thus, nuggets for AR may need to be able to guide users from one AR experience or PoI to another one (RQ2a).

Additionally, interactions in VR differ from interaction paradigms in AR. Typically, VR applications can be experienced using an Head-Mounted Device (HMD), and users interact with the application using controllers or their hands. In contrast, AR is common for HHD as well as HMD. For AR applications on HHD, interactions can include taping or gestures on a touch screen. With HMD, gestures in the air or eye tracking can be used to interact with an AR application. Nuggets for AR need to consider the more extensive range of device options. Because such interactions can be challenging for the users, they may need support to use the AR applications [HD19a]. VR nuggets, targeting different interactions, cannot provide this support. Some AR applications include tutorials (e.g., [Gra+09; GMF10; FGF11; MGF11; Pon+11; Kim+14]. However, then it can still be challenging for users to 1) realize that they need support, 2) find the correct tutorial, and 3) memorize what they learned in the tutorial. Some AR applications target to approach these challenges by providing on-demand support, e.g., [WLF07; PRD09; WCL19]. The support functions implemented by these are specific to their use case and do not apply to AR in general. For example, Wu et al. [WCL19] support users by providing information about how to improve scanning quality when 3D scanning a physical object. This cannot be applied to challenges in general AR functionalities like tracking or other interactions. Therefore, it remains open if and how AR nuggets can support users in these general AR interactions (RQ2c).

Furthermore, one idea of VR nuggets is that they can be integrated with other media, e.g., a lesson can include traditional slides and some additional VR nuggets on specific topics where they add value. Thus, the question arises if VR nuggets can also be combined with nuggets in AR (RQ2b).

RQ3: How can AR nuggets address location-specific content and tangible interactions?

One main challenge that nuggets in AR need to target is how to gather, incorporate, and use information about the users' physical environment and surroundings. AR can connect to the real world using tangibles, which serve as UI, and users can manipulate them to control virtual objects [BKP08]. Because using tangible objects can give users a more natural feeling than freehand interactions [LRS10; DLB15; Ssi+19], it can be useful to integrate tangibles in nuggets for AR. However, there is a research gap if and how nuggets in AR can include tangibles (RQ3b) and what tangibles would be suitable (RQ3a).

Furthermore, AR can connect to the real world with location-specific virtual elements. To realize this, virtual elements in AR must be anchored at specific points in the real world. Because VR nuggets implement solely virtual elements without connection to the real world, they cannot solve this challenge. Thus, there is a research gap in how nuggets in AR can anchor virtual elements in the real world (RQ3c).

RQ4: How can AR nuggets support authors to develop qualitative valuable AR applications?

Authoring AR applications involves several challenges [Mac+04a; GM14; NS18; Ash+20; Kra+21]. For example, it is challenging to find the right authoring tool to realize one's idea [NS18]. Moreover, component-based authoring tools require authors to connect multiple components with each other to create a single AR application. Thus, authors spend time working with these authoring tools before realizing if it suits their ideas [NS18]. Current authoring tools do not provide components that represent whole 3D scenes or applications. VR nugget-based authoring [HD19c; Hor+22a] uses such components that represent whole VR scenes, but VR nugget authoring tools do not target AR-specific challenges. Another authoring challenge is to connect virtual elements with the real world. Immersive authoring tools target this by immersing authors in the same environment as the user when experiencing the AR application. However, immersive authoring can be perceived as tedious [LKP02; LK09] for some authoring tasks. Here, it could be helpful to have an authoring tool that supports immersive and non-immersive authoring [Lee+04]. While there are authoring tools that distinguish an immersive viewing mode from an editor mode (e.g., [WTS10; AYPVCD42]), these do not necessarily provide an immersive authoring mode. How nuggets in AR can support authors in a workflow that targets these challenges is a research gap (RQ4a). Also, it remains open how tools that support authors using AR nuggets can be conceived and implemented (RQ4b).

Finally, nuggets need to be delivered and executed on an AR device. Exchange formats existing for virtual elements and VR nuggets [Hor+20], but they cannot include real-world information. How and if AR nuggets can be exchanged, delivered, and executed remains open (RQ4c).

Chapter 3

Concepts for AR Nuggets

This chapter introduces a concept for using pattern-based components that we call AR nuggets. We illustrate our concept by identifying suitable patterns and describing examples. Hereby, we introduce concepts for tangible interactions, user assistance, and the combination of multiple AR nuggets with each other or with VR nuggets. We target how AR nuggets can support the authoring process of AR applications in Chapter 5.

3.1 Definition

In this section, we define AR nuggets and the roles of actors involved in creating and using AR nuggets.

3.1.1 AR Nuggets

Our definition for AR nuggets derives from VR nuggets [HD19b], which derive from the educational approach of microlearning. Because VR nuggets target challenges related to the authoring of VR applications, there are challenges in using and authoring AR applications that VR nuggets cannot address. We described these in Section 2.6.2. With regard to these additional challenges, as well as the challenges described in Section 1.1 and our two example scenarios, we define AR nuggets as follows.

AR nuggets are ready-to-use, stand-alone, and self-contained AR applications based on patterns. Authors can experience the ready-to-use AR nuggets without having to create or adapt something before. This way, authors can choose the AR nugget that fits their needs most. Furthermore, a pool of AR nuggets can serve as inspiration in the form of example applications and support authors in gathering ideas of what applications are possible with AR.

As stand-alone applications, they include their runtime environment so that users do not have to install additional applications to be able to experience an AR nugget. From an author's point of view, the AR nuggets are independent of each other because they are self-contained. By this, the author can change, remove, or replace an AR nugget without affecting other AR nuggets. Additionally, the AR nugget implements all functionalities and user interactions it requires.

Each AR nugget implements one pattern we identified from recurring scenarios in AR applications. We refer to the underlying pattern of an AR nugget as its nugget type. One pattern is implemented by one AR nugget, while the AR nugget can have multiple variants. Because AR nuggets are self-contained, the patterns similarly do not rely on external factors or other patterns. One example of such a pattern is to label a virtual object anchored to a physical tangible as illustrated in Figure 3.1. Patterns, e.g., comparing two objects or doing a quiz, can be familiar to domain experts from other media types and thus easily be understood and used [Kli+02].

AR nuggets are composed of placeholder objects, parameters with default values, default real-world anchors, and interactions. We elaborate on these in the following.

- Virtual elements that are mandatory for the AR nugget's pattern are provided as placeholder objects. For example, these can be basic geometric shapes. Authors can replace these to adapt the AR nugget to their own needs.
- The placeholder objects have parameters with default values, e.g., their size and position in relation to their anchor in the real world.
- A real-world anchor is a reference to one specific part in the real world to which virtual content can be attached, e.g., a static point in the physical environment, a defined (trackable) image, or a defined (trackable) tangible object. By default, AR nuggets use printed images that AR devices can detect and track as real-world anchors.
- All AR nuggets allow users to move freely in their environment while the augmentations remain in the correct position. Additionally, some AR nuggets include further interactions. For example, a quiz AR nugget supports user input so that the user can answer the question.

The ready-to-use AR nuggets are composed of default parameters and placeholder objects. Therefore, we call them *default* AR nuggets. Using AR nugget authoring tools, authors can adapt the default AR nuggets without programming. An AR nugget authoring tool is software that allows one to replace placeholder objects in AR, change the parameters' default values, or add further functionalities to the AR nugget. Once the default AR nugget has been altered, we refer to it as an *adapted* AR nugget. Similarly, Horst et al. distinguish between default and adapted VR nuggets [Hor+22a]. Authoring tools for AR nuggets can be stand-alone tools but can also be integrated into other software. Therefore, it is possible to use more than one AR nugget authoring tool to adapt an AR nugget.

3.1.2 Involved Actors for Creating and Using AR Nuggets

Authoring and using AR nuggets incorporates multiple roles. These do not exclude each other. One person can hold one or more roles, and one role can be held by more than one person.

- AR nugget author: To create an AR nugget in the first place, the AR nugget author identifies an underlying pattern and conceptualizes the AR nugget by identifying mandatory

Figure 3.1: Top: Example of a default AR nugget. Bottom: Example of an adapted AR nugget. Both are executed on a handheld device. The default AR nugget shows a cube with two labels anchored to an image target. The adapted AR nugget was adapted based on the default AR nugget and shows a cross-section of skin with labels anchored to a 3D print of the same cross-section. [Rau+21]

placeholder objects, finding suitable default values for the parameters, and identifying appropriate interactions.

- Developer: The developer implements an AR nugget. AR nugget author and developer work closely together to test an AR nugget. When the AR nugget author and developer are satisfied with the newly developed AR nugget, the developer provides it to the pool of existing AR nuggets.
- Author: Authors use the provided default AR nuggets and adapt them to their own needs. To do so, authors do not need to have programming knowledge. Instead, authors from other domains than programming can use and adapt the AR nuggets to implement their domain-specific knowledge in AR applications.
- User: The end-user experiences the AR nuggets, e.g., in an educational context as a learner or in a museum as a visitor.

3.2 Examples

This thesis applies the concept of AR nuggets to the educational domain in the context of continuing medical education and museum exhibitions.

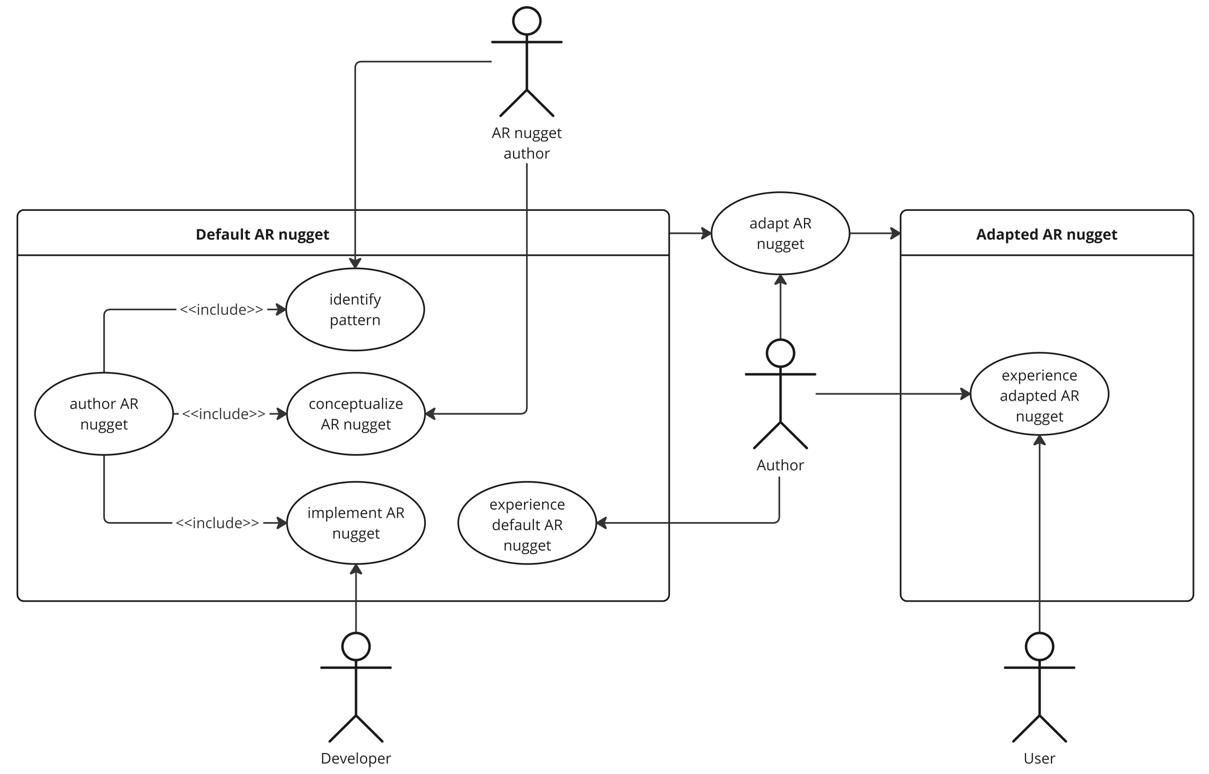


Figure 3.2: Use case diagram of involved actors for creating and using AR nuggets.

In these domains, we identified recurring scenarios where AR can add value. From these, we derive patterns that serve as a basis for creating AR nuggets. Besides these two educational domains, the patterns aim to be universal so that the resulting AR nuggets can also be applied to other domains. We used three different approaches, which we describe in the following. To this end, we gather a total of eight patterns with different variations.

First, we analyzed patterns that Horst and Dörner [HD19a; HD19b] identified to create VR nuggets. For each of those patterns, we investigated if the pattern or a similar one is also applicable to AR and what alternations are necessary to apply the pattern to AR.

Second, we identified patterns with discussions in meetings with experts from the industry. These meetings were held regularly over a period of three years.

Our third approach to identifying patterns is by conducting expert interviews. We planned and executed one expert interview in a qualitative semi-structured form. Based on [RM16], we structure the interview into an introduction, warm-up, main session, cool-off, and closure. In the introduction, we introduced ourselves and the goal of the interview. Next, in the warm-up session, we introduced the interviewees to AR and tangible AR. In the main session, we asked questions we developed based on guidelines from work by Robson and McCartan [RM16] and Preece et al. [PRS15]. In the cool-off session, we summarized the interview and asked if there was anything else that we had not talked about yet. Finally, in the closure session, we thanked the interviewee and said goodbye.

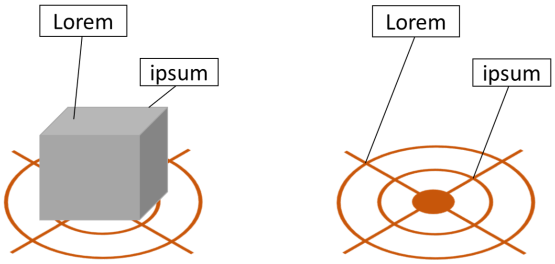
We categorized results using an affinity diagram as described in work by Beyer and Holtzblatt [BH99]. Based on the diagram, we identified our patterns. To document the identified patterns, we use a structured form based on Horst et al. [Hor+21c]. That work describes each pattern with a name, concern, illustration, example image, when to use it, visuals, interactions, example scenario, and what is needed to adapt the nugget. Due to the self-contained and stand-alone properties of AR nuggets, we also make each pattern description self-contained, i.e., without references to other patterns. We make four adaptions and extensions to the structured form by Horst [Hor+21c]. First, we add pattern variants to the description scheme. One example of a variation is whether a physical object is, additionally to other virtual elements, augmented with a 3D model. For example, a tangible can either be labeled without an additional virtual 3D model augmenting it (so that the labels directly connect to the tangible) or with one (so that the labels seem to connect to the virtual augmentation). Second, when we describe the interactions, we concentrate on pattern-specific interactions. In general, all patterns allow the users to move in their environment, and the augmentations remain stable in their correct position. This applies similarly if a tangible is involved. A physical tangible can be grabbed, moved, or rotated. Virtual elements anchored to the tangible move accordingly with the tangible and keep their relative position and orientation to the tangible. Because these interactions apply to all patterns, we do not list them specifically in the documentation below. Third, instead of the element "what is needed," we separate between "what virtual elements are needed" and "what requirements to the physical world apply." Because this form was designed for VR, it includes no information about connections to the real world. However, this is crucial for AR nuggets. "What virtual elements are needed" describes virtual assets like 3D models, sounds, or text information. Requirements to the physical world can, e.g., be a specific anchor point in the real world for a location-specific AR nugget or a tangible that virtual content is anchored to. Fourth, we add the "configurable parameters" element to the structured form to describe different options that authors can configure.

Using this structured form, we list exemplary patterns we identified that can serve as a basis for AR nuggets in the following. Our illustrations include targets (orange color), which are PoIs or tangibles in the physical world where virtual elements can be anchored. Each target is augmented with cubes, text boxes, or other virtual objects that serve as placeholders for custom virtual objects.

Name: Show & Tell

Variants: a) including a virtual 3D object, b) labels attached to the physical object

1. Concern: This pattern labels a real object or PoI with text or images and optionally augments it with a virtual object.
2. Illustration: a) and c) a) and d)



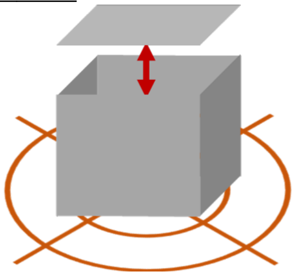
3. Example Image: a) and c) b) and c)

4. When to use: This pattern can be used to present what a physical or virtual object is composed of.
5. Visuals: Virtual text labels or images are connected to a physical object, which can be a tangible or a point in the real world. In variant a), the physical object is augmented with a virtual object, while in variant b), the label lines are directly connected to the physical object.
6. Pattern-specific interactions: With variant e), labels can show more information if they are selected or if the user moves closely to the labels.
7. Example scenario: One example in a setting of medical education is to label a tangible cross-section of human skin to educate about the skin's structure.
8. What virtual elements are needed: Authors can adapt the text labels, the optional virtual object, and optionally more information that is shown on demand.
9. What requirements to the physical world apply: The labels need to be anchored to one or more PoIs or to one tangible.
10. Configureable parameters: Authors can choose if the labels orient themselves towards the user to ensure good readability or keep an orientation independent from the user's point of view to encourage the user to explore the object interactively. Instead of text labels, the authors can also use images. Additionally, authors can configure whether labels are clickable and show more information on click or show more information if the user moves closely to the label.

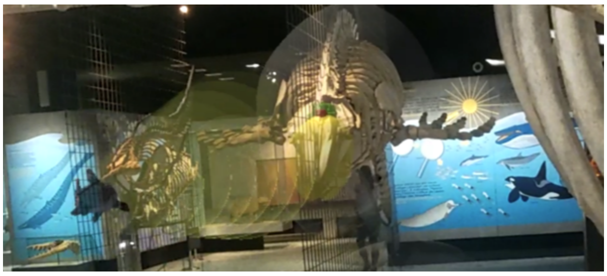
Name: Progression

Variants: a) pre-defined speed, b) user controls speed

1. Concern: This pattern is related to the visualization of a process.
2. Illustration: a)



3. Example Image: a)



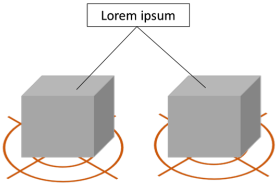
4. When to use: The pattern can be used to show how a real-world object changes during a process or over a period of time.
5. Visuals: A tangible object or real-world point of interest is augmented with animated virtual elements.
6. Pattern-specific interactions: In variant a), the animation's speed is pre-defined by the author. In variant b), a slider allows users to control the animation's speed so that they can view the animation in slow motion or as a time-lapse.
7. Example scenario: In the context of medical education, an animation can show a disease's progression and how it affects a body organ over time.
8. What virtual elements are needed: An animated virtual 3D model is required. One animation is sufficient as it can be played as a loop.
9. What requirements to the physical world apply: The virtual 3D model needs to be anchored to a PoI in the world or to a tangible.
10. Configureable parameters: Authors can configure if there is a slider for the time-lapse speed (variant a or b) and the start speed of the animation.

Name: Compare

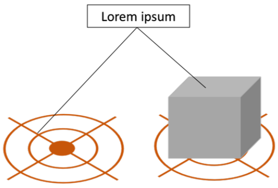
Variants: a) including a virtual 3D object, b) labels attached to the physical object

1. Concern: This pattern compares two physical objects with each other. Optionally, one or both objects can be augmented with a virtual object.

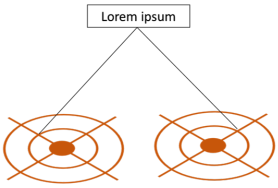
2. Illustration: c)

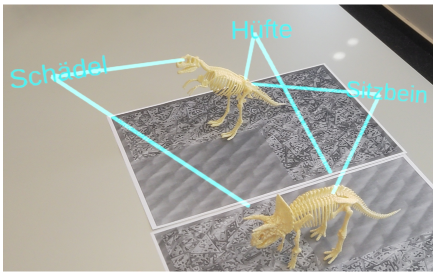
   c) and d)

   d)

3. Example Image: d)

4. When to use: This pattern can be used to point out differences and similarities between two objects.

5. Visuals: Text labels or images are connected to two virtual (variant a)) or physical objects (variant b)) with a line each.

6. Pattern-specific interactions: With variant e), the labels can be selected by the user and then show additional information.

7. Example scenario: In the context of a museum, two physical animal skeletons can be compared with labels.

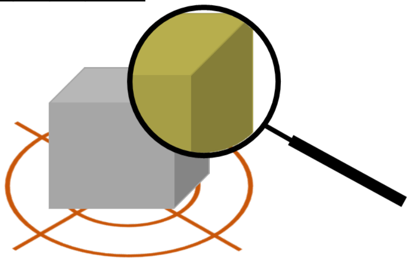
8. What virtual elements are needed: The labels' text or images and the optional virtual objects can be edited by authors.

9. What requirements to the physical world apply: The labels or the virtual 3D objects need to be anchored to tangibles or PoIs in the real world.

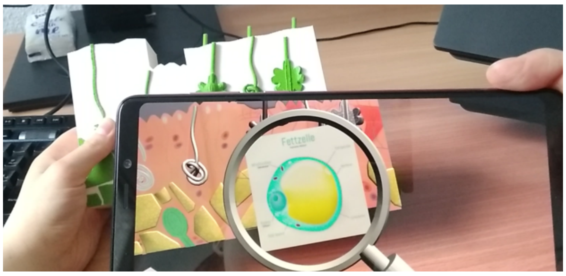
10. Configureable parameters: Authors can configure if labels show more information on click. Instead of text labels, the labels can also show images. Also, authors can configure if labels orientate themselves towards the user or keep their orientation in the room while the user moves around.

Name: Semantic Zoom

1. Concern: This pattern provides on-demand information about specific points of interest by showing an additional layer through a physical or virtual magnifying glass.
2. Illustration:



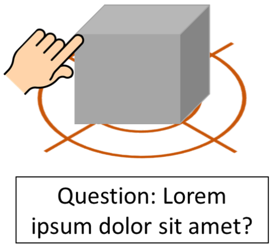
3. Example Image:



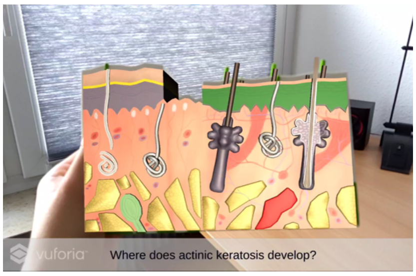
4. When to use: This pattern can be used to visualize information from different layers that are overlapping and would normally occlude each other.
5. Visuals: A physical object is augmented with a virtual object. Through a magnifying glass, additional information in the form of images, videos, or further 3D objects is shown.
6. Pattern-specific interactions: The magnifying glass can be moved and rotated by grabbing it, similar to a real magnifying glass.
7. Example scenario: One example from the domain of medical education is a cross-section of human skin as a 3D model. The 3D model is augmented with a virtual overlay of the 3D model and additional information about a fat cell, where the additional information is only visible through the magnifying glass.
8. What virtual elements are needed: This pattern can be filled with one custom 3D object and additional information (another 3D object, text, images, or videos).
9. What requirements to the physical world apply: The 3D object attaches to one anchor (tangible or PoI).
10. Configureable parameters: Optional, the magnifying glass can enlarge content, and authors can define its enlargement factor.

Name: Quiz

1. Concern: This pattern can challenge users to answer location-related questions like "Where is ...?".
2. Illustration:



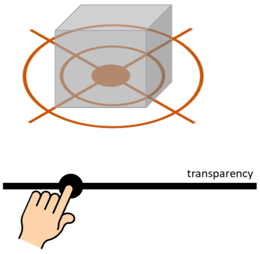
3. Example Image:



4. When to use: This pattern can be used to realize or gamify education in the form of a quiz. The quiz can ask to point to specific points on a real or virtual object.
5. Visuals: This pattern shows a quiz question in text form and a virtual 3D object. The virtual 3D object has two or more parts that can be clicked or touched. Alternatively, the quiz can include multiple 3D objects with one or more parts each as answer options.
6. Pattern-specific interactions: Users can tap on the 3D models to get feedback on whether their answer was right or wrong
7. Example scenario: In the context of medical education, a 3D model of a cross-section of human skin could be shown with the quiz question "Where does actinic keratosis develop?". If users tap on any part of the virtual cross-section, they get feedback on whether their answer was right or wrong.
8. What virtual elements are needed: For the right and the wrong answer, one or more 3D models are required each, resulting in two or more 3D models total. Alternatively, a single 3D model could be split into two or more parts.
9. What requirements to the physical world apply: For each 3D model, one anchor in the form of a PoI or tangible is required.
10. Configureable parameters: Authors can define which (parts of the) 3D model is the right answer and which one is the wrong one.

Name: Superimposition with Interactive Transparency Control

1. Concern: This pattern superimposes a real object with a virtual one. Users can interactively change the virtual object's transparency with a slider.
2. Illustration:



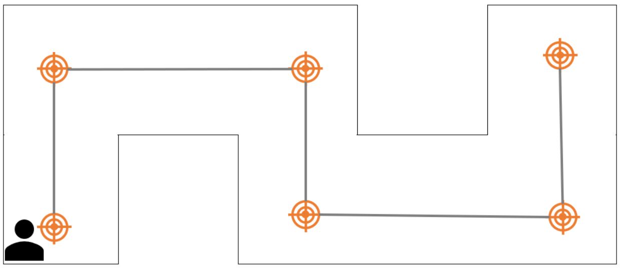
3. Example Image:

4. When to use: It can be used to support users in understanding the connection between a physical object and a virtual one because users can change the virtual object's transparency on their own.
5. Visuals: Besides the virtual 3D model, this pattern includes a slider to control transparency.
6. Pattern-specific interactions: Users can move the slider to control the level of transparency.
7. Example scenario: One example can be in a natural history museum where an animal's skin is augmented to the physical animal bones.
8. What virtual elements are needed: One or more virtual 3D models are required.
9. What requirements to the physical world apply: The virtual 3D model or models are anchored to a physical object or PoI.
10. Configureable parameters: Authors can configure the range of the transparency and the transparency's start level.

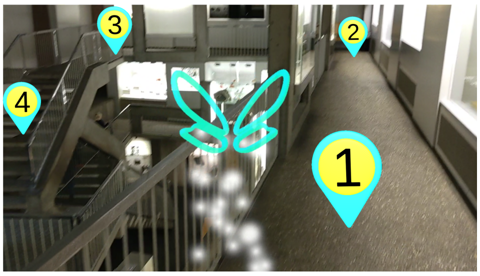
Name: Navigation

Variants: a) node-based, b) based on pre-processed scan data, c) based on spatial mapping information

1. Concern: This pattern is concerned with the navigation process from one location in the real world to another.
2. Illustration: a)



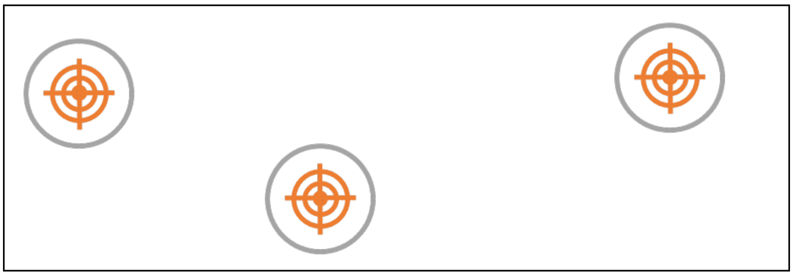
3. Example Image: a)



4. When to use: It supports users in navigating between points of interest.
5. Visuals: A virtual avatar moves in front of the user along the way until it reaches the destination.
6. Pattern-specific interactions: Users can start the navigation by clicking a start button. When the user follows the avatar, the avatar matches its speed to the user's so that the user does not need to wait or hurry. Also, the avatar waits for the user or goes back to the user if the distance between it and the user becomes large.
7. Example scenario: An example in a museum could be to navigate from one exhibition to another one in another room.
8. What virtual elements are needed: The default avatar that guides the user can be replaced with any virtual 3D model.
9. What requirements to the physical world apply: For variants a) and c), real-world anchors for the pathway and the destination are required. Variant b) requires a pre-processed scan of the environment.
10. Configureable parameters: Authors can configure the pathway that the avatar follows.

Name: Indicators

1. Concern: This pattern's concern is to indicate to users where they can find more points of interest in a room.
2. Illustration:



3. Example Image:

4. When to use: It is typically used in a large room with multiple points of interest to attract the user's attention to the points of interest.
5. Visuals: The pattern includes indicators in the form of circles or arrows.
6. Pattern-specific interactions: When users move close to an indicator, this indicator disappears to allow the user to view the object at the point of interest.
7. Example scenario: In a large exhibition room in a museum, this pattern can be applied to point users to the most interesting exhibits they should see for a 30-minute tour.
8. What virtual elements are needed: The default indicators may be replaced with custom 3D models that can serve as indicators.
9. What requirements to the physical world apply: Each indicator is anchored to a physical object or PoI.
10. Configureable parameters: Authors at which distance to a user an indicator disappears.

3.3 Utilization of Tangible Interactions in AR Nuggets

One option to interact with AR nuggets is to utilize tangible objects. With tangible objects in AR, users can grasp and move the tangibles to manipulate the virtual content accordingly. Studies showed that users prefer interacting with physical objects over using hand gestures [DLB15].

However, interactions with tangibles also involve challenges. AR nuggets can target to support users in approaching such challenges. We observed users with no or little experience in AR and identified two challenges they faced with using tangible AR applications. For these challenges, we aim to find ways of supporting users in the following subsections.

First, it is not always obvious to users that they can interact with a tangible to manipulate virtual content. Therefore, AR nuggets target to support users in realizing the connection between tangible and virtual content. This can support users in 3D interactions and provide affordances. An object has an affordance if one of its properties invites for a specific action [Gib77], e.g., a door handle invites to lever it, or a button suggests to press it. Also, manipulation performance increases if the virtual objects look similar to the tangibles they augment [KKL09]. We approach our first challenge in Subsection 3.3.1 by using 3D printing technology to receive a physical and virtual object with the exact same shape. Then, we apply our idea to the medical domain, where we design and 3D print two tangible objects. The options for intuitive affordances and increased manipulation performance are advantages of realistically shaped tangibles.

The second challenge we identify applies to AR applications on handheld devices. When users see AR through a handheld AR device, they typically have only one hand available for interactions because their second hand is needed to hold the AR device. Depending on the tangible, interactions like holding, rotating, or moving it with a single hand can be difficult. Furthermore, using two hands simultaneously for different movements requires coordination and practice. In some cases, users can occlude the tangible with their hands when rotating it, which results in a problematic tracking loss. Here, one of our goals is to create a tangible that supports one-handed user interactions in AR nuggets. We strive to create a versatile tangible suitable for multiple scenarios and offer several affordances for interactions in all types of AR nuggets. We approach this second challenge by designing and creating a generic tangible for handheld AR in Subsection 3.3.2. Our goals here are to support users in comfortably interacting with the tangible using one hand and to provide affordances for different established interactions, e.g., pressing buttons. Advantages of a generic tangible are that it can versatilely be designed to support one-handed or other specific interactions and can be used in multiple scenarios.

Finally, we combine advantages from realistically shaped and generic tangible types in a combined tangible with a plug connection system that we describe in Subsection 3.3.3. With this, we also propose a way to track tangibles that are otherwise hard to track stably.

3.3.1 Realistically Shaped Tangible

We design our realistically shaped tangibles for the application domain of CME. More specifically, we designed one model representing a cross-section of human skin and another representing two vertebras with a spinal disk as a part of a human spine. We show both tangibles in Figure 3.3 and describe them in the following.

The skin model has a cuboid shape of 12cm × 19cm × 5cm. This size is big enough for stable tracking, even if parts of the tangible are covered by a user's hand, but still small enough to naturally fit in a user's hand. On the two largest sides, the skin model shows fat cells, sebum, and

Figure 3.3: Two 3D-printed realistically shaped tangibles. Left: A cross-section of human skin. Middle and right: Two vertebras with a spinal disk.

hairs. The tangible weighs 314g, which allows comfortable handling. Because the digital model looks similar to the 3D printed object in form, size, color, and texture, it is possible to use the digital 3D model file as a reference for the tracking algorithm. The tracking algorithm can search the images from the AR device's camera stream for known 3D models and detect and track the 3D model.

The vertebras model has a size of 8cm × 6cm × 10cm and a weight of 72g. In contrast to the skin model, the vertebras model has little contrast as it is composed of only two vertebras in one color and a spinal disk in a second color. Even though it has a unique form, it has too few features for stable tracking functionality. We solve this challenge by creating and applying an image target as texture to it. Similar to our object in Subsection 3.3.2, we glue adhesive paper with this texture to the vertebras model. Then, we scan the vertebras tangible to get a textured virtual 3D model of it. This can serve as a reference for the tracking algorithm.

Creating a specific tangible for each application scenario also has its disadvantages. Most obviously, a realistically shaped tangible can only be applied to one use case, so a lot of different tangibles are needed to use multiple applications. Designing, creating, and 3D printing a tangible takes time and resources. Additionally, only some virtual 3D models can be 3D printed right away. In some cases, 3D printing specific adaptions are required. Also, with standard 3D printers, objects are 3D-printed monocolored or bicolored. With image tracking technologies, a single-colored object is harder to detect and track than a multi-colored object. Therefore, coloring or adding texture is necessary for some 3D printed objects, like our vertebras model. Also, it is important to find a suitable shape and size that users can handle comfortably. Otherwise, users may need to twist their hands or occlude the tangible with their hands as shown in Figure 3.3. Especially for handheld AR, users need to be able to handle the tangible with one hand. Rotating a tangible to view it from all sides can be challenging with one hand, depending on the tangible's shape. While this type of tangibles supports users in dealing with our first challenge, there is room for improvement to meet our second challenge.

3.3.2 Generic Tangible

In contrast to realistically shaped tangibles, a generic tangible can versatilely be applied to any AR nugget or tangible AR application. Additionally, a generic tangible can be designed to allow comfortable handling and sufficient tracking quality. In this subsection, we identify and describe requirements for such a generic tangible. Based on these, we create one generic tangible to approach our second challenge. In the following, we describe the requirements and how we approach them with our generic tangible. Figure 3.4 shows what our final generic tangible looks like.

1. Comfortably one-handed interactions like movements and rotations: When users hold an object in one hand, viewing it from all sides is difficult. For example, to view the object from its backside, users must uncomfortably rotate their hand. If they occlude the tangible while doing so, further movements can no longer be detected until it is no longer occluded. Our idea is to attach a handle to the tangible to make it easier to grasp. This allows rotating the tangible easily in all directions without occlusion through the user's hand. Therefore, our handle facilitates viewing virtual 3D content anchored to the tangible, e.g., virtual 3D models, from all sides. Our handle simultaneously works as a tripod. This allows users to place and rest the tangible on a surface, for example, on a desk, while still being able to view it from all sides, including from the bottom. Then, users can hold their handheld AR device with both hands or use one hand for interactions like tapping on the screen.

2. Comfortable size to grab: The tangible should be small enough to allow users to hold it in one hand but large enough that vision-based tracking technologies can detect and track it. According to a study conducted by Sheridan et al. [She+5], cubes should be designed to fit naturally in the user's hand. The optimal size for a tangible object depends on the size of the user's hand. Jimenez et al. [Jim+15] suggest a size of 8cm × 8cm × 8cm as appropriate. They use a webcam with a resolution of 2 megapixels and a focal length of 3.7 mm and determine that this resolution is sufficient for recognizing targets at distances of up to 1.5m between the object and camera. Additionally, they noted that this size falls within the range of sizes suggested by AR software developers. Our tangible's size is 3.5cm × 3cm × 8cm. However, our idea is that users grab the tangible by its handle. Its handle is 14cm long with a diameter of 2.5cm. This size makes it easy to rotate the tangible with one hand.

3. Lightweight: Another goal is to make the tangible lightweight to prevent users from fatigues while holding the tangible. We use 3D printing technology with light plastic material to create our tangible. Our final tangible weighs 85g.

4. Provide affordances: Generic tangibles can have multiple affordances to fit many use cases. General interactions useful in several AR applications could be selecting, viewing 3D content, navigating, or scrolling. Our generic tangible has three bulges that serve as buttons. Authors of AR applications can assign various functionalities to the buttons. A fourth bulge has the form of a house, which is commonly known as a standard symbol for a home menu. It can be assigned the functionality to switch to the AR application's home menu.

5. Different aspect ratios: The tangible's sides have different aspect ratios. Our idea here is that virtual 2D content can be shown on the side of the tangible that suits the virtual content's aspect ratio. The sides' different lengths also allow for the placement of 3D content so that it is partially occluded. The occlusion aims to increase immersion and to support users in more clearly identifying where the virtual object is located in relation to the tangible. If the handle is held horizontally, one side of the tangible can be used as a stage. Virtual objects could appear here, which can also increase presence.

6. Sufficient tracking: Stable tracking is essential because otherwise, the application cannot react appropriately to user interactions with the tangible. Therefore, a reliably trackable object should ideally combine multiple properties that make it easy for computer vision to detect and track, e.g., a textured surface and sufficient edges, dents, and bulges. However, when users hold the tangible object, they might occlude these properties partly or entirely, making the object hard to detect or contributing to a tracking loss. To provide enough detectable and trackable properties despite occlusion, each part or side of the tangible object should be reliably trackable. Our 3D printed model has two colors, and most of its sides have a few features that computer vision can detect and track. One option here is to adapt the 3D model with 3D modeling software to include more such features before 3D printing it. Another option is to print a texture on adhesive paper and glue it to the tangible. We apply the second option to our tangible because it allows us to include more features and colors than the first option. Similar to the skin model from Subsection 3.3.1, we use the tangible's virtual model as our reference for the tracking algorithm. To be able to do so, we need to apply the texture that we put on the tangible also to our virtual model.

7. Geometry or shape: According to a study by Sheridan et al. [She+5], the geometry of an object plays a crucial role in how well users can grasp it. The authors find that curves in an object's geometry or a larger surface area can improve its grip. For instance, the study shows that objects with rhomboid or star-shaped geometries are easier to grab than cubes. Moreover, the study finds that the object's material impacts its grip and that flexible materials can enhance the grip.

3.3.3 Plug System Combination of Realistically Shaped and Generic Tangible

The challenges pointed out in Subsection 3.3.1 and Subsection 3.3.2 inspired us to work on a solution combining both approaches' advantages. We combine our generic tangible object with the realistically shaped tangibles for this. Figure 3.5 shows the vertebras model combined with the generic model. We modify the vertebrae tangible and create a plug connection between it and our generic tangible. This is similar to standard plug connections, so we can call it an affordance to plug both tangibles together. The realistically shaped vertebras model can then be plugged on top of our generic tangible. Our idea is that the generic tangible is detected and tracked while the vertebras model on top of it is augmented. In this way, objects that are otherwise difficult or

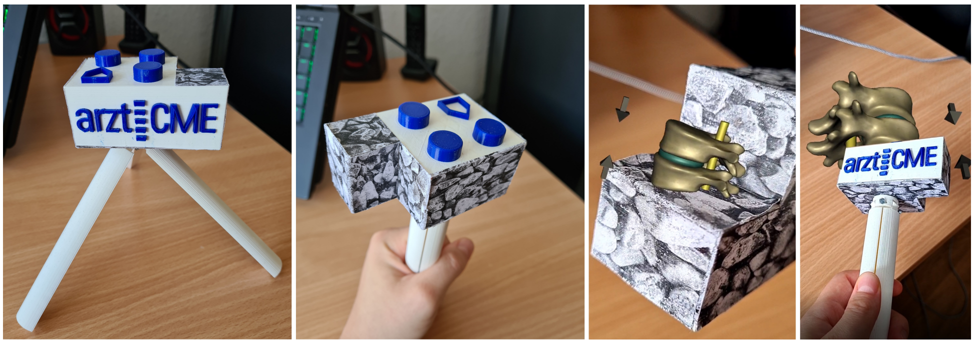


Figure 3.4: Images of our generic tangible. From left to right: 1) rested on its tripod, 2) held using its handle, 3) augmented with a virtual 3D model of vertebrae on one side that is used as a stage, 4) occluding a part of the virtual vertebrae 3D object.

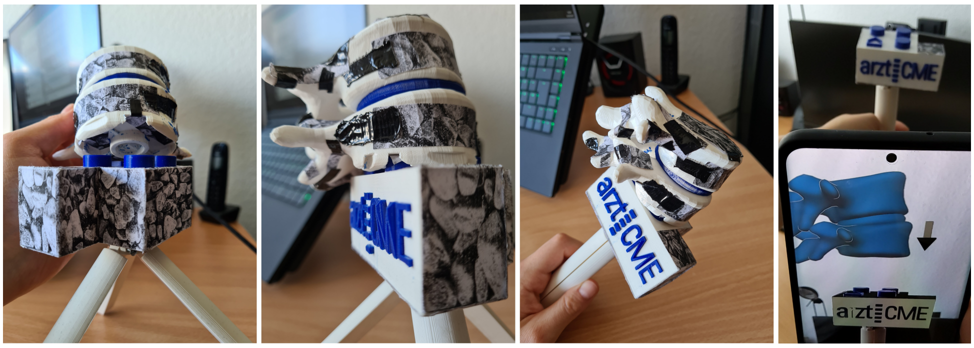


Figure 3.5: The vertebras tangible can be plugged on top of the generic tangible to receive a combined tangible. Right: An application running on an HHD guides the user to plug the vertebras tangible on top of the generic tangible.

impossible to track stable can be utilized as tangibles. The generic tangible combined with the vertebras model weighs 156g and is 10cm tall, plus the generic tangible's 14cm long handle.

By this, we also propose a way to start applications intuitively. Here, we create an application that prompts the user to plug the vertebras tangible on top of the generic tangible. Once it is plugged onto the generic tangible, the application starts a part about the vertebras. Other parts of the application could be started by plugging other realistically shaped tangibles on top of the generic tangible. Even if it is difficult or impossible to track the model plugged on top of the generic tangible, it can still be possible to identify it. This is because the tangible must be found in every or most camera frames to track it. For detecting, it is enough to start the corresponding part if the tangible is found in a single camera frame.

3.4 Integration of User Assistance in AR Nuggets

As discussed in Section 2.4, using AR can be challenging, especially for persons who have not or hardly used AR or the AR application's specific interactions yet. AR nuggets are intended to be accessible to a wide range of users, including persons without prior AR experience. Therefore, AR nuggets need to address the challenges that users may face when experiencing AR applications.

To do so, we introduce a virtual assistant that aims to help users overcome these challenges and simplify using AR nuggets. It guides users through the AR experience and proactively provides contextualized support. To identify where users typically need support, we observed users in various informal settings. The challenges we identified and where our virtual assistant could support the users include the following.

a) The lighting conditions are too dark or too bright.
b) The user holds the device too shaky and not stable enough.
c) The AR device's camera is too close or far from the tracked target or anchor.
d) The user does not point the AR device's camera towards the tracked target or anchor.
e) The user is unaware of or has trouble with input like gestures, voice commands, or touch on an HHD.
f) The user tries to use interactions like gestures or touch input that the application does not support.
g) The user is unaware that a tangible or the HHD can be grabbed and moved or that the user can move together with the HMD.
h) The user does not view the tracked target or anchor from the correct side.

Although tutorials are commonly used to address these issues, they may not cater to users' diverse experience levels. Additionally, we observed that users repeatedly needed reminders and instructions about possible interactions as well as demonstrations of gestures. Over time, they may forget the initial instructions, especially when using HMD, where they may forget how to press a virtual button or which gestures are available once the application is started. Thus, users may still encounter interaction difficulties despite a tutorial at the beginning of an AR application.

To proactively and effectively support users, we propose extending AR nuggets to detect when a user needs assistance and to provide assistance only as required. Therefore, we introduce a virtual assistant that extends the AR nuggets. If one application includes multiple AR nuggets (as described in Section 3.5), one virtual assistant supports users for all AR nuggets. Unlike a traditional tutorial, our virtual assistant focuses solely on the interaction where the user requires assistance. For instance, if an AR nugget detects poor lighting conditions, it can ask the user to turn on the lights without explaining other interactions. Each hint is provided only when an AR nugget detects that the user faces the respective challenge.

To detect the challenges that a user faces, we use data measurements from accelerometer sensors, camera images, and AR nugget-specific information. Additionally, we measure the time spent in an AR nugget when no target or anchor is detected to detect challenges d) and g) and the time frame within no input is detected to find when challenge e) occurs.

The level of assistance an individual requires varies greatly from person to person. While some may only require subtle guidance, others require more extensive support. Nevertheless, we want to avoid interrupting those who require minimal assistance. Consequently, our virtual assistant targets to provide the appropriate level of assistance for each individual person by operating on multiple stages.

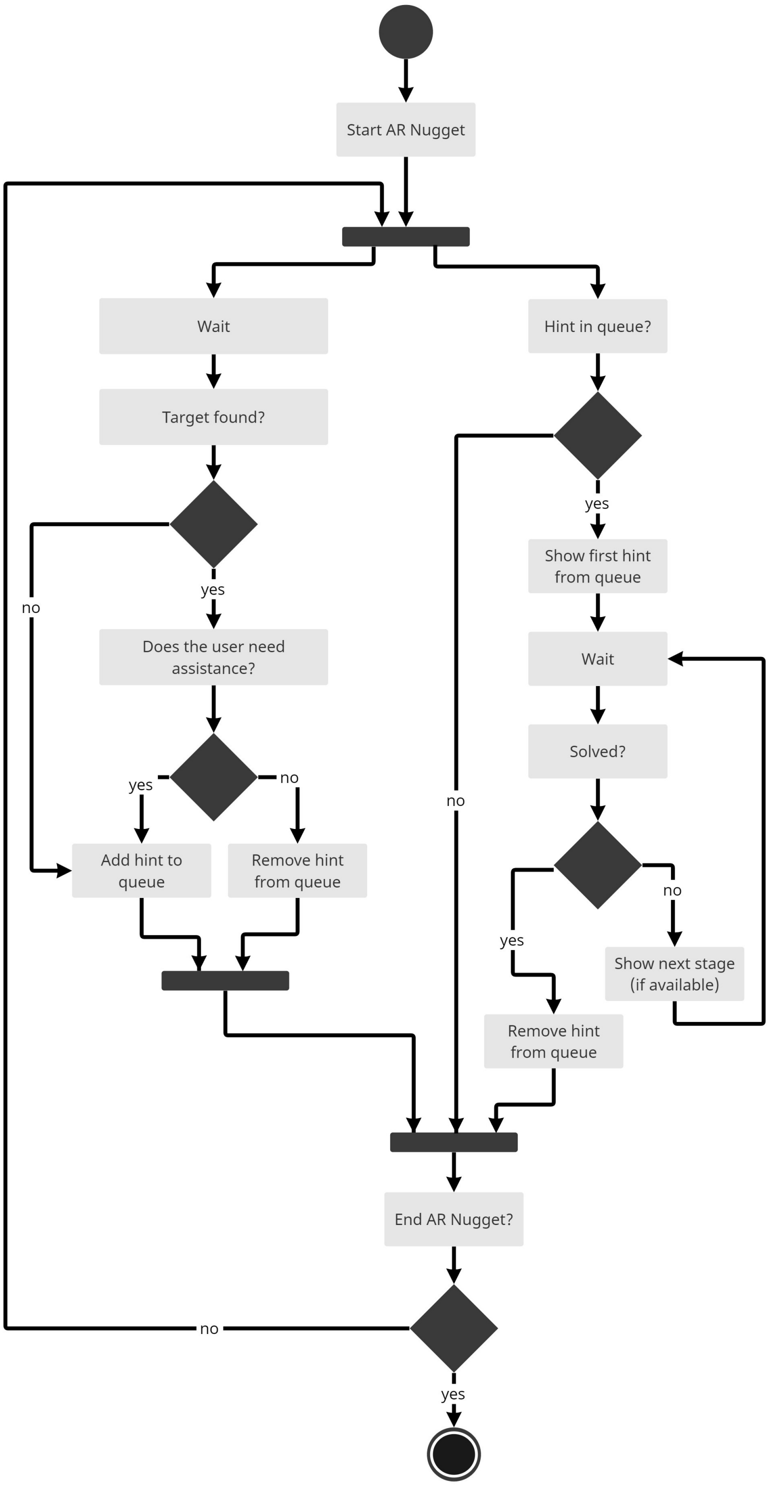


Figure 3.6: Activity diagram of the virtual assistant. When the AR nugget is started, it checks if and what kind of assistance the user needs and adds those hints to a queue. The first hint in the queue is shown and only removed once the user followed the hint.

Figure 3.6 visualizes the virtual assistant's procedure. In the first stage, the user is prompted with a hint in the form of text and an icon. We use icons similar to those used by Wu et al. [WCL19] and Renner et al. [RP18] to guide users. As soon as the user follows the suggestion provided in the hint and the challenge is no longer detected, the prompt fades away. If the system detects that the challenge persists after a certain period, it switches to the next stage. At stage two, additional information is provided. For example, for the challenge of detecting a tangible, a rotating outline of the tangible is shown in the middle of the screen. If the challenge involves gestures, an animation demonstrating the gesture is displayed. Again, the hint remains visible until the challenge is resolved. In case multiple challenges occur simultaneously, a queue for the hints is conceptualized, where the challenge is stored behind the challenges that occurred earlier. Once the current challenge is solved, the system displays the hint for the next challenge in the queue if that challenge still persists.

3.5 Usage of Multiple AR Nuggets in Complex AR Setups

Multiple AR nuggets can be combined to create a more complex AR application, such as a CME course with multiple modules or an experience with multiple exhibits in a museum. To ensure that the AR nuggets can be easily replaced or adapted at a later stage, they remain independent of each other. However, in some cases, it may be desirable for the AR nuggets to communicate with each other. For instance, if a quiz question is answered correctly, the AR experience should continue with the next AR nugget. If it is answered incorrectly, the previous AR nugget should be repeated. To enable this exchange of information, we propose an AR nugget manager that aims to ensure the independence of the AR nuggets while allowing references and connections between AR nuggets by defining pre- and postconditions. Additionally, it controls which AR nugget is executed at which time and targets to give authors an overview of their AR nuggets.

Figure 3.7 shows the activity diagram of the AR nugget manager and AR nugget to depict the execution process of AR nuggets. When an application includes more than one AR nugget, the AR nugget manager is added automatically. Once the application is started, the AR nugget manager initiates the first AR nugget, which by default is the first one created by the author, but authors can modify this order. Subsequent AR nuggets start and stop based on their defined pre- and postconditions. For instance, an AR nugget may start based on its distance to the user or when a previous AR nugget has ended. We provide a default set of versatile pre- and postconditions, but this list can be expanded based on emerging use cases.

When the current AR nugget's postconditions are met, the AR nugget sends an event to the AR nugget manager. Then, the AR nugget manager verifies whether the preconditions for any other AR nuggets are also met. To facilitate this, the AR nugget manager maintains a list of all AR nuggets whose preconditions are met. Each AR nugget independently verifies its preconditions and adds or removes itself from this list. If no AR nugget has its preconditions met, the current AR nugget continues until another AR nugget meets its preconditions. The AR nugget manager repeatedly checks if AR nuggets with preconditions met are available from the list. If one is available, the AR nugget manager starts that one after stopping the current AR nugget. For instance, in a museum visit, the first AR nugget may have a precondition that the user is in close proximity to a starting point, and its postcondition could be that the AR nugget has been fully experienced. Subsequently, the precondition for the next AR nugget could be that the previous

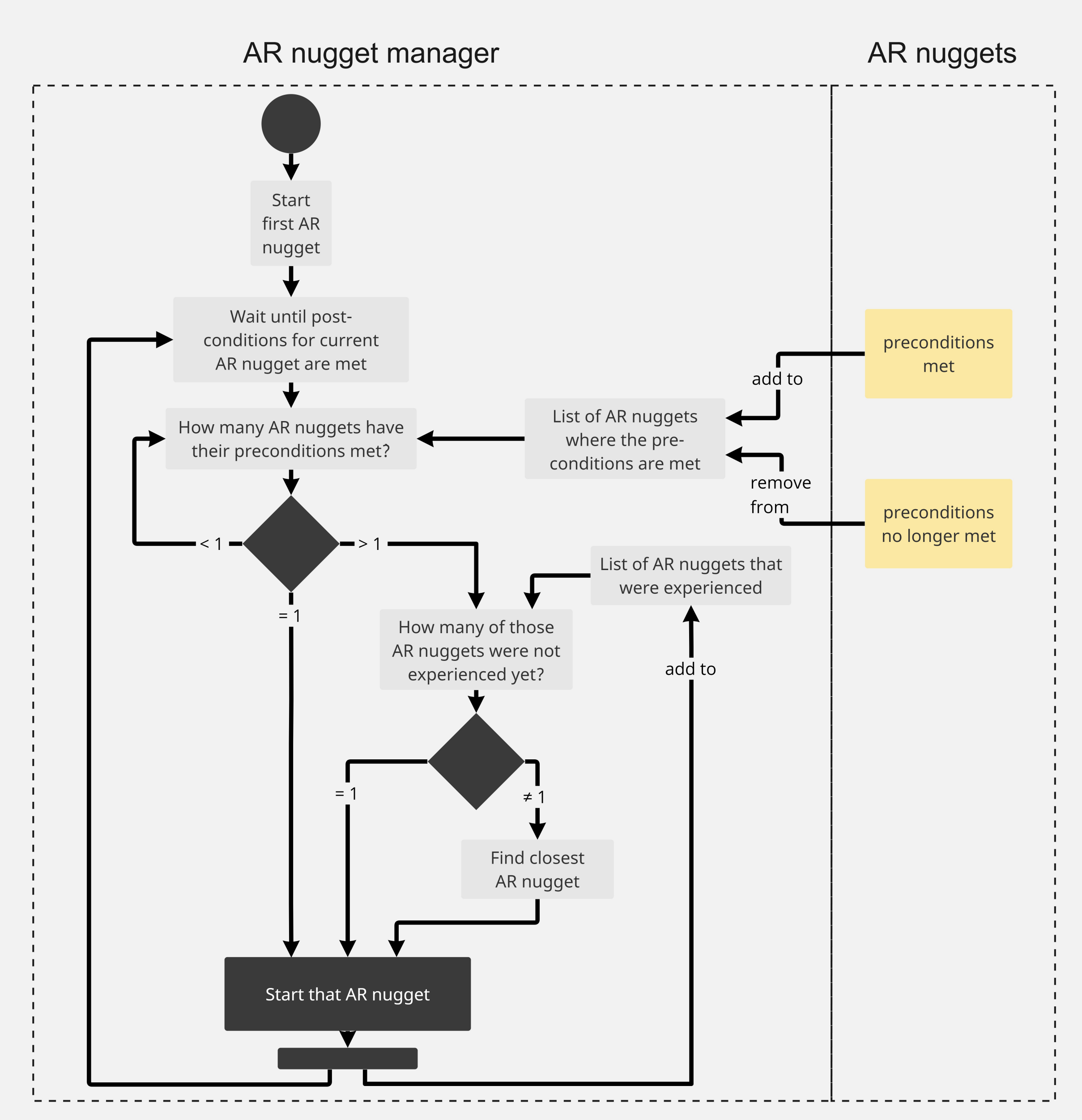


Figure 3.7: Upon initiating the initial AR nugget, the AR nugget manager waits until the postconditions of the AR nugget have been met. Once these postconditions are met, the AR nugget manager proceeds to search a list to identify AR nuggets that meet their specified preconditions. The list is continuously updated as each AR nugget adds or removes itself from the list when its preconditions are met or no longer met. If the list contains only a single AR nugget, the AR nugget manager initiates that one. Alternatively, it inspects if there are multiple AR nuggets that have not yet been experienced. In this case, the AR nugget manager initiates the AR nugget closest to the user. Otherwise, it initiates the AR nugget that has not yet been experienced. Finally, the AR nugget manager adds the recently initiated AR nugget to the list of previously experienced AR nuggets. Thereafter, it waits again until the postconditions of the then current AR nugget are met. [Rau+22a]

one was fully experienced and the user is close to it. In this scenario, the AR nugget manager would initiate the first AR nugget for the visitor to experience. After the postconditions for the first nugget are met, the AR nugget manager would check for other AR nuggets with their preconditions met. If the visitor is close enough to the other AR nugget to meet its precondition, the AR nugget manager would stop the first nugget and start the other one.

If multiple AR nuggets have their preconditions met, the AR nugget manager selects the one that has not been experienced before. For instance, suppose there are two *navigation* AR nuggets, one for each PoI, and the preconditions for both are met. In that case, the AR nugget manager would select the *navigation* AR nugget that the user has not experienced yet. If more than one AR nugget has not been experienced, the AR nugget manager selects the one closest to the user's location. If there is also more than one AR nugget with the same distance to the user, the AR nugget manager starts the first one from the list.

If multiple AR nuggets from one AR experience are spread across different rooms or floors, users may require guidance to navigate to each location. Therefore, we incorporate a guiding function within the AR nuggets themselves. As an alternative to human guides, we use a virtual avatar that guides users. A virtual entity called agent performs the calculations for the guiding process. The virtual avatar is the visual representation of the agent The guiding logic should be able to adapt to various scenarios, such as users changing their destination, taking a break, or going back to the starting point. We develop a state machine with six states that control the agent's behavior. Additionally, authors can adjust the agent's behavior by manipulating its parameters. Based on these parameters and the user's movements during the guiding process, the agent adjusts its behavior and state accordingly, as illustrated in Figure 3.8.

The agent initially starts in an idle state and continually evaluates its state during the guiding process based on the distance between itself and the user (distance agent to user d_a2u). If this distance exceeds the value specified in a, the agent moves quickly; if it falls between the values specified in a and b, the agent moves slowly, and if it falls below the value specified in b, the agent waits for the user to catch up. Based on these three states, the agent can adjust to the user's pace or pause when needed. Additionally, if the distance between the agent and user becomes larger than the value specified in c while in the waiting state, the agent moves back towards the user to avoid obstructed views behind exhibits. During the guiding process, the distance between the agent and the current node (distance agent to node d_a2n) is continuously calculated. When this distance falls below the value specified in d, the agent begins moving to the next node. If the current node is the last on the path, the agent switches to the "arrived" state. We provide default AR nuggets for navigation that include a default agent and implement a pathfinding algorithm. We offer three ways to define the AR nugget's pathway, which are described below.

Option a) needs authors to pre-define the pathway during the authoring phase by walking the space with the AR device and placing path nodes interactively. When a user starts the guiding process, the avatar moves toward the first node and automatically rotates toward the currently targeted node. Once the targeted node is reached, the avatar will move toward the next node on the pathway until it arrives at the pathway's goal. The nodes from different pathways, i.e., different AR nuggets, stay independent from each other. Therefore, one node-based guiding AR

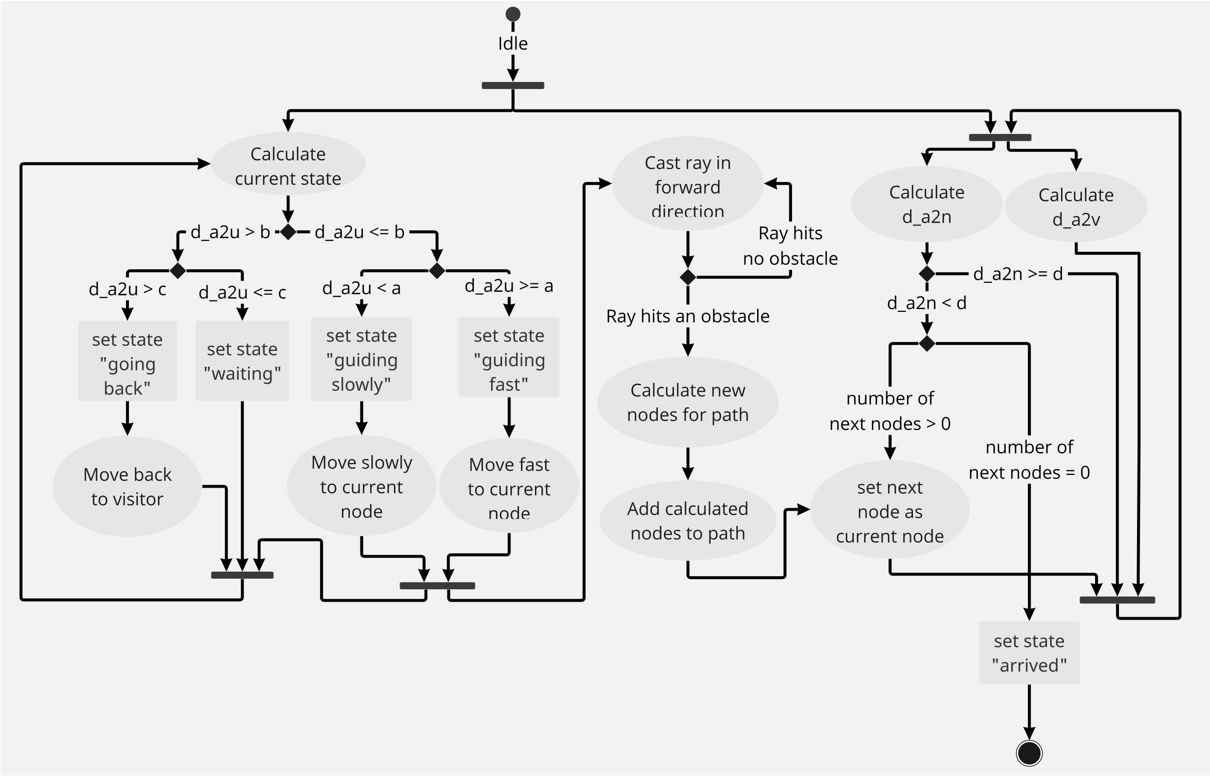


Figure 3.8: State diagram of the agent's guiding state machine. The agent continuously calculates the distance between it and the user (distance agent to user: d_a2u) and the distance between it and the current node (distance agent to node: d_a2n). The agent calculates its state by comparing d_a2u, its distance to the user, with the parameters a - c, which can be configured by authors. While moving, the agent casts a ray in the forward direction to detect possible obstacles. If the ray hits an obstacle, the agent calculates additional path nodes to avoid the obstacle and adds these nodes to the path. Depending on the state, the agent either waits for the user, goes back to the user, or moves slowly or fast to the current node. At the same time, the agent also compares its distance to the current node (d_a2n) with the parameter d, which can also be configured by authors. If the current node is reached, the agent targets the next node. If the final node is reached, its state becomes "arrived." Based on [Rau+22b].

nugget can only navigate from one point to another and cannot have multiple pathways available. To navigate along another path, another AR nugget can be started. It would also be possible to have a pool of nodes shared between all guiding AR nuggets. Then, authors would need to record which node is connected to other nodes, e.g., in a table or an array. Without this information, the avatar would fly through walls or would be unable to find the best way. This makes authoring more complex, and it is easy to forget certain connections when authoring the table.

However, our option b) allows calculating pathways at runtime so that the navigation AR nugget can be started from any point. The ad-hoc calculation of a pathway can also find other pathways if a user leaves the initial pathway. For this, knowledge about the environment is necessary. For instance, a 3D scan of the environment can be used to calculate pathways on it, which can then be mapped to the physical environment. With current technology, users without experience in 3D scanning can scan objects and rooms [Pla+21]. The AR nugget can then map the virtual 3D scan to the physical environment and calculate the pathway based on the 3D scan utilizing well-established pathfinding algorithms.

Our third option c) can be combined with options a) or b) or can be used as a stand-alone solution. AR devices continuously gather information about the user's environment. AR nuggets can use this information to detect obstacles along the path and to calculate a new sub-path around the obstacle and back to the original path. This is especially useful in dynamic environments, e.g., museums where users are surrounded by other moving people. If used as a stand-alone solution, an author would only define the pathway's goal, and the guide would navigate straight toward the destination and automatically adjust its course to avoid any barriers along the way. However, this option may not always result in the most efficient path, and there may be situations where the agent is unable to locate a path, such as when it reaches a dead end.

Multiple AR nuggets that are distributed in the environment and not located close to each other include another challenge besides navigation between them: their correct positioning and anchoring in the real world. It is important to include multiple anchor points and to make sure that all virtual elements are positioned close to their individual anchor points. Otherwise, unprecise assumptions of the AR device about the environment may lead to unprecise or wrong positions of the virtual elements (see Figure 3.9). If the AR device gathers new information about the distance or orientation between an anchor and a virtual element, the virtual elements might move unexpectedly.

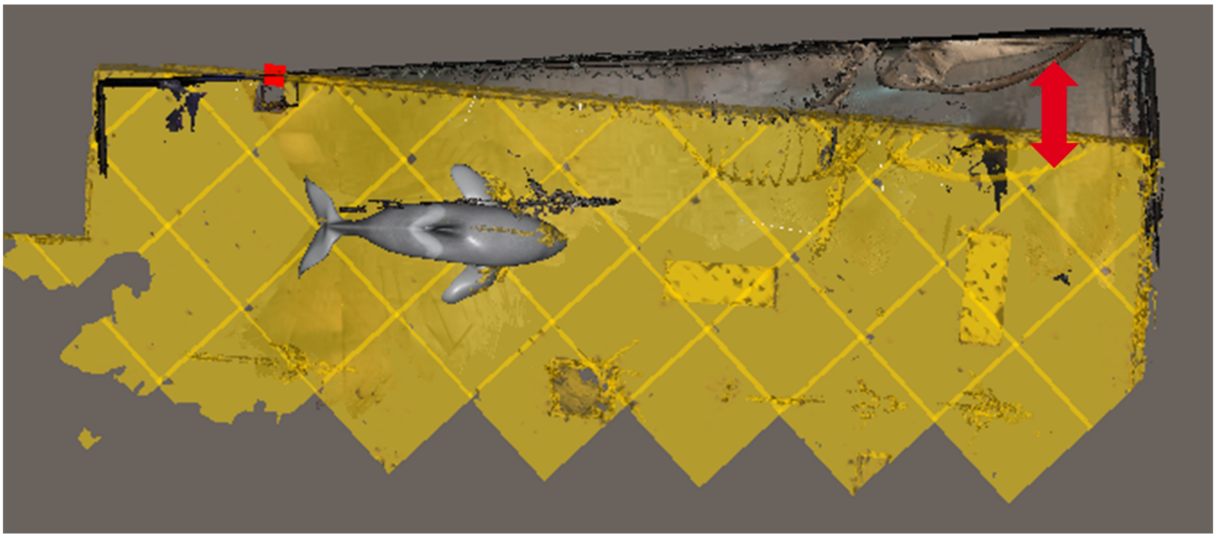


Figure 3.9: Leverage effect on virtual elements that are not placed close to their real-world anchor and therefore drifted away from the correct position. In the top-down view of a whale exhibition room, a virtual whale is augmented on physical whale bones. The real-world anchor is visualized as a red cube. A 3D scan of the room, shown in yellow color, is attached to the real-world anchor and lies in the real-world anchor's coordinate system. A slight change in the real-world anchor's rotation caused the virtual whale, the 3D scan, and all virtual elements attached to the real-world anchor to drift. Virtual elements that are far away from the real-world anchor drift the most, as highlighted by the red arrow in the figure's top right corner.

We illustrate this with an example in the context of a museum, where two AR nuggets augment two whale skeletons that are 20 meters apart. Anchoring both AR nuggets in the middle between the whale skeletons would result in each nugget being 10 meters away from its anchor. The user is standing in the middle between both whales and starts the AR experience so that the AR device starts to scan the room. From the distance, the AR device cannot precisely determine the distance from its position to the whale. Therefore, the AR device might wrongly presume that 10 meters was a point half a meter behind one whale and place the AR nugget there, behind the whale instead of on the whale. As the device continues to scan the environment, it may gather enough information to calculate distances more precisely and shift the virtual elements to their correct positions. However, this sudden movement can further disrupt the AR experience. This example shows that some AR nuggets need to support authors in minimizing this effect. To do so, it is important that authors place virtual elements close to their anchor. When an AR nugget contains multiple virtual elements that are not located in close proximity to one another, it can use one anchor point per virtual element to achieve accurate positioning.

3.6 Combination of AR Nuggets and VR Nuggets

Besides combining AR nuggets with each other, it can also be useful to combine AR nuggets with other digital applications and traditional media. This allows choosing the format (AR, VR, video, images, text) that best suits the use case. Here, we develop a concept to connect multiple AR and VR experiences to one immersive experience by using intro and outro transitions.

After the user experienced an AR or VR nugget, an outro transition takes place and brings the user back to reality. From here, an intro transition brings the user to the following AR or VR nugget. We refer to this connection from AR/VR via reality to AR/VR as a *bridge* because it connects the AR/VR nuggets like a bridge that connects two banks. Our idea is to use the same type of transition to connect two nuggets. For example, if one AR nugget ends with a fade-out transition, the subsequent nugget starts with a fade-in transition.

In our *bridge* metaphor, we define different phases of transitions as bridge piers. We apply the three outro transition phases initiation, interlude, and exit [Hor+21d] to our intro and outro transitions. The initiation is an event or action that triggers the transition's start. The optional interlude can include visualizations or interactions that take the user's focus away from the original scene. In the exit phase, the application indicates users to take the HMDs off. This phase also includes taking the HMD off and other actions to exit the virtual or augmented environment, e.g., stepping outside of a tracking area.

Together with the phase of switching HMDs in reality, this results in seven phases, i.e., seven bridge piers: 1) *outro initiation*, 2) *outro interlude*, 3) *outro exit*, 4) *reality*, 5) *intro initiation*, 6) *intro interlude*, 7) *intro exit*. 1) The *outro initiation* initiates the outro transition. This can be performed by the user or an external person, e.g., a presenter. 2) The *outro interlude* is an optional part. Whether the transition includes one or not depends on the type of transition. 3) The *outro exit* includes an indication to users to take their HMD off and the process of doing so. 4) In the *reality*, the user puts the HMD aside then grabs and puts on the other HMD. This step involves no application. 5) The *intro initiation* can be performed by the user or an external person like a presenter. 6) The *intro interlude* is an optional part, and it again depends on the type of transition, whether the transition includes one or not. 7) Finally, the *intro exit* affirms to users that the AR or VR nugget is about to begin. Figure 3.10 visualizes this notion and phases.

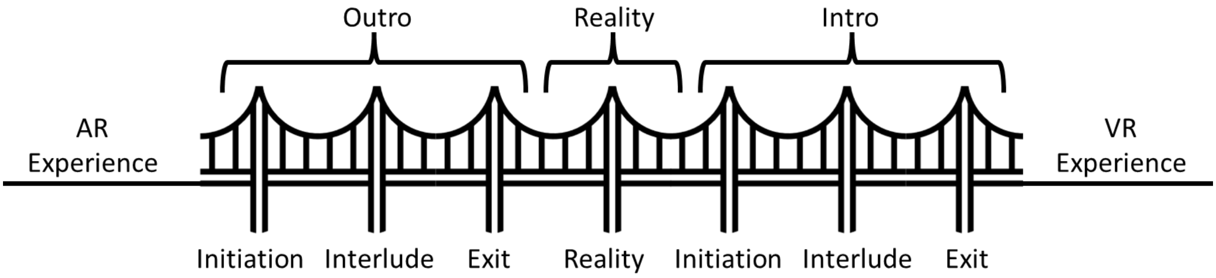


Figure 3.10: A bridge connects an AR nugget with a VR nugget. The bridge has seven bridge piers: The outro transition from the AR nugget and the intro transition to the VR nugget each consist of *initiation*, *interlude*, and *exit*. The bridge pier in the middle is the *reality*, where users can switch HMDs. [Rau+23].

We apply our bridge metaphor and create five intro and outro transitions. For all transitions, the *outro initiation* can be executed by another person than the user, e.g., a presenter or educator, or the applications could be configured to initiate the transition on their own. The *intro initiation* is triggered when the user puts the HHD on for all transitions. Our transitions do not include an *interlude* phase. In the following, we describe the transitions' *outro exit* and *intro exit* phases.

Indicator: The outro transition shows an indicator, e.g., an arrow, that points to the location of the HMD that the user should switch to (in AR) or a virtual representation of it (in VR). A text above it instructs the user to change the HMD. In the intro transition, the indicator points to the previous HMD's location, and a text thanks the user for returning the previous HMD to its place.

HMD on hand: In AR, this transition augments the user's hand with the VR HMD and in VR with the AR HMD. In the outro transition, it verbally instructs the user to switch to the HMD on the user's hand. The verbal instructions in the intro transition reassure the user that the experience continues with the HMD augmented to the hand.

Arrows: In the outro transition, virtual arrows move from the HMD on the user's head to a location where the user may place the HMD to instruct the user to place the HMD there. In VR, we add a virtual desk that represents this location. In the intro transition, the arrows move the other way to visualize that the HMD was put on, and the next experience can start now.

Portal: The outro transition shows a portal through which the user can see what to expect in the next nugget. The intro transition shows a portal through which the user can look back to the previous nugget. In AR, the portal shows the previous or next VR nugget, while in AR, it renders the camera feed and can show parts of the virtual elements from the previous or next AR nugget.

Fade: In the intro transitions, virtual elements are faded in, while the outro transition fades them out. The AR outro transition fades the virtual elements out until they disappear. In the AR intro transition, they are invisible at the start and then fade in. In this transition, virtual elements are faded in (intro transition) or out (outro transition). In VR, the virtual world fades to black (outro transition) or vice versa (intro transition).

Chapter 4

Implementation of AR Nuggets

In this chapter, we implement the concepts that we introduced in the previous chapter. First, we describe how we implemented AR nuggets in general. Then, we dovetail to implementations for tangible interactions in AR nuggets. We also present how AR nuggets implement user assistance functions for challenges that users may face when using tangible interactions or AR nuggets. Next, we implement three different types of AR nuggets for navigating between multiple other AR nuggets or PoIs. Finally, we implement combinations and transitions for AR nuggets with VR nuggets.

4.1 AR Nuggets

As proof of concept, we implemented default AR nuggets for exemplary patterns that we identified in Section 3.2. For the implementation, we use the game engine Unity. This allows the utilization of core functionalities from Unity, e.g., a runtime environment for AR devices like smartphones or tablets. As a result, our AR nuggets can be executed on different hardware devices. Furthermore, several third-party suppliers provide additional toolkits for Unity. To track tangibles, we use Vuforia Engine as a tracking toolkit. To anchor virtual elements in the real world, we use the MRTK and its spatial anchors. A spatial anchor is a representation of a specific point in the real world that can be saved and where virtual elements can be anchored.

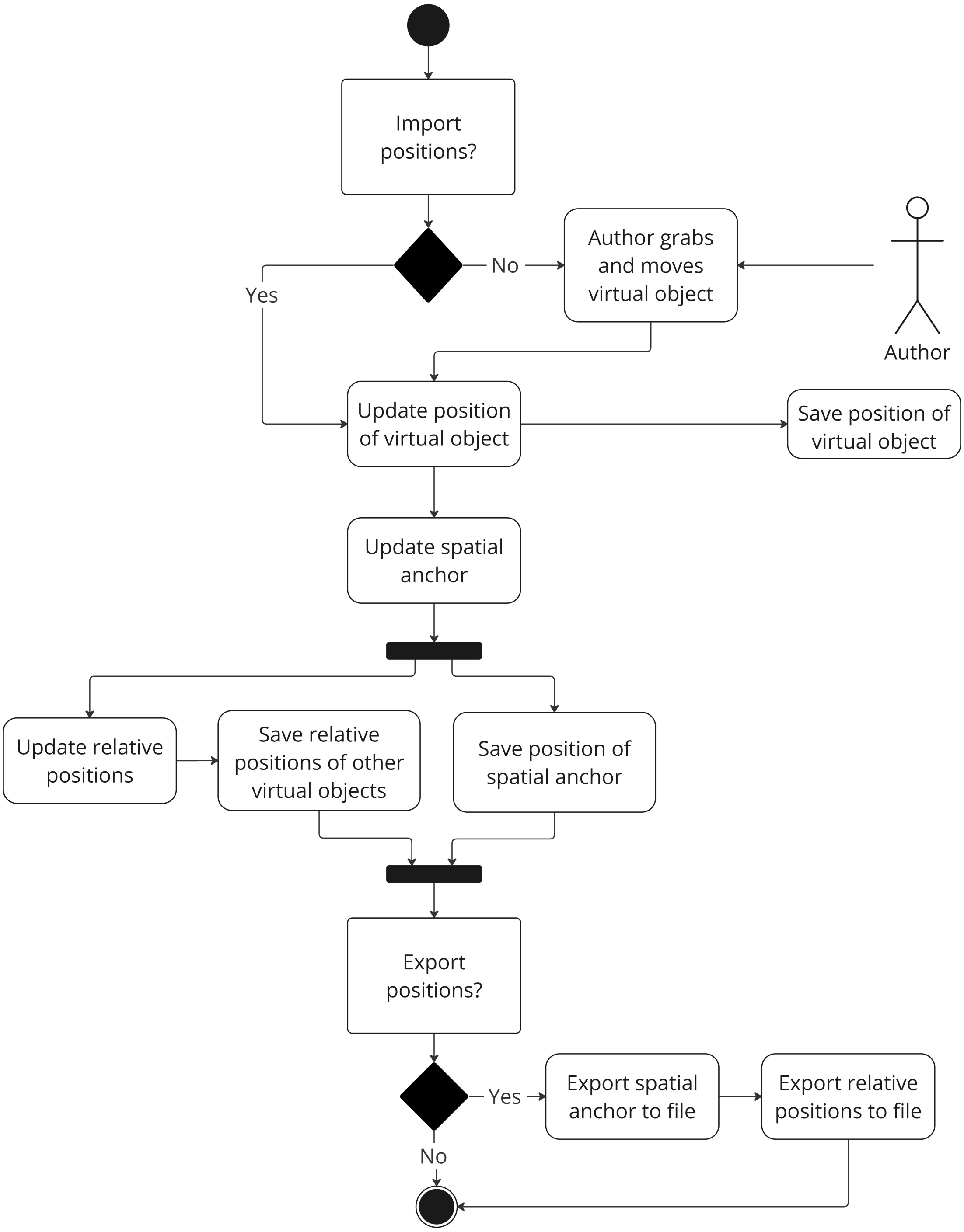


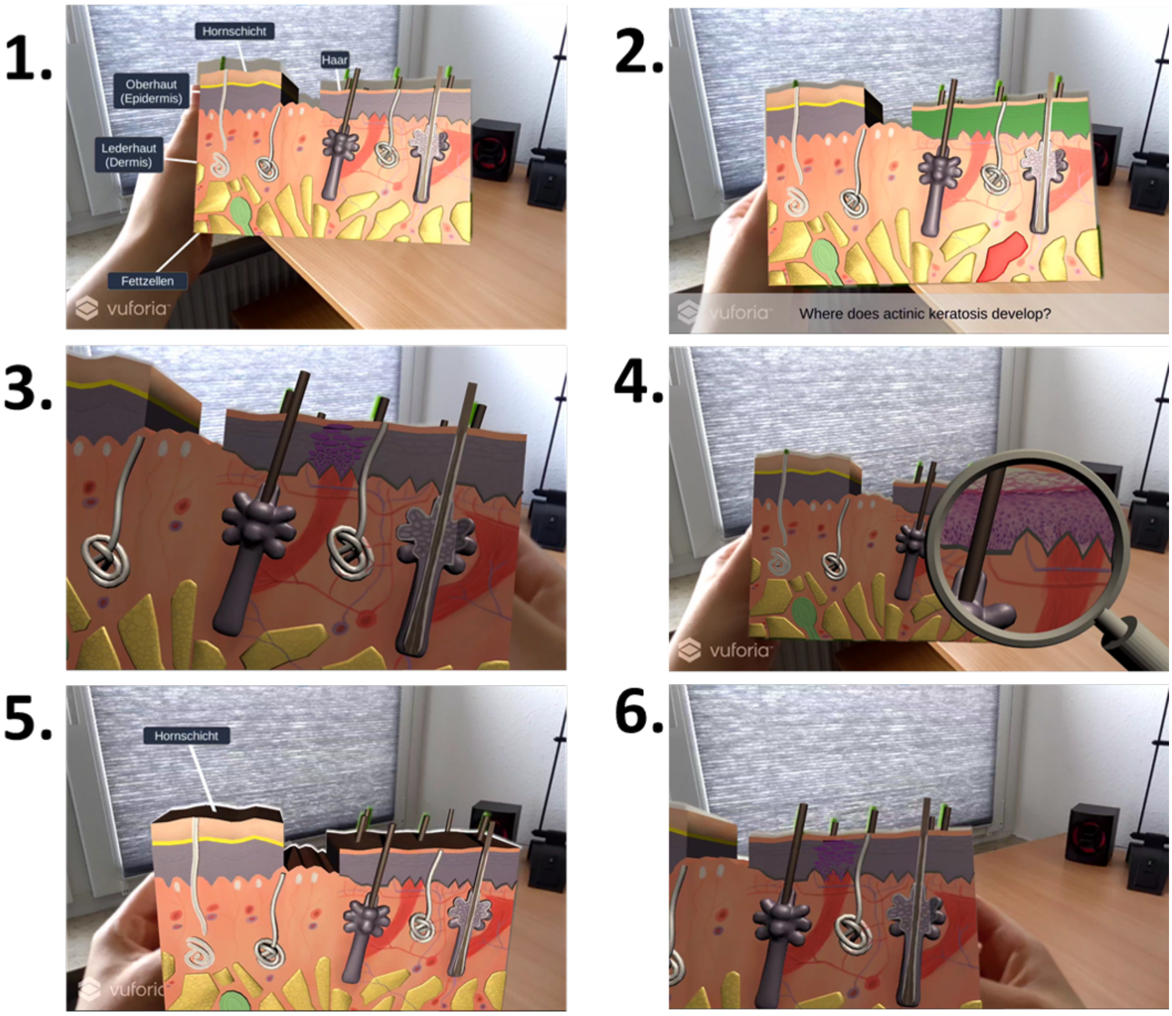
Figure 4.1: Process of importing, exporting, and updating virtual objects' positions and their spatial anchors. Spatial Anchors can be imported, or the author can move virtual objects to update the virtual objects' positions. Then, our implementation automatically updates the corresponding spatial anchors and positions of other virtual objects that have a position relative to the main virtual object. The spatial anchors and relative positions can also be exported to files. The files can be shared between devices, e.g., using cloud services or external storage mediums.

To implement an AR nugget, we defined its specific placeholder objects and default values for their parameters based on the pattern. Next, we created placeholder objects and set their parameters' default values. If a pattern involves user interactions, we defined and implement these. We save all placeholder objects as templates with Unity prefabs. Prefabs are virtual objects that can, similar to templates, store components, properties, and values. This allows the initialization of additional placeholders by drag and drop. For example, additional labels within a *show & tell* AR nugget can be initialized based on the prefab. Similarly, we save the ready-implemented AR nugget as a prefab.

Our prefabs contain scripts that realize sophisticated behaviors. For example, the label prefabs can be configured to constantly rotate towards the user so that the texts are always oriented to the screen and easily readable. Additionally, all AR nuggets that include labels automatically adapt the labels' size to their texts. We use AR nuggets in the example scenarios we introduced in Section 1.1 to show that AR nuggets and their patterns can be used in multiple use cases.

We implemented an import and export functionality for spatial anchors, targeting the HoloLens 2. These functions could also be adapted to Android's storage system and then be executed on Android-based smartphones as HHDs. Figure 4.1 visualizes the process of importing, exporting, and updating spatial anchors and virtual objects' positions. To represent the spatial anchor internally, they can be exported to a folder that is accessible through a default file explorer. This allows copying the files to other devices, where they can be imported again. One spatial anchor only applies to one virtual object and sets its position in the real world. Other virtual elements that are positioned relative to this virtual element and its spatial anchor keep their relative position to it. However, if a user or an author moves the virtual object with the spatial anchor, the new positions of virtual objects relative to it are not updated with the spatial anchor. We implemented a function that saves these positions relative to a spatial anchor in a text-based file. With this, we can import and export spatial anchors, including relative positions of virtual elements that belong to the spatial anchor. For example, one adapted *show & tell* AR nugget can include a virtual 3D model of a whale to which the spatial anchor is attached and a few labels placed relative to it. The virtual 3D model is placed to superimpose the physical whale bones as precisely as possible. The person who positioned the 3D model realizes that one of the labels does not point to the correct bone. The label can be grabbed, positioned correctly, and its new relative position is stored in a separate file when the spatial anchor is exported. The next time when the spatial anchors are imported, both, the virtual whale and the label, will be moved to the correct position.

To use AR nuggets for CME, we developed an exemplary CME course about a skin disease called actinic keratosis using AR nuggets. The CME course includes six AR nuggets (see Figure 4.2): two *show & tell*, two *progression*, one *quiz*, and one *semantic zoom* AR nugget. The first AR nugget is of the type *show & tell*. It overlays virtual labels on a tangible 3D model of a cross-section of human skin, highlighting specific parts of the skin, such as hair or fat cells. The second AR nugget is a quiz that challenges the user to identify the location of actinic keratosis on the skin cross-section by tapping on the correct virtual label. The third AR nugget is of the type *progression* and animates the progression of the disease (see Figure 4.2). It is followed by

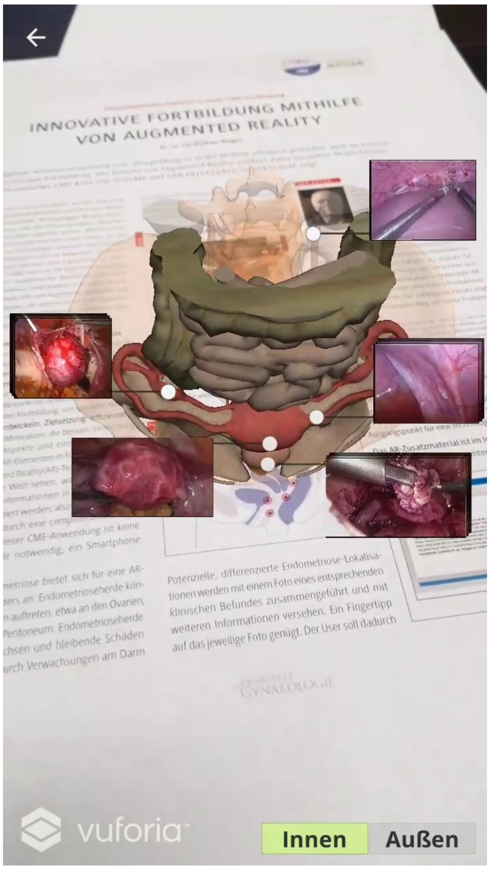


Where does actinic keratosis develop?

Figure 4.2: Screenshots of a CME course about actinic keratosis, implemented with six AR nuggets. (1) A *show & tell* AR nugget describes the skin's structure. (2) A *quiz* AR nugget challenges the user to figure out where actinic keratosis develops. (3) A *progression* AR nugget shows the disease's progression. (4) A *semantic zoom* AR nugget allows to view the affected part of the skin on a cellular layer. (5) A *show & tell* AR nugget explains the disease's effects. (6) A *progression* AR nugget shows how a further stage of actinic keratosis, called squamous cell carcinoma, develops. [Rau+21].

a *semantic zoom* AR nugget that allows users to view the disease at the cellular level through a virtual magnifying glass. Next, another *show & tell* AR nugget annotates the skin cross-section with a label, indicating which parts are affected by the disease. Finally, the course concludes with a *progression* AR nugget that visualizes a more advanced stage of the disease, called squamous cell carcinoma.

Another CME course that educates about endometriosis is certified by the German Medical Association. In Germany, physicians are required to gather CME points, for example, by completing certified CME courses. This CME course is available for physicians, so physicians can gather CME points by completing it. One *show & tell* AR nugget enhances the CME course and is now available



INNOVATIVE FORTBILDUNG MITHILFE VON AUGMENTED REALITY

Potenzielle, differenzierte Endometriose-Lokalisationen werden mit einem Foto eines entsprechenden klinischen Befundes zusammengeführt und mit weiteren Informationen versehen. Ein Fingertipp auf das jeweilige Foto genügt. Der User soll dadurch

Figure 4.3: Screenshot of a CME course about endometriosis [arz] that is enhanced with a *show & tell* AR nugget. The course includes information in the form of text and images, while the AR nugget serves as an optional part for visualization.

for physicians to get educated about endometriosis while gathering CME points. Figure 4.3 shows a screenshot of the course's AR nugget.

Moreover, we utilized AR nuggets to enhance a whale exhibition at the Senckenberg Museum, a museum for natural history. We developed two guiding AR nuggets to assist museum visitors in navigating their way to and from the exhibition. In the default *navigation* AR nugget, we replaced the default avatar with a blue butterfly and adjusted its speed. At the whale exhibition, we incorporated three additional AR nuggets: an indicator, a *show & tell*, and a *progression* AR nugget. The *indicator* AR nugget is used to indicate which whales can be explored with AR nuggets. Here, we replaced the default indicators with yellow circles on the floor. For the *show & tell* AR nugget, we replaced the placeholder with a virtual whale and adapted the labels to describe the physical whale skeleton's bones. The *progression* AR nugget explains how orca whales hunt for food. Here, we replaced the placeholder object with a virtual orca whale and animations.

4.2 Utilization of Tangible Interactions

We apply our three types of tangibles to CME and develop two educational prototypes focusing on two specific topics: the human skin and the vertebras in the human spine. Both prototypes

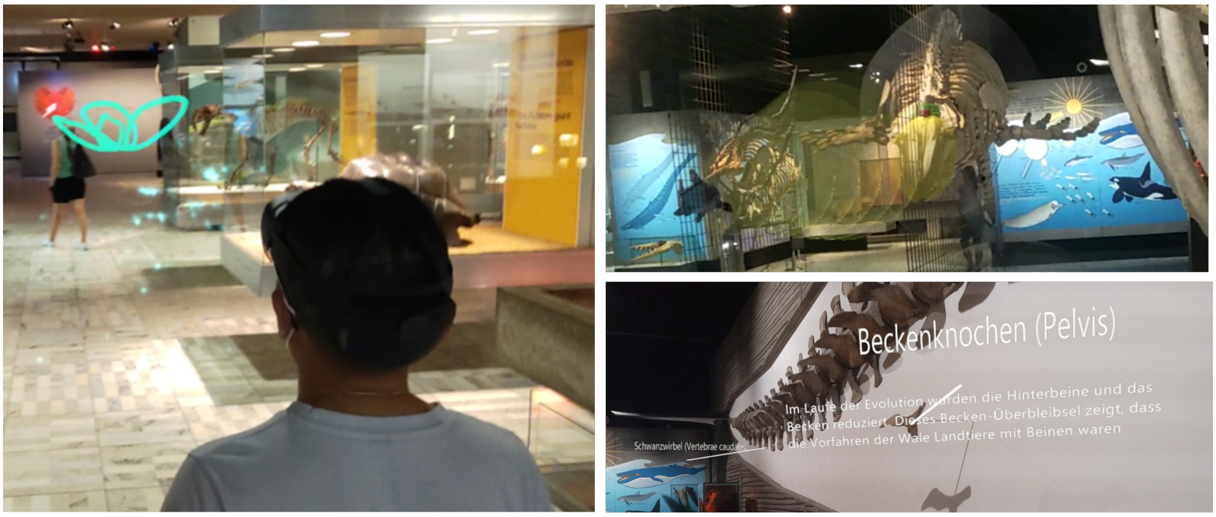


Figure 4.4: AR nuggets in the Senckenberg Museum. Left: An node-based *navigation* AR nugget utilizes a blue butterfly as an avatar that navigates a museum visitor toward the whale exhibition. Top Right: A *progression* AR nugget that visualizes how the orca whale hunts for food. Lower Right: A *show & tell* AR nugget labels the fin whale's physical bones. Based on [Rau+22a].

encompass a *show & tell* and a *progression* AR nugget and are implemented with the Game Engine Unity and Vuforia for the tracking of our tangibles.

We create two instances of realistically shaped tangibles with 3D printing. The prototype that educates about human skin employs a 3D model of a cross-section of human skin. The virtual 3D model that is used for 3D printing also serves as a reference for the tracking toolkit. For the vertebras prototype, we create a virtual 3D model depicting two vertebras along with a spinal disk, representing a segment of the human spine, and also 3D print it. Initially, the vertebrae tangible presents challenges for Vuforia's tracking system due to its symmetrical shape and limited color information. To address this, we apply a texture to the tangible's surface. We print the texture on paper and glue it to the tangible. Additionally, we apply the same texture to the corresponding virtual model using UV mapping techniques. Figure 4.5 shows both of the realistically shaped tangibles.

For the generic tangible, we utilize 3D printing to create a tangible as described in Subsection 3.3.2. To ensure reliable tracking, we also print a texture that we glue to the tangible.

For our combined tangible with plug system, the initial step involves detecting the generic tangible. In the absence of any other tangible placed on top of it, the application prompts the user to plug the vertebras tangible onto the generic one. To guide the user in this process, an animation (see Figure 4.5) is displayed, providing visual instructions for aligning and connecting the tangibles. Once the vertebras tangible is detected on top of the generic tangible, the application proceeds to start the AR nugget about the vertebrae. If users want to switch to another AR nugget, they can detach the vertebras tangible and replace it with another realistically shaped tangible, e.g., the skin tangible.

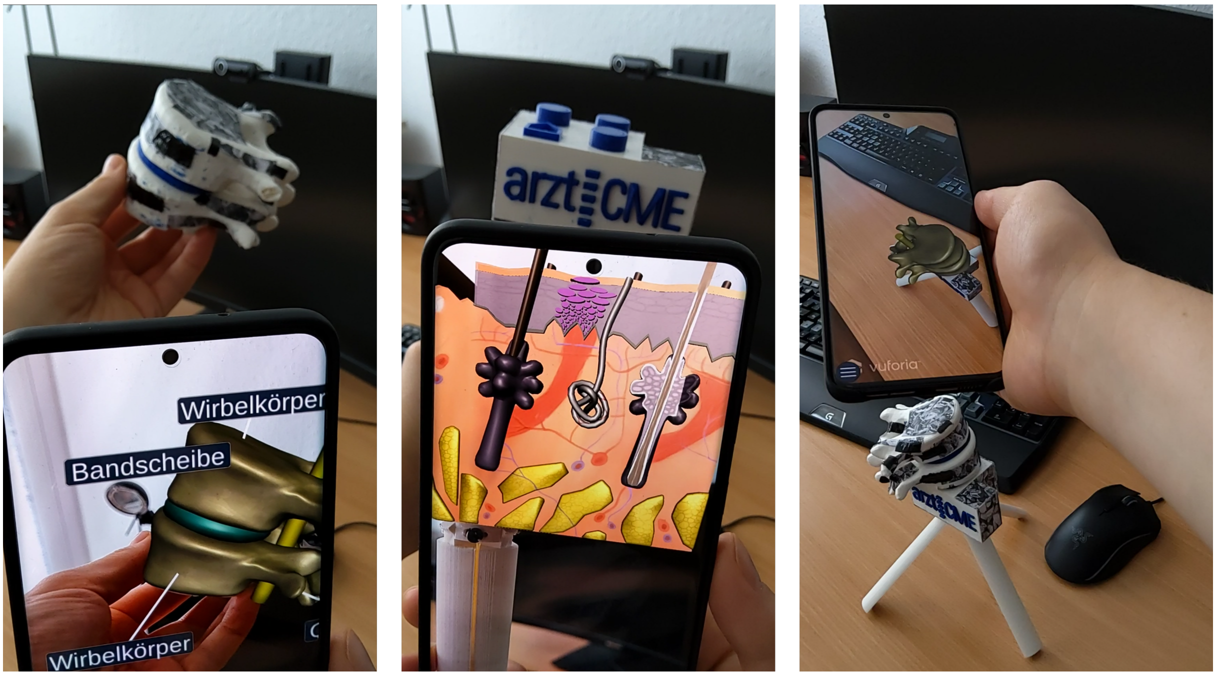


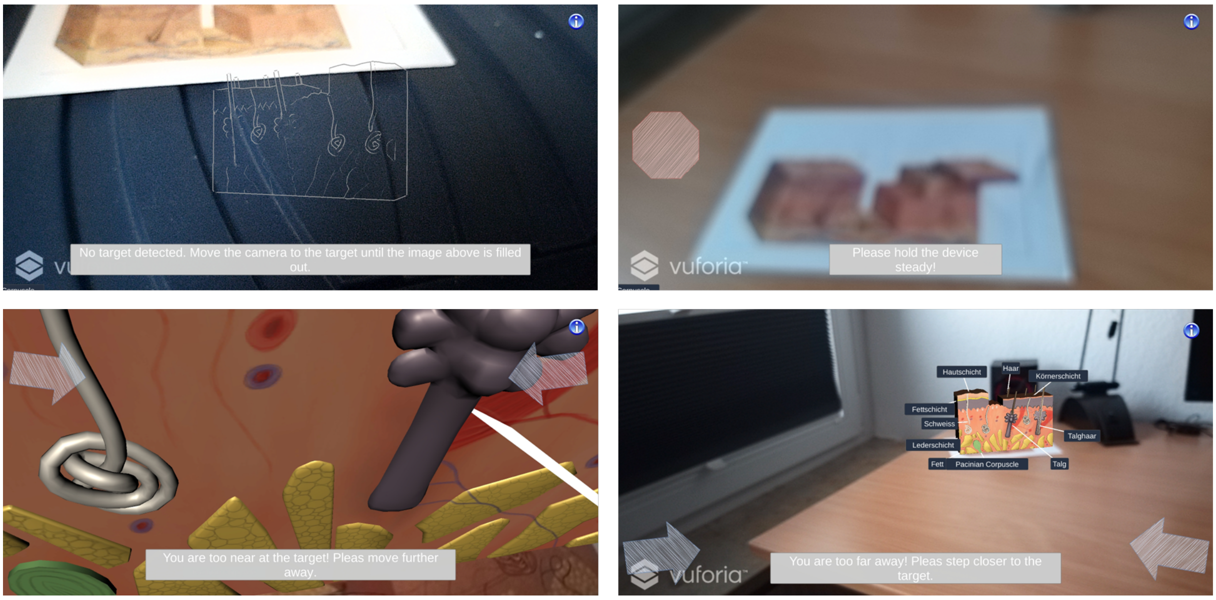
Figure 4.5: Prototype with the realistically shaped tangible interface (left), the generic tangible interface (middle), and the plug connection system interface (right).

4.3 Integration of User Assistance in AR Nuggets

Exemplary functions of a virtual assistant framework for HHD were implemented together with students of computer science using the Game Engine Unity. We applied these to AR nuggets and implement further assistance.

For the first of the two stages, each hint consists of a prompt with text and an icon. We use Vuforia to detect targets. When the target is detected, we calculate its distance to the camera and compare it to predefined thresholds. The threshold is calculated in relation to the target's size, with larger targets such as posters having larger thresholds and smaller targets like postcards having smaller ones. If the distance is not in a range within the thresholds, the target is too close or too far away for stable tracking. When the hint is triggered due to the user being too far away, the text of the hint reads "You are too far away! Please step closer to the target," and arrows are displayed to indicate that the user needs to move the AR device closer to the target. Conversely, if the user is too close, the symbols show arrows in the opposite direction, accompanied by the text "You are too close to the target! Please move further away."

To determine when to display the hint advising the user to hold the device more steadily, we access the AR device's acceleration sensors. Utilizing linear interpolation, we calculate the current acceleration of the AR device using the formula $v_{currenttimeframe} = v_{lasttimeframe} + (a - v_{lasttimeframe}) * t$, where v is the vector of the acceleration, a is the value from the acceleration sensor, and t is the timeframe between the current and the last measurements. Following this calculation, we determine the delta between the


No target detected. Move the camera to the target until the image above is filled out.

Please hold the device steady!

You are too near at the target! Pleas move further away.

You are too far away! Pleas step closer to the target.

Figure 4.6: Screenshot of the virtual assistant's hints. Top left: Second stage of hint that is activated when no target is detected. It shows a rotating outline of the 3D object and a textual hint. Top right: Hint asks the user to hold the AR device steady with a text and an icon with the shape of a stop sign. Lower left: Arrows and text informs the users that they are too close and need to move the AR device further away. Lower right: The hint shows arrows and text to ask the user to hold the AR device closer to the target. All: In the top right corner, an icon is displayed that can be clicked to play a traditional tutorial.

calculated value of v and the current acceleration. If the delta exceeds a predefined threshold, we trigger the hint that displays a stop sign and the text "Please hold the device steady!"

In cases where users are unfamiliar with gestures or experience difficulty in using them, we provide an animation that demonstrates the required gesture. To achieve this, we utilize the Hand Coach function of MRTK. This function incorporates a 3D model of a hand that visually guides users on how to interact with virtual elements. If the user attempts to use touch input on an HHD, but the scene does not support it, the hint informs the user with the message "This scene does not support any touch input. Please scan the target for more actions."

When the user fails to point the AR device at the target, resulting in no target detection, the prompt displays the message "No target could be detected. Please point the camera at the center of the target, ensuring it is clearly visible and well-illuminated." In the hint's second stage, a rotating outline of the tangible object is displayed in the center of the screen, accompanied by the prompt "No target detected. Move the camera towards the target until the image above is completely filled." In addition to the proactive assistance functions described above, we implemented a help video that users can access by clicking on a help icon located at the top-right corner of the screen. Figure 4.6 shows the implemented hints.

4.4 Usage of Multiple AR Nuggets in Complex AR Setups

If an AR application incorporates multiple AR nuggets, the application can control when to start and stop which AR nugget based on pre- and postconditions. We implemented our default AR nuggets with default pre- and postconditions. The AR nugget type determines the conditions' default values. For example, the default precondition for AR nuggets of the type navigation is that the user is close to the pathway's beginning, and their default postcondition is the guiding procedure has been accomplished. For other AR nuggets, the user's proximity to the AR nugget's content is the default precondition, and the postcondition is satisfied once the user has moved away from the nugget. Each AR nugget has dropdown menus that allow authors to alter the pre- and postconditions to choose the ones they want. By extending the code, programmers can add more pre- and postcondition types to the system. Our AR nuggets offer the following options for pre- and postconditions:

- the user collides/does not collide with another virtual object,
- the user is closer/further than a certain value to another object,
- a specific other AR nugget was experienced,
- another virtual object is visible,
- or an agent's state from a navigation AR nugget.

Additionally, authors can choose and specify various pre- and postconditions for a single AR nugget. For example, an author could specify that the preconditions of an AR nugget have been met, and therefore, the AR nugget is ready to begin when the user is close to a specified location and the agent's status is "arrived." We use the AR nugget manager to access the pre- and postconditions of the AR nuggets to start and stop the AR nuggets, depending on the conditions. One initial AR nugget that runs at the beginning without verifying that its prerequisites are satisfied can be configured by authors.

In a setup where multiple AR nuggets from one application are distributed among different rooms, navigation AR nuggets can be used to guide users to the other AR nuggets. While it is suitable for most of our default AR nuggets to be anchored in the real world by a single real-world anchor, this does not apply to navigation AR nuggets. As a rule to maintain the stability of virtual elements, Microsoft advises rendering virtual elements within three meters of their real-world anchor. Navigation AR nuggets are extensive, i.e., not all pathway nodes or a pathway's start and destination are located close to each other by three meters. Therefore, anchoring virtual elements of a navigation AR nugget to a single real-world anchor can negatively affect their stable position in the real world. Instead, we anchor each node and the pathway's start and destination with one real-world anchor. Similarly, for indicator AR nuggets, we use one real-world anchor for each indicator.

Each navigation AR nugget implements one agent. In contrast to the *navigation* AR nuggets' nodes or to other AR nuggets, the guiding avatar is not anchored to a real-world anchor. As a dynamically moving virtual object, it is instead positioned in the world coordinate system and its stationary frame of reference. Once the user starts the navigation process, the agent calculates its state, e.g., moving fast, moving slowly, or waiting, and moves depending on the state. It

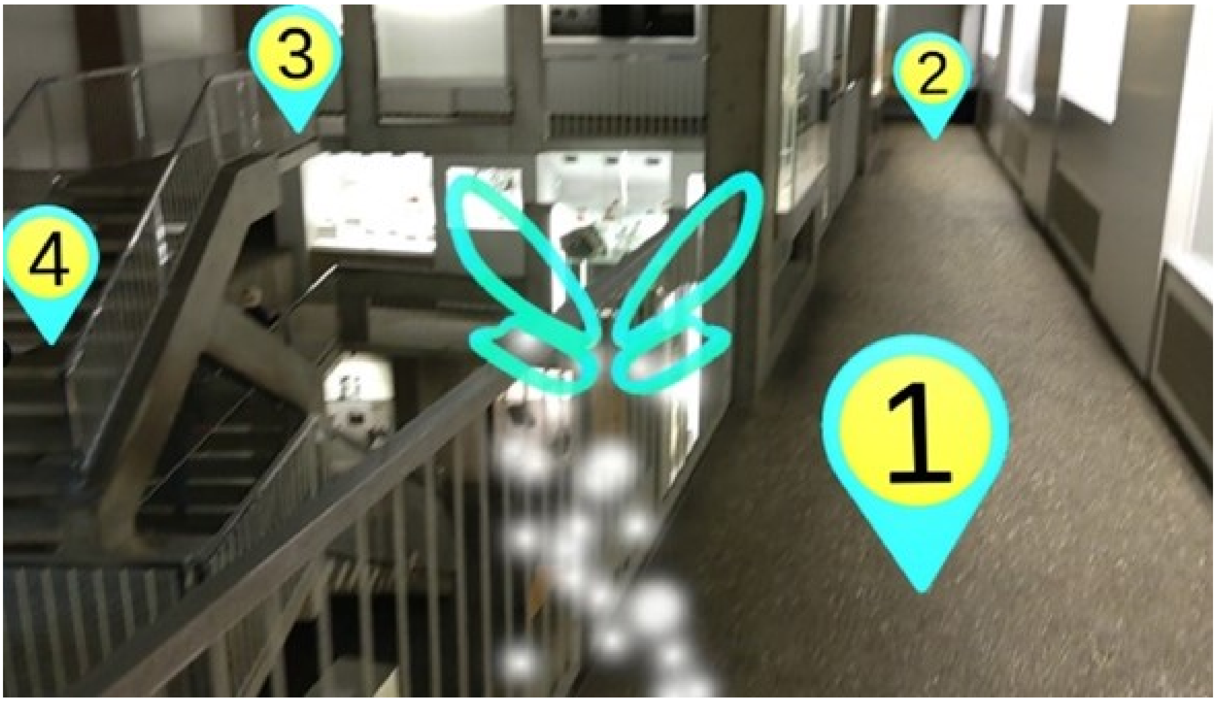


Figure 4.7: Screenshot of an AR nugget taken through the HoloLens 2. The blue butterfly guides the user through the exhibition using a pathway based on three nodes (numbers 1 to 3) and the destination goal (number 4).

re-calculates its state in asynchronous functions every few seconds and continues to do so until the navigation process is paused or completed. Additionally to the agent, each navigation AR nugget implements a menu for controlling the agent. In user mode, the menu comprises options for initiating, suspending, and recommencing guidance. Within the authoring mode, additional menu components facilitate the creation of a pathway or the establishment of an objective for a given pathway. Each navigation AR nugget implements one pathway with a single goal. To connect various pathways or provide users with a variety of options, numerous navigation AR nuggets may be included in one application. We implemented one AR nugget for each of the three navigation types node-based, based on pre-processed scan data, and based on spatial mapping information.

The *navigation based on nodes* AR nugget includes additional menu items to let authors create, edit, and delete nodes for the pathway. The authors create the pathway by placing nodes on each of the pathway's turns and placing the last node on the pathway's destination. The AR nugget represents the nodes through location markers, as depicted in Figure 4.7. The AR nugget utilizes real-world anchors to anchor the nodes in the physical world.

The *navigation based on scan data* AR nugget needs a 3D scan of the physical environment as a reference to automatically calculate a pathway. Using an indoor mobile mapping system, more precisely the NavVis M3 Mapping Trolley, we created one continuous 3D scan of several rooms. Based on the scan data, we generate an Area Target using the Vuforia Area Target Generator tool. This scan and the area target serve as the default value for the AR nugget. To execute the AR nugget at another location, authors need to replace the default scan with a scan of their desired

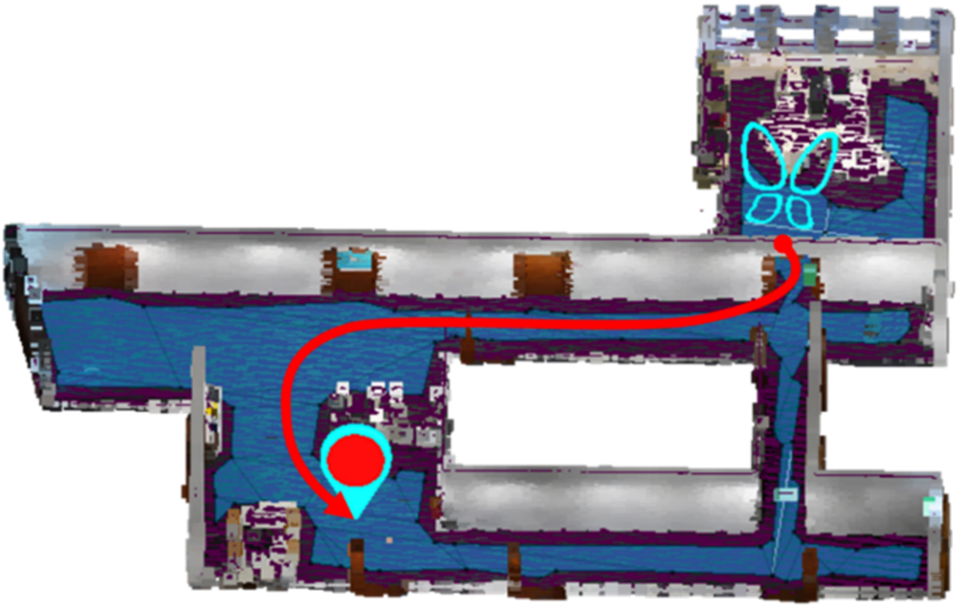


Figure 4.8: Pathway calculations based on a 3D scan that has been pre-processed. The virtual 3D model of the scan (pink color) is overlaid with the physical world (original colors) using Vuforia. Based on the scan, a NavMesh (blue color) is calculated and placed to determine a pathway. The pathway (red line) starts at the top right corner, where the Figure shows a blue butterfly as an agent, and ends at the location marker at the Figure's bottom. [Bit+22]

location. Vuforia is capable of detecting and tracking areas, thereby enabling the superimposition of virtual objects onto the scanned area. For instance, if the user situates the avatar at a particular location on the floor of the Area Target, the avatar will manifest at said location on the floor in the physical area. However, the AR nugget can only be experienced in the scanned environment, or authors need to replace the 3D scan with their own one. To calculate the pathway, we use a 3D model of the scan and anchor it to the Area Target. Figure 4.8 illustrates the scan-based approach. Vuforia's tracking algorithms align the area target, and thus also the 3D model of the scan, onto the actual physical environment. However, the 3D model is not rendered because it conforms to and blends the corresponding physical space. It functions to compute the pathway during program execution utilizing Unity's NavMesh framework. Unity can generate a mesh on a virtual environment and define traversable areas where users can walk and areas where obstacles do not allow this. This mesh is called NavMesh and can be used to calculate pathways. When the 3D model of the scan is aligned with the physical environment, the AR nugget bakes the NavMesh during runtime. This calculates where in the real world the NavMesh is anchored and thus where in the real world traversable areas are. At runtime, the traversable areas on the NavMesh are utilized to compute the pathway between the agent's location and the objective using the A* algorithm [HNR68], which is a common pathfinding algorithm.

One difficulty is that the agent can only calculate a pathway if one exists on the NavMesh, which implies the start and endpoint must be placed on the NavMesh. Additionally, the NavMesh should have no holes from missing data because the pathway can only be calculated where data is. To determine the best pathway, we can use characteristics that indicate if one option for a path

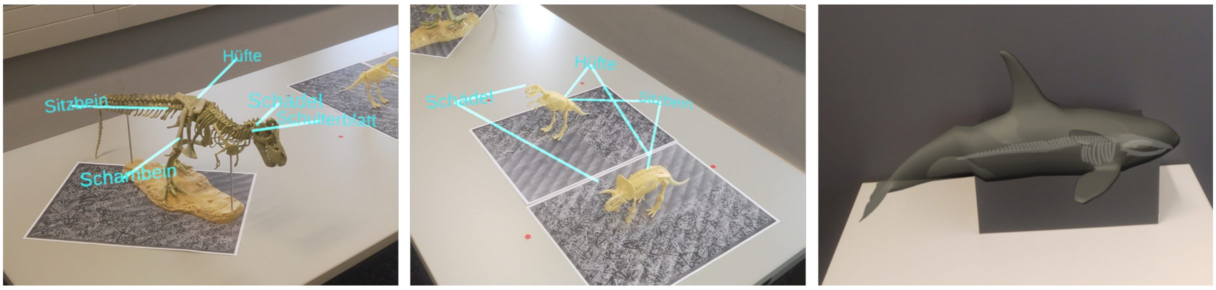


Figure 4.9: The three AR experiences. Left: A physical dinosaur skeleton augmented with text labels. Middle: Two physical dinosaur skeletons are compared to each other with virtual text labels. Right: A physical whale skeleton is augmented with a virtual 3D model of the whale's skin. [Rau+23].

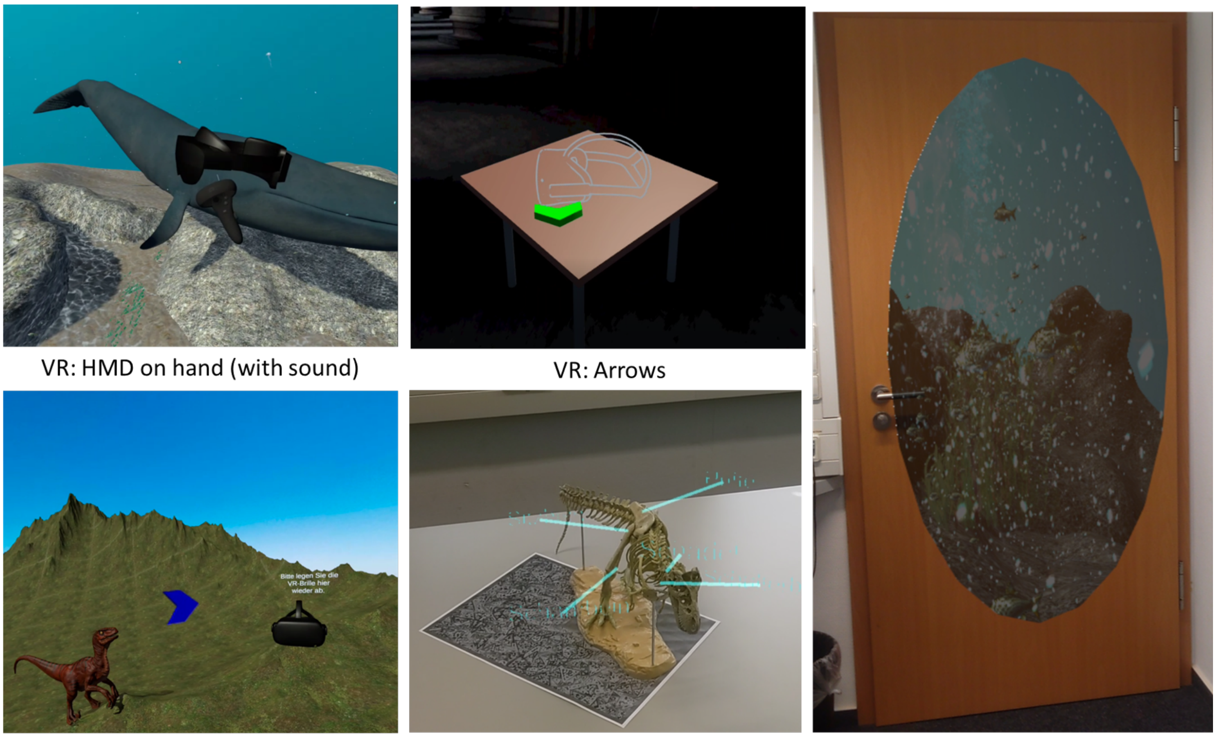
or part of it is more difficult than another one. For example, it might be shorter by distance to go via a heavily trafficked passageway to reach one's destination, but because it is so overcrowded there, it would be faster in time to go around. The height of the steps is another consideration, e.g., users may be able to utilize the stairs but may not want to jump over a one meter high block. In our early prototype, we discovered that Vuforia's mapping process of the Area Targets required a lot of processing power, which is why our agent computations and rendering became unsatisfactory. Therefore, the AR nugget uses the Area Target only to establish the location and orientation of the 3D model at the beginning and then uses a real-world anchor to secure it. Then, the AR nugget periodically verifies that the 3D scan's location and rotation still fit and updates it if needed.

The *navigation based on spatial mapping* AR nugget starts to navigate users straight in the destination's direction. It uses real-time information that the AR device gathers to detect if physical objects or parts of the environment block the pathway. If that is the case, the AR nugget uses real-time information about its environment to calculate a sub-pathway and sub-nodes around the obstacle. Here, we implemented the A* path search algorithm [HNR68].

4.5 Combination of AR Nuggets and VR Nuggets

To combine AR nuggets not only with each other but also with VR nuggets, we implemented the five transitions from Section 3.6. Furthermore, we implemented three AR and three VR nuggets that could be experienced in the context of a natural history museum. For the implementations, we use the Game Engine Unity and the Mixed Reality Toolkit. Figure 4.9 shows screenshots of the AR and VR nuggets through the HMDs. These nuggets can be connected by our intro and outro transitions that we visualize in Figure 4.10.

To control when which transition starts, we use the network technology Photon PUN for the VR nuggets and Bluetooth for the AR nuggets. For example, one museum visit with the three AR and three VR nuggets, connected with transitions, could look as follows.



VR: HMD on hand (with sound)

VR: Arrows

VR: Indicator

AR: Fade

AR: Portal

Figure 4.10: Screenshots of our five transitions. Upper Left: A virtual HMD is augmented to the user's hand, where the VR controller is visible. Lower Left: A virtual HMD with text is placed in the virtual environment, and an indicator points to it. Upper Middle: A virtual desk is placed in the virtual environment. The outline of an HMD is augmented to the desk. Lower Middle: Virtual text labels that are half transparent during the fade transition. Right: In AR, a portal to the next VR experience (a virtual underwater scene) is shown. [Rau+23].

1. The prototype starts the dinosaur exhibition in the museum. Here, the user experiences a *show & tell* AR nugget where the physical dinosaur bones are augmented with virtual text labels. The AR nugget ends with a fade-out outro transition.
2. After switching HMDs, the visitor experiences a *progression* VR nugget that starts with a fade in intro transition. The VR nugget shows a virtual world that represents the dinosaur's natural habitat and visualizes how the dinosaur moved. It ends with an arrow outro transition.
3. The visitor again switches HMDs, and the next nugget is started with an arrow intro transition. It is a *compare* AR nugget that compares the dinosaur's bones to the bones of another dinosaur using text labels. The AR nugget ends with a portal outro transition. The portal shows what the visitor can expect in the next nugget. In this case, the visitor can see an underwater world through the portal.
4. The visitor walks from the dinosaur exhibition to the whale exhibition next to it. After switching to the VR HMD, a portal intro transition shows what the user just experienced in the previous nugget, in this case, the dinosaur. Then, the user experiences the VR nugget,

which shows an animated 3D model of a whale in an underwater scene. The VR nugget ends with an HHD on hand outro transition.

5. The visitor switches to the AR HMD where an HHD on hand intro transition starts. The *superimposition with interactive transparency control* AR nugget augments the physical whale bones with the whale's skin and ends with an indicator outro transition.
6. After a final switch of HMDs, the final VR nugget starts with an indicator intro transition and then continues to show a 360° video about volcanoes and their eruptions.

Chapter 5

Authoring With AR Nuggets

This chapter introduces novel AR authoring approaches and tools based on AR nuggets. An AR nugget authoring tool is an application or parts of an application that support authors in adapting AR nuggets. When creating an AR nugget authoring tool, one challenge is to design it in a way that allows authors without experience with AR or programming knowledge to work with it without overwhelming the authors with options. Simultaneously, more experienced authors may need an authoring tool that offers more possibilities to realize their ideas. As pointed out in Subsection 2.6.2, different types of authoring tools exist that target authors with different fields and levels of expertise. This chapter introduces three different authoring approaches that target authors with different fields and levels of experience. AR nuggets can be exchanged between the different authoring tools as introduced in Section 5.4. First, we introduce one novel authoring tool that allows adapting AR nuggets without programming. This authoring approach incorporates different degrees of immersion. Additionally, we introduce another authoring approach that incorporates multiple small authoring tools integrated into an existing authoring environment. This authoring approach distinguishes between process-specific and location-specific authoring tasks. Furthermore, we explore an authoring approach based on constraints. Finally, we describe how AR nuggets can be integrated and added to other existing applications.

5.1 AR Nugget Authoring Using Different Degrees of Immersion

In this section, we develop a stand-alone authoring tool that authors can use to adapt AR nuggets. Similar to AR nuggets, the authoring tool targets to support persons without programming knowledge. The AR nugget authoring tool utilizes different degrees of immersion by using non-immersive desktop computers and HHDs as immersive AR devices. We call this AR nugget authoring tool ARNAUDDI (AR Nugget Authoring Using Different Degrees of Immersion).

5.1.1 Concept

An AR nugget authoring tool should support authors in customizing, replacing, and adding 3D objects to adapt AR nuggets. Because AR nuggets always remain in an executable state, an

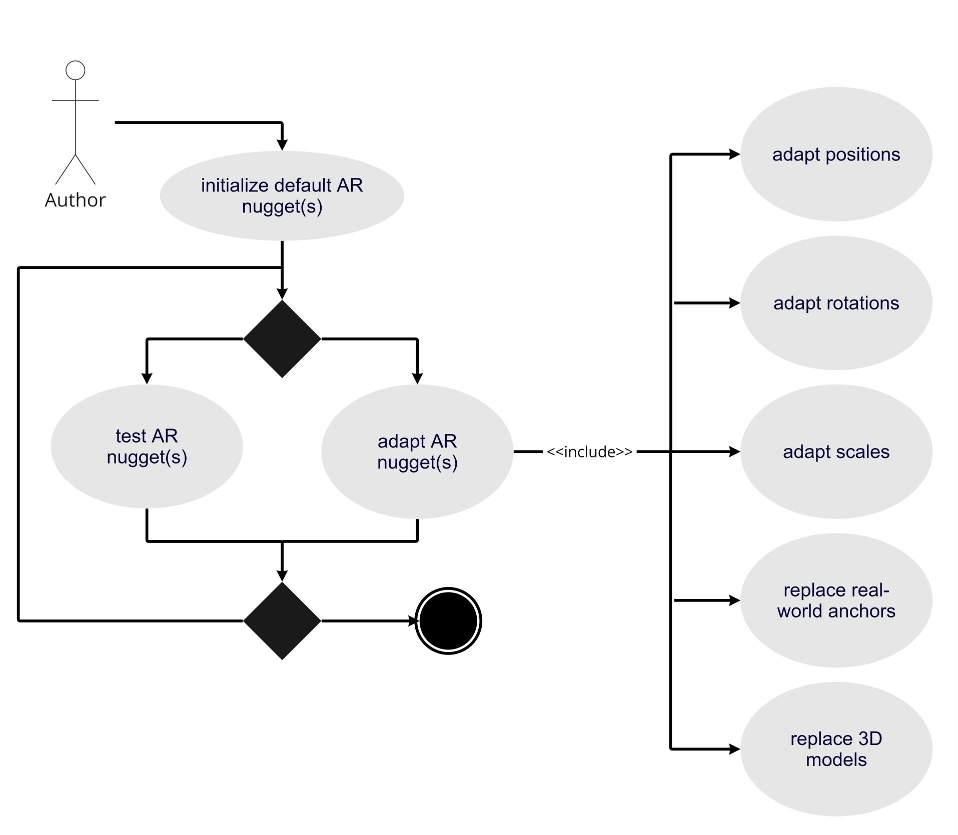


Figure 5.1: Conceptual AR nugget authoring workflow. Authors initialize an AR nugget that they can then adapt or test. Adapting includes adapting positions, rotations, and scales and replacing 3D models and real-world anchors.

AR nugget authoring tool should enforce that mandatory objects can only be replaced but not deleted. For example, the default quiz AR nugget includes one mandatory 3D object for the correct and one for the wrong answer. If these mandatory 3D objects were deleted, the AR nugget would no longer be executable and functional. Thus, the authoring tool should allow authors to replace these 3D objects but not delete them. This limitation provides support to authors by guaranteeing the continued executability of the AR nugget. If authors intend to remove one of these objects and repurpose the AR nugget in a different manner, the AR nugget authoring tool could propose an alternative type of AR nugget. If no suitable default AR nugget is available, AR nugget authors and developers (see Subsection 3.1.2) can create a novel one.

In the following, we develop an authoring workflow for how authors can adapt AR nuggets and describe it from an author's point of view. We also visualize the workflow with Figure 5.1. As the first step in the authoring workflow, authors can choose a default AR nugget. For this, the AR nugget authoring tool should provide a collection of AR nuggets, with one default AR nugget for each AR nugget type. The default AR nuggets can help authors get a first impression because they can experience them as a first step. For example, an AR nugget authoring tool could provide default AR nuggets of the types *show & tell*, *quiz*, and *semantic zoom*.

In an optional second step, authors can adapt the AR nugget by replacing the placeholder objects, adapting the default parameters' values (including position, rotation, and scale), and replacing the real-world anchors with custom ones. Authors may work on these adaptions in any order. However, it can be beneficial to suggest a structured workflow, particularly for novice authors. Therefore, we suggest authors to start with steps that make prominent changes. Our authoring tool suggests starting with adapting an object's real-world anchor and then continuing with replacing the placeholder 3D object. Finally, authors can adapt nugget-specific parameters like a quiz question or label text and move, rotate, or scale virtual objects.

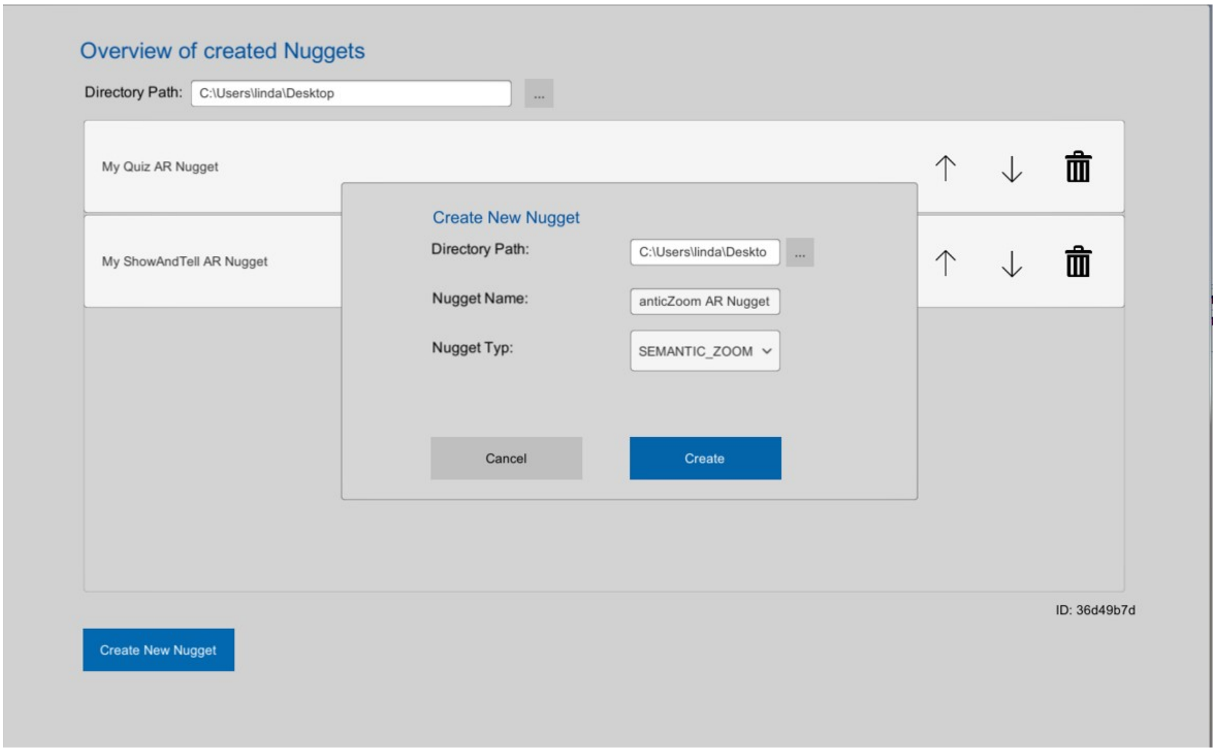
As third step, authors can experience the AR nuggets. During the whole workflow, the AR nuggets remain in an executable state and the authors can switch back and forth between experiencing and adapting. For example, as first step an author could choose to start with a *quiz* AR nugget. Before adapting the AR nugget, the author skips the optional second step of adapting the AR nugget and goes straight to experiencing it to get a first impression. The author finds the AR nugget type suitable and starts to adapt it by replacing the real-world anchor and 3D objects. To test the adaptions, the author experiences the AR nugget again. The author continues to switch between adapting and testing until the AR nugget meets the expectations.

We distinguish two versions of our authoring tool ARNAUDDI; one targeting desktop computer and one targeting HHDs. The desktop application can be used as a stand-alone tool and includes all of ARNAUDDI's functionalities. This version also includes a preview function that allows experiencing the AR nuggets in a non-immersive environment. With the HHD, authors can immersively experience the AR nuggets. The version for HHDs incorporates preview functions and adds the option to experience and adapt AR nuggets immersively. Because the HHD's screen size is smaller than the one from the desktop computer, the HHD's screen is likely too small to fit all authoring UI elements like the desktop computer version. Therefore, we popose to include only selected functionalities and UI elements in the HHD version that allow adapting parameter values that require 3D interaction, like adapting positions, rotations, and scale. We also support replacing 3D objects with the HHD. The two versions of ARNAUDDI need to synchronize changes to the AR nuggets, i.e., changes made using the desktop computer need to be synchronized to the HHD and vice versa. In this context, a network connection may be employed.

5.1.2 Implementation

We implement ARNAUDDI based on the game engine Unity. Additionally, we implement three default AR nuggets for ARNAUDDI using Unity to show its feasibility. This allows using core functionalities from Unity, like runtime environments for multiple AR devices, e.g., Android and iOs based smartphones and tablets. With these, ARNAUDDI and its AR nuggets can be executed on different hardware devices. We use Vuforia to detect and track images in the real world that serve as real-world anchors for the AR nuggets.

Authors can install and start ARNAUDDI on a desktop computer and an HHD as an AR device. On the desktop computer's start screen, authors see a server ID and an overview of all previously created AR nuggets with the option to create new ones. Authors must enter the server ID on the HHD's start screen to connect the HHD to the desktop computer. When the devices are connected, ARNAUDDI synchronizes any changes made on one device to the other device. To initialize new



Directory Path: C:\Users\linda\Desktop

My Quiz AR Nugget

My ShowAndTell AR Nugget

Directory Path: C:\Users\linda\Deskto

Nugget Name: anticZoom AR Nugget

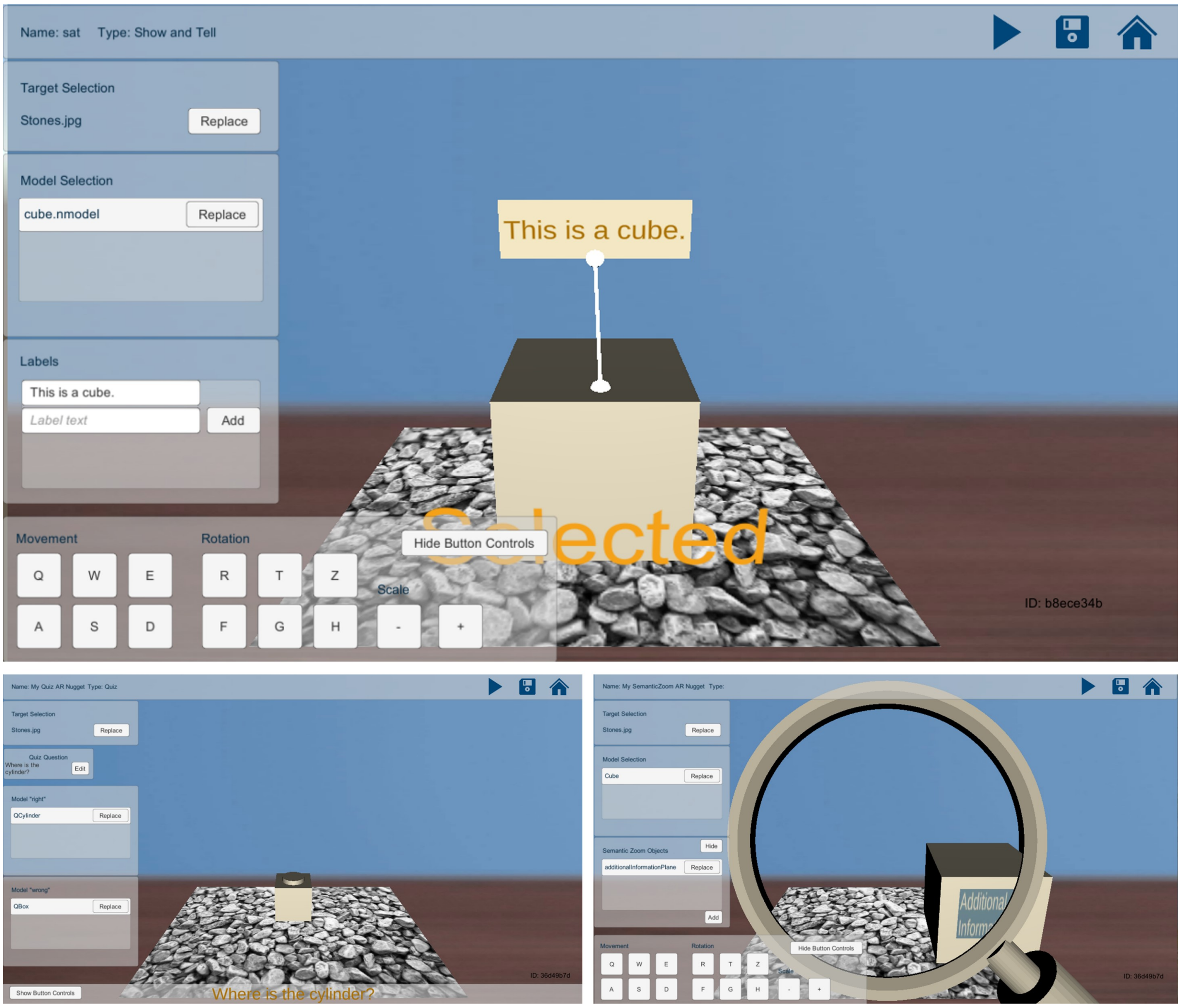
Nugget Typ: SEMANTIC_ZOOM

ID: 36d49b7d

Figure 5.2: Screenshot of the AR Nugget authoring tool ARNAUDDI on the desktop device. Using the drop-down menu, authors can initialize new default AR nuggets. With the list in the background, authors can select AR nuggets to experience and adapt. [Rau+22b].

default AR nuggets, authors can use ARNAUDDI on the desktop computer. Authors can select a default AR nugget from a drop-down menu and enter a custom name to do so. Then, as shown in Figure 5.2, the AR nugget is added to a list. Authors can experience and adapt an AR nugget from the list by clicking on it. If authors create multiple AR nuggets, ARNAUDDI allows them to customize the order of succession by sorting the list.

After initializing a new default AR nugget, authors can adapt the AR nugget using ARNAUDDI's UI that Figure 5.3 shows. The default image target that serves as a real-world anchor is an image with stones, a default image from Vuforia. Authors can use the image file to print this image on paper. The layout of the menus on the left side of ARNAUDDI's UI suggests the workflow described in Subsection 5.1.1. The top left menu implements the *Target Selection*. Here, authors can replace the default stones image with a custom one. When authors click the replace button, a file explorer opens, allowing them to select the custom image they want to use as a replacement. For the *quiz* AR nugget, the menu below the *Target Selection* allows editing the quiz question. Below these menus is the menu for *Model Selection*, which provides the option to replace the placeholder 3D objects. In the case of the *quiz* AR nugget, it is divided into a menu for each category of correct and incorrect answers. Likewise, the menu for the *semantic zoom* AR nugget separates model selections for the main 3D object and semantic zoom objects. Depending on the application pattern that the AR nugget is based on, additional 3D models can be added or deleted. Located at the bottom part of the UI are the menus that relate to the manipulation of


Name: sat Type: Show and Tell

Target Selection: Stones.jpg

Model Selection: cube.nmodel

Labels: This is a cube.

ID: b8ece34b

Name: My Quiz AR Nugget Type: Quiz

Target Selection: Stones.jpg

Quiz Question: Where is the cylinder?

Model "right": QCylinder

Model "wrong": QBox

ID: 36d49b7d

Name: My SemanticZoom AR Nugget Type:

Target Selection: Stones.jpg

Model Selection: Cube

Semantic Zoom Objects: additionalInformationPlane

ID: 36d49b7d

Figure 5.3: Screenshots from the AR Nugget authoring tool ARNAUDDI on the desktop device. Top: A default *show & tell* AR nugget with the placeholder 3D object selected. Lower left: A default *semantic zoom* AR nugget. Lower right: A default *quiz* AR nugget. [Rau+22b]

movement, rotation, and scale. In Figure 5.3 (top), the author selected the placeholder 3D object, a cube, by clicking it. Authors can move, rotate, and scale selected objects using the buttons on the lower left side of the UI or the keys on the computer's keyboard. As seen in the *quiz* AR nugget in Figure 5.3 (bottom left), authors can also hide the UI buttons for movement, rotation, and scaling. By clicking the play/pause button in the top right corner of the UI, authors may switch between edit and preview mode. The UI's top right corner also contains buttons for saving and returning to the AR nugget overview. The bottom right corner displays the server ID required to connect the AR device to the desktop computer. For the HHD, we decide to omit the UI elements for replacing, adding, or deleting custom 3D objects to maximize the screen space available for the camera stream and augmentations. The custom 3D objects are likely stored on the desktop computer and not on the HHD. Thus, this function is more relevant for the desktop computer version. In contrast, interactions like moving and rotating 3D objects are relevant for the HHD because they can facilitate 3D interactions. Therefore, we implement UI elements to adapt the

Name: Show_And_Tell Type: Show and Tell

Target Selection: Stones.jpg

Model Selection: hsmainzcampus.nmodel

Figure 5.4: A *show & tell* AR nugget that was adapted using the ARNAUDDI by a person without programming knowledge. Left: Screenshot from ARNAUDDI executed on a desktop device. Right: Picture with ARNAUDDI running on the HHD. [Rau+22b].

3D objects' position and rotation immersively using the HHD. Additionally, we use LeanTouch to use touch controls, which are a common way to input data on mobile devices. Authors can move 3D objects by dragging their fingers when they tap and hold them on the HHD's screen. Using two fingers, rotation and scaling are possible in a similar manner.

Show & Tell AR Nugget

The default *show & tell* AR nugget (see Figure 5.3 top) uses a cube as a placeholder 3D object and implements one label with the text "This is a cube." The UI on the desktop computer lists all labels below the *Model Selection*. By typing a label text into one of the empty text fields and clicking "Add," authors can add a new label. Authors can also edit the text of existing labels using the text fields. The text size automatically scales to fit into the label. If there are multiple labels, authors can delete all except for one label using the "Del" button. The last label cannot be deleted because at least one label is mandatory for the AR nugget of the type *show & tell*.

Authors can select, move, and rotate labels similar to 3D objects. They can do so using the desktop computer or the HHD, and ARNAUDDI synchronizes the changes to the respective other device. A white dot at the label's bottom represents the label's anchor point. If authors click it, they draw a line to connect the label to the 3D object by clicking the point on the 3D object to which they want the label to attach. Then, a white line appears and connects the label and the 3D object. The point on the 3D object connected with the label can also be selected and moved.

Figure 5.4 shows one adapted *show & tell* AR nugget. An author without programming knowledge adapted it in 11 minutes using ARNAUDDI. The author stuck to the default image as the real-world anchor and replaced the placeholder 3D object with a 3D model of a building. Then, the author rotated the 3D model and labeled each block, e.g., "block A," "block B," and "assembly hall." Finally, the author moved and rotated the labels.

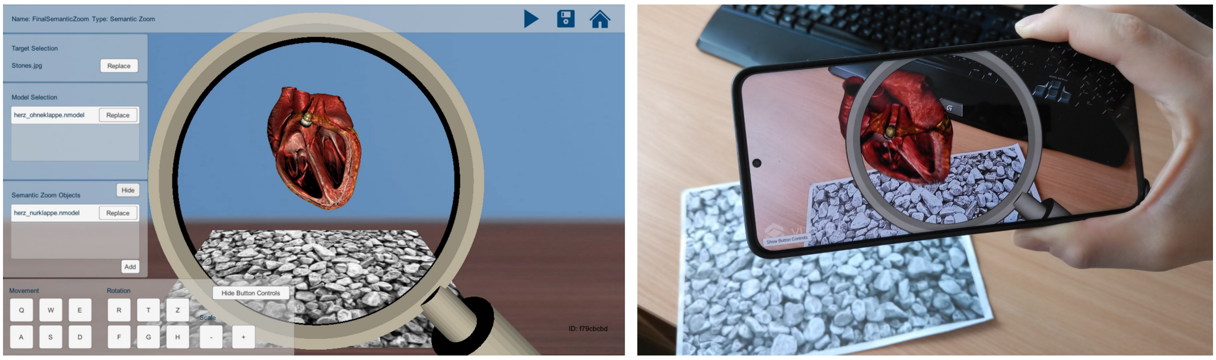


Figure 5.5: A *semantic zoom* AR nugget that was adapted using ARNAUDDI by a person without programming knowledge. Left: Screenshot from ARNAUDDI executed on a desktop device. Right: Picture with ARNAUDDI running on the HHD. [Rau+22b].

Semantic Zoom AR Nugget

The default AR nugget of the type *semantic zoom* (see Figure 5.3 lower left) uses a cube and an image with the text "additional information" on it as placeholder objects. While the cube is always visible when in the camera view, the image is only visible through the semantic zoom magnifying glass. The AR nugget already includes the magnifying glass. Authors can add one or more virtual objects that are only visible through the magnifying glass and replace or delete them. However, the system makes sure that at least one 3D object remains because it is mandatory for this type of AR nugget. The UI includes a button to hide the magnifying glass to make it easier for authors to select the 3D objects.

Figure 5.5 shows an adapted *semantic zoom* AR nugget that an author without programming knowledge adapted. Adapting this AR nugget took the author 11 minutes, including trying other ideas with other 3D objects first before deciding on the heart. The author replaced the placeholder cube with a 3D model of a human heart and added a 3D model of a stent with an aortic valve. While the heart is visible through the magnifying glass as well as beside it, the stent is only visible through the magnifying glass.

Quiz AR Nugget

The default quiz AR nugget (see Figure 5.3 lower right) uses a cylinder as a placeholder object for the correct answer and a cube for the wrong one. As a quiz question, it shows "Where is the cylinder?". Authors can edit the quiz question by clicking the "Edit" Button and typing text into the text field. When authors are in the preview mode and click (on the desktop computer) or tap (using the HHD) on the cylinder, it flashes in green color. The cube flashes in red color then.

Figure 5.6 shows one adapted *quiz* AR nugget. An author without programming knowledge adapted it in five minutes using ARNAUDDI. Similar to the *show & tell* AR nugget above, the author stuck to the default image as the real-world anchor. The quiz question asks "Which of

Name: Quiz Type: Quiz

Target Selection

Stones.jpg

Quiz Question

Welche der beiden Figuren stellt eine

Model "right"

tiamat updated.nmodel

Model "wrong"

green dragon adult updated.nn

ID: f79cbcbd

elche der beiden Figuren stellt eine babylonische Göttin dar?

Figure 5.6: A *quiz* AR nugget that was adapted using ARNAUDDI by a person without programming knowledge. Left: Screenshot from ARNAUDDI executed on a desktop device. Right: Picture with ARNAUDDI running on the HHD. [Rau+22b].

these two figures is a Babylonian goodness?" in German. The author defined the 3D object on the image's left side as the correct answer, so it will flash green if tapped or clicked on.

5.2 Integrated AR Nugget Authoring Tools

As introduced in the previous section, authors can use a stand-alone AR nugget authoring tool to adapt AR nuggets. However, if they want to implement something the AR nugget authoring tool does not support, developers must first implement these functions, re-deploy the authoring tool, and update it on the author's device or devices. An alternative for authors is to work directly with a Game Engine instead of using a stand-alone authoring tool implemented with a Game Engine. Typically, this requires programming knowledge. To allow adapting AR nuggets using a Game Engine but without requiring programming knowledge, we present AR nugget authoring tools that we integrate into a Game Engine environment. Still, authors with more experience can use the Game Engine's extensive functionalities.

5.2.1 Concept

Our authoring tools distinguish two phases for authoring location-specific AR nuggets, similar to Kampa and Spierling [KS17]. In the following, we first describe the authoring phases before we introduce the authoring tools. We differentiate between a process-specific and a location-specific phase, which we visualize in Figure 5.7. In the process-specific phase, authors choose and initialize AR nuggets that suit their ideas. They also adapt the parameters' values and replace placeholder objects. Thus, authors need a tool that allows them to switch between user and authoring modes with one click. With this, it allows to create and build an authoring application and a user application in the same authoring workflow. Typically, this first authoring phase is realized on a desktop computer. The location-specific phase takes place at the target location and is most suitable to be performed using an immersive AR device, preferably the target AR device. Here, authors can precisely fit the virtual objects and elements to the location in the real world. This includes adjusting and saving the AR nuggets' position, rotation, and scale. Using the

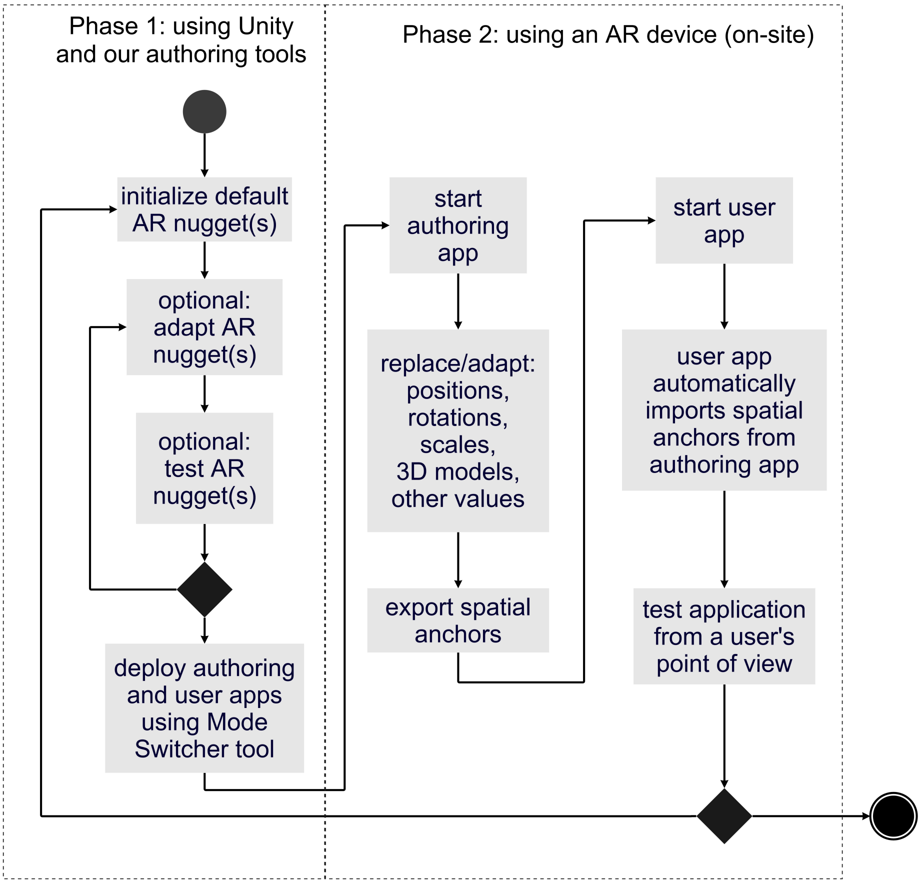


Figure 5.7: Flowchart of the twofold authoring process. In the first phase, authors concentrate on process-specific tasks. Using Unity and our authoring tools, they can adapt AR nuggets and deploy an authoring and user application. Adapting AR nuggets can include identifying and adding suitable 3D models, sounds, or other assets as well as determining the application flow with pre- and postconditions for the AR nuggets. In the second phase, authors focus on location-specific tasks like placing, scaling, and testing their adapted AR nuggets. Authors may repeat the two phases may in an iterative process. Based on [Rau+22a].

AR device, authors can directly see and inspect how the virtual objects fit and experience their adapted AR nuggets. Authors may iteratively repeat the two phases until they are satisfied with the AR nuggets. In the following, we introduce authoring tools that authors can apply and test in the first authoring phase and immersively use in the second authoring phase. Figure 5.8 provides an overview of these tools.

Placing Location-Specific Elements

When authors start an application, the AR device places virtual elements in its initial coordinate system. This coordinate system's orientation and point of origin depend on the AR device's position and orientation when the application is started. Therefore, the AR device renders virtual elements relative to its position and orientation at the start of the application. We visualize this in Figure 5.9.

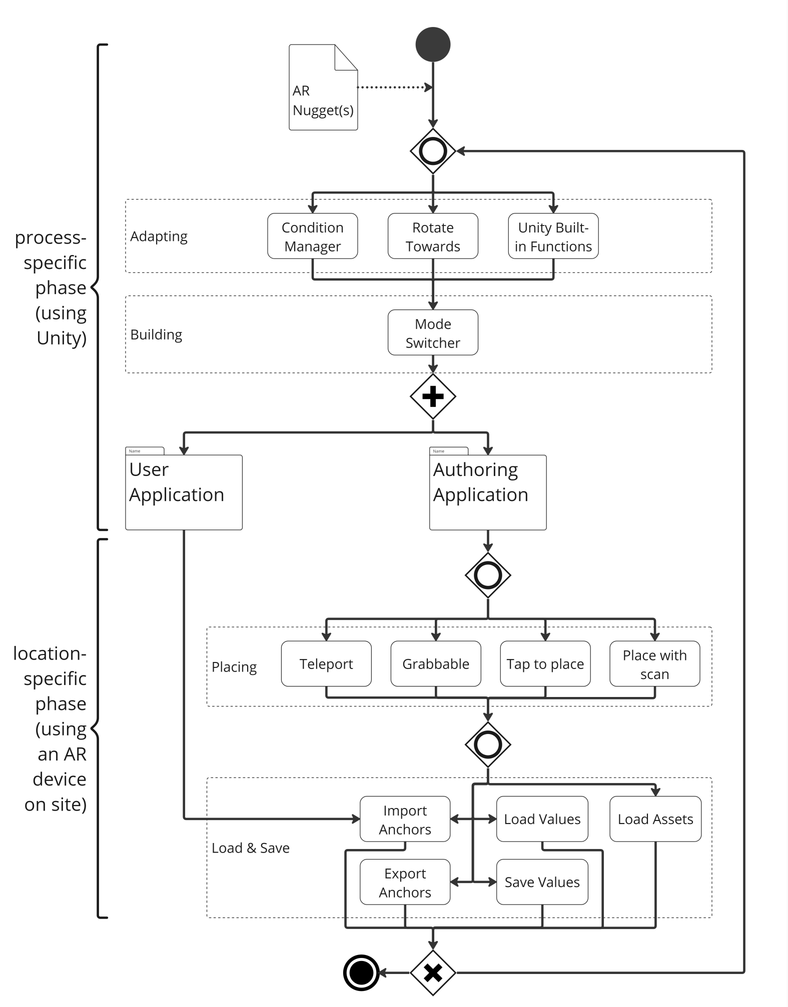


Figure 5.8: Overview of integrated AR nugget authoring tools. In the process-specific authoring phase, authors can adapt the AR nuggets using the *condition manager* and *rotate towards* tools as well as all Unity built-in functions, e.g., adjusting positions and scales. Using the *mode switcher* tool, they can build two separate applications based on the same Unity scene: one user application and one authoring application. Both applications can be used in the location-specific authoring phase. With the authoring application, authors can use one or more of the placement tools to place their virtual objects at the location. Then, the authors can ex- or import these positions as anchors or save and load values or assets. The user application automatically imports the anchors. All steps can iteratively be repeated.

However, location-specific AR nuggets must be placed and anchored in the real world, not relative to the starting position. This authoring step can only be executed on-site, during the location-specific authoring phase. Our AR nuggets include three different approaches for this authoring step: grabbing and placing virtual elements separately, teleporting AR nuggets, and moving all virtual elements using a reference scan of the real world. For each of these approaches, we introduce authoring tools that support authors in this step.

1) Grabbing and placing virtual elements individually: In our first option, authors can grab a virtual element and place it in the correct position in the real world. For this, we introduce a

Figure 5.9: Influence of the AR device's position and rotation on application start. Top: AR Nuggets are located in the building as the author intended. The coordinate system's origin is close to the room in the lower left corner. Bottom: The application was started while the AR device was not located where the intended origin of the coordinate system is. The AR nuggets have the correct position in this coordinate system but are not correctly placed in the building.

grabbable tool that allows authors to grab a virtual element and move, rotate, or scale it using one or two hands. To prevent accidental scaling when trying to move a virtual object using two hands, the option for scaling can also be deactivated by unchecking a box so that the virtual object can only be moved and rotated. We restrict the rotation to the up-axis to facilitate finding the correct position. However, authors can also deactivate this option to be able to rotate the virtual objects freely. Alternatively to the *grabbable* tool, we introduce a *tap to place* tool that authors can use to place virtual elements on real-world surfaces, e.g., a virtual pillar on the physical floor. In contrast to the grabbable tool, authors do not need to keep holding the virtual object while moving it.

Instead, they point a virtual ray from their hand to a virtual object, tap once to select it, and then the virtual object moves along with their hand movement. The virtual object moves on top or in front of real-world surfaces, i.e., its position is determined by where the virtual ray from the author's hand hits a surface. When the author taps again, the AR device places the virtual object in its current position.

2) Teleporting AR nuggets: Besides placing each virtual element individually, authors can also place AR nuggets by *teleporting*. Teleporting can ease placing when AR nuggets are initially not close to their desired location in the real world. With teleporting, author do not have to drag an AR nugget across a whole room. In some cases, AR nuggets could be invisible to the author when the application starts, e.g., when physical or other virtual objects occlude them or if the environment is complex and the author oversees the AR nugget. Then, *teleporting* an AR nugget in front of or close to the author can help authors to find the AR nugget's virtual objects.

3) Moving all virtual elements using a reference scan of the physical world: Another option is to automatically place AR nuggets using a scan of the physical environment where the AR device should place the AR nuggets as a reference (*place with scan*). Authors can use this scan to work on the virtual objects' positions, rotations, and scales already during the first authoring phase, so they only need to fine-tune them in the second authoring phase. Then, the authors need to correctly place, orientate, and scale the scan in the second authoring phase.

Authoring and User Mode

In the first authoring phase, authors must define which virtual objects are moveable so that their built application allows them to place them on-site during the second phase. However, only authors should be able to place AR nuggets and their virtual elements in the real world because it is an authoring task. Users should be restricted and have no access to the authoring functionalities. It is not feasible to include a button or other commands within the application to switch between access to authoring functionalities and no access because users could still access this, either by mistake or to change the application against the author's intentions. One solution is to develop different applications and restrict users to only using the intended application. However, in this case, authors would need to create an authoring application that includes authoring functionalities and an additional user application. Because manually creating and adapting two seperate applications would be cumbersome, we propose a *mode switcher* tool that automatically creates two seperate applications based on the same AR nuggets. The tool distinguishes an authoring mode from a user mode and allows authors to switch between these two modes with one click.

Anchoring Virtual Elements in the Real World

During the location-specific authoring phase, authors can save their real-world positions after the virtual elements are positioned in the authoring application. For this, we provide an *anchor* tool that anchors a virtual element at a specific position in the real world. The anchor tool stores the real-world position in form of representations of the geometries and colors in the room surronding the position. Our anchor tool can also save the positions of other virtual elements relative to the anchored element. For example, in a *show & tell* AR nugget where a virtual 3D model superimposes physical animal bones, the author might need to fine-adjust the labels' position after positioning the 3D model. In this case, the *anchor* tool saves the 3D model's position in the real world and the labels' positions relative to the 3D model. We save these positions and orientations in files with our *anchor ex-/import* tools. In combination with distinct authoring and user applications, it is mandatory to be able to export and import real-world anchors to exchange the real-world anchors between the applications. Authors create and export real-world anchors using the authoring application. When the user application is started, these real-world anchors need to be imported so that the virtual elements appear in the correct positions on site.

Save or Load Values and Assets

During the location-specific authoring phase, authors can also test their application and might realize that they need to adjust or fine-tune specific values. In this case, it would be tedious and time-consuming to go to the first authoring phase and re-deploy the whole application, only to change one small value. For example, finding the right speed for an avatar in a *navigation* AR nugget can be challenging using a desktop device and more suitable in an iterative process on site, where authors can easily and quickly test different values. Thus, we develop tools that allow authors to change parameters during the second authoring phase from within the application. Authors can then save the parameters to an exchangeable file using our *save and load values* tool. Different versions with different parameters can be saved and loaded with this tool. Besides parameters, we also introduce a *load assets* tool that allows exchanging an asset, e.g., exchanging one 3D model with another. These tools allow adaptions of the AR nuggets without the need to go back to the first authoring phase and to re-deploy the application.

Rotate Towards

We identified that there are several scenarios where authors want to orient a virtual object's rotation in relation to the user or other virtual objects. Therefore, we introduce a tool that applies billboarding to virtual objects. Authors can apply the *rotate towards* tool during the process-specific authoring phase. Using this tool, authors can, e.g., choose that a virtual object continuously rotates to the user. This could be useful for text to ensure it is always readable from any user viewpoint.

Condition Manager

Our AR nuggets implement all programming logic they require. For example, show & tell AR nuggets implement the function to show more details additionally to the labels on demand. However, some authors might want to add further functionalities of the type "if [event] occurs, then do [action]." They require a tool that supports this without requiring programming. In Section 3.5, we described an AR nugget manager where authors can select such events from a drop-down menu to start or end an AR nugget. However, in this case, authors might want to define actions other than starting or ending an AR nugget. For example, an author might want a virtual object to flash or become highlighted if the user approaches it. We introduce a *condition manager* tool that allows authors to select events and actions from a drop-down list during the process-specific authoring phase.

5.2.2 Implementation

Before we implement the authoring tools, we need to implement a template scene in Unity that serves as a base for authors. The template scene includes system functionalities like a camera or tracking functionalities. For the authoring mode, it adds an authoring menu that authors can use to start the ex- or import of spatial anchors. Authors can drag & drop default AR nuggets and our authoring tools into this scene. Furthermore, we implement our default AR nuggets using Unity and save them as Unity Prefabs. Prefabs are Unity-specific files that can store components, properties, and values. Authors can use the Prefab as a template and instantiate AR nuggets with drag & drop. We also implement the authoring tools as Prefabs in Unity so that authors can apply the authoring tools by drag & drop. For example, to make a virtual object grabbable, authors can use the *grabbable* tool and drag & drop the grabbable Prefab to the virtual object. That the authors can directly work within Unity has the advantage that the tools can easily be extended and updated. Additionally, experienced authors are not limited to using only the authoring tool's functions, e.g., when a stand-alone AR nugget authoring tool as described in Section 5.1 does not implement a desired functionality. We implement our authoring tools targeting the HoloLens 2, operating on an Universal Windows Platform (UWP). While most functions from the authoring tool also run on other devices, some functions, e.g., accessing the AR device's storage system, differ from device to device and might not be applicable to other devives. In the following, we briefly describe the implemented authoring tools.

Placing Location-Specific Elements

These tools support authors in placing AR nuggets or other virtual objects in their correct position in the real world and anchoring them there. The *grabbable* and the *tap to place* tools are based on the MRTK and implemented as prefabs that add the correct MRTK functionality to the virtual objects. However, the tools also include functionalities that ensure their correct application and execution. For example, to grab a virtual object, Unity must detect a collision. Unity can only detect collisions if a so-called collider attaches to the virtual object. Our tools check if the virtual objects fulfill such requirements. If they do not, the tools add the needed components. In this example, the tools would add colliders to the virtual objects. For the *grabbable* tool, we also implement an option that allows users as well as authors to grab and move a virtual object. Figure 5.10 visualizes how authors and users can use the *grabbable* tool.

Our *teleport* tool adds one button for each AR nugget to a menu during authoring phase 1. In phase 2, authors can press a button from the menu to make the associated AR nugget teleport in front of them. This can be combined with the *grabbable* or *tap to place* tool so that authos

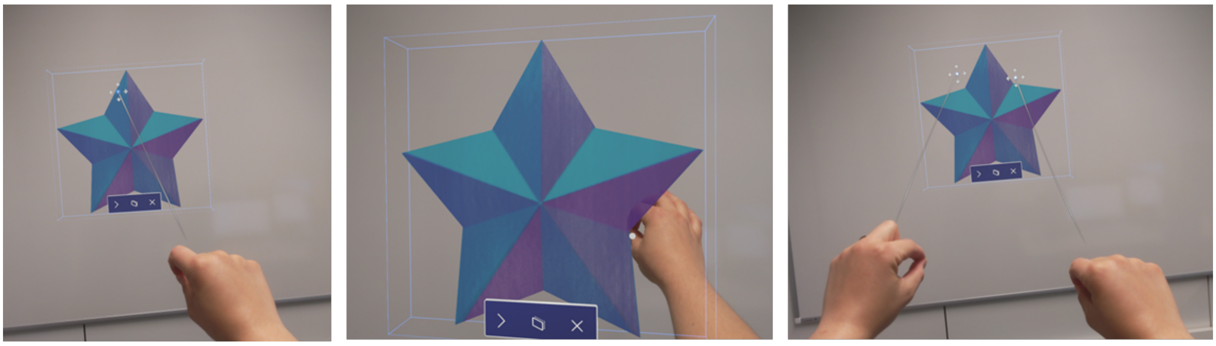


Figure 5.10: Placing a location-specific virtual object with the *grabbable* tool. Left: Grabbing the virtual object from a distance with a hand ray. Middle: Grabbing the virtual object directly with the hand. Right: Scaling the virtual object from a distance using two hand rays.

can quickly place AR nuggets with the *teleport* tool first, and then fine-adjust the AR nuggets' positions using one of the other two tools.

When using a scan of the environment as a reference, the author must place the AR nuggets relative to the scan in the first authoring phase. In authoring phase 2, the scan is mapped to the environment so that its orientation and position correctly overlay on the physical environment. The AR nugget system can execute the mapping process automatically if the scan is prepared as a Vuforia Area Target. However, automatic placement is not possible if the environment is challenging to track. We introduce a *scan placement* tool for these cases. Using this tool, authors can manually grab the scan, similar to using the *grabbable* tool, to correctly place and orient the scan to the real world. While the authors do so, the AR nuggets remain in their position relative to the scan. Therefore, the AR nuggets get placed in the physical environment as they were in the scan in the first authoring phase.

The *anchor* tool attaches a spatial anchor to the virtual object to allow saving the position even after closing the application. These anchors can be ex- or imported again using the *export* or *import* tool. The anchor tool communicates with the placement tools. When one of the placement tools moves a virtual object, it communicates this to the *anchor* tool. Then, if the virtual object has a spatial anchor, the *anchor* tool updates it.

Mode Switcher

The *mode switcher* tool allows authors to switch between authoring and user mode with one click, as visualized in Figure 5.11. We define whether certain functionalities are available in authoring mode, user mode, or both by default. For example, authors typically use the *grabbable* tool to place AR nuggets during the location-specific authoring phase and want to restrict users from moving the AR nuggets. Thus, we enable the grabbable tool by default for only the authoring mode. However, authors can change this and make AR nuggets or other virtual objects grabbable for users, too, with one click. Furthermore, we implement a *select parent mode* tool that allows authors to define in which mode certain functionalities or virtual objects are available or not. For example, an author may want to see a semi-transparent virtual 3D object to place it precisely

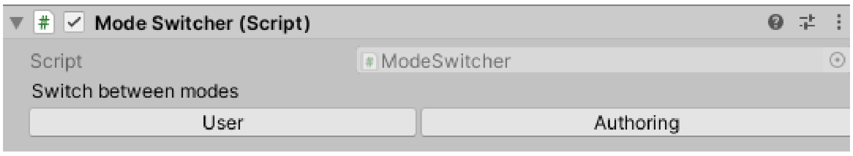


Figure 5.11: Screenshot of the *mode switcher* tool in the Game Engine Unity. Authors can switch between user and authoring modes using one click.

in the right location, yet wants the user to see the 3D object without transparency. In this case, the author can define the semi-transparent 3D object as only available in authoring mode and the other 3D object as only available in user mode. When the author switches modes using the *mode switcher*, the *mode switcher* automatically activates the 3D objects for the chosen mode and deactivates the ones not available in the selected mode.

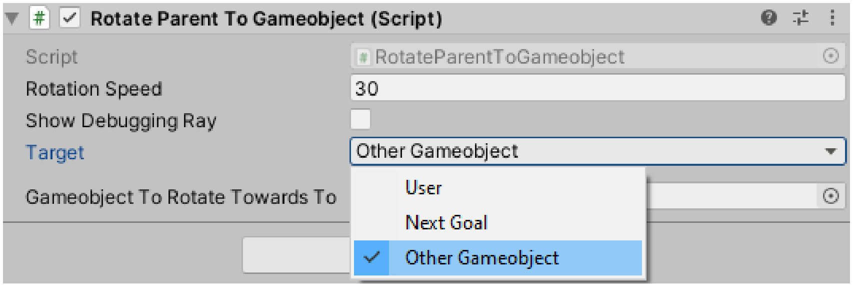
Ex- and Import Spatial Anchors

We use spatial anchors from the MRTK to anchor AR nuggets in the real world. The *export* tool allows to export the spatial anchor to a binary file. Additionally, it exports virtual object positions relative to the spatial anchor in a text-based file. We create a save system to access the AR device's file storage. Using this, we store the file on the AR device. This file can be stored and reaccessed later or by another application. Authors can also copy the file from one AR device to another to share spatial anchors with multiple AR devices. Using the file, the *import* tool can import the information about the spatial anchors and automatically position AR nuggets and their spatial anchors accordingly. Authors can also configure to import spatial anchors automatically when an application starts. For example, an author could create and export spatial anchors using an application configured for authoring mode with the *mode switcher*. Then, if a user starts an application configured for user mode with the *mode switcher*, the previously created and exported spatial anchors can be automatically imported.

Save or Load Values and Assets

We create functions that allow writing the AR nuggets' parameters and values to a JSON file. We base these on Unity's built-in functions for JSON data, called JsonUtility, which allows creating a JSON string from values. Our save system writes the JSON string to a file and saves it on the AR device. Similar to saving, our system can also read files and load the data into the application.

We also develop a *load assets* tool to allow loading assets like 3D objects, images, or videos and replacing an existing asset in an application with the newly loaded one. To realize this, we utilize Unity's assetBundle functions. An assetBundle is a unity-specific file format that can store one or more assets. Authors can use Unity to export their 3D objects or other assets as assetBundle. Then, they can copy the assetBundle to the target AR device and replace an asset within the AR application.



☑ Rotate Parent To Gameobject (Script)

Script: RotateParentToGameobject

Rotation Speed: 30

Show Debugging Ray ☐

Target: Other Gameobject

Gameobject To Rotate Towards To

Figure 5.12: Screenshot of the *rotate towards* tool in the Game Engine Unity. Authors can choose from a drop-down list where a virtual object should continuously rotate to. If authors choose to rotate the virtual object towards another virtual object, the field "Gameobject To Rotate Towards To" appears, and authors can drag & drop the other virtual object there.

Rotate Towards

Authors may drag & drop this tool's Prefab to any virtual object. Then, they can choose if the virtual object should rotate towards any PoI using a drop-down list. The drop-down list includes rotation to the user and rotation to a pathway's current node for *navigation* AR nuggets. The authors can also define custom objects to rotate to.

Condition Manager

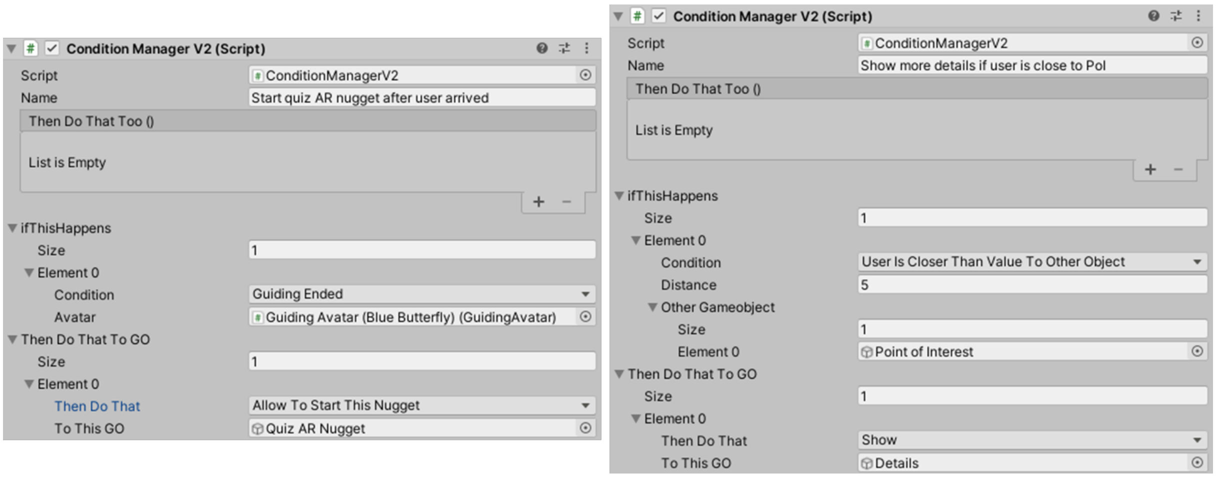
We implement the *condition manager* with drop-down menus from which authors can choose conditions. The available conditions are similar to pre- and postconditions for AR nuggets described in Section 3.5. The events triggered when the conditions are met can also be chosen from a drop-down list. We also implement a Unity event that allows more experienced users to define any event, even when the event is not available from our list. For this, authors drag & drop a virtual object to our implemented event and select any event that Unity provides.

5.3 Constraint-based Authoring with AR Nuggets

The user's physical environment is not always known in advance. In such cases, authors cannot define specific PoIs to anchor virtual objects to. Instead, authors can define constraints surfaces must fulfill to have virtual objects positioned there. For example, an author can specify that a virtual object needs a minimum surface area of two square meters on a wall. The constraints allow an AR application with AR nuggets to adapt to fuzzy conditions, which makes it applicable to different locations. This section describes an authoring approach where AR nuggets can place virtual objects based on constraints that authors can define.

5.3.1 Concept

AR devices re-construct a virtual scene based on the information they gather about their real environment. Computer algorithms can categorize the surfaces found in the environment by their


Condition Manager V2 (Script)

Script	ConditionManagerV2
Name	Start quiz AR nugget after user arrived
Then Do That Too ()	
List is Empty	
ifThisHappens	
Size	1
Element 0	
Condition	Guiding Ended
Avatar	Guiding Avatar (Blue Butterfly) (GuidingAvatar)
Then Do That To GO	
Size	1
Element 0	
Then Do That	Allow To Start This Nugget
To This GO	Quiz AR Nugget

Condition Manager V2 (Script)

Script	ConditionManagerV2
Name	Show more details if user is close to PoI
Then Do That Too ()	
List is Empty	
ifThisHappens	
Size	1
Element 0	
Condition	User Is Closer Than Value To Other Object
Distance	5
Other Gameobject	
Size	1
Element 0	Point of Interest
Then Do That To GO	
Size	1
Element 0	
Then Do That	Show
To This GO	Details

Figure 5.13: Screenshot of the *condition manager* tool in the Game Engine Unity. Authors can choose conditions and actions that are executed if the conditions are met using drop-down menus. Left: The author defined that if the guiding from the blue butterfly avatar has ended (i.e., the user arrived), then the quiz AR nugget may start. Right: The author defined that if the user is closer than five meters to a PoI, then the virtual object "Details" is shown.

purpose, e.g., floor, wall, ceiling, or more precisely like closet, chair, or table. The AR device can then use this information to automatically place virtual objects on a suitable surface. In our constraint-based authoring approach, the author places virtual objects in any environment, and the AR nuggets can then automatically calculate constraints. The AR nuggets recognize on which category of surfaces the virtual objects are placed, and, based on this information, calculate category constraints. Additionally, the surface's area or the distance between the user and the surface can be further constraints. For example, a virtual object can be constrained to be placed on a desk with at least two square meters or closer than one meter to the author.

To automatically calculate the constraints, we rank all surfaces that were detected in the room by their distances to the author and their surface areas. Surfaces with a surface ares close to zero can be classified as noise and filtered out. For the surfaces that are not filtered out, we calculate the surface area and distance ranks by normalizing the values for distances to the author and surface areas on ranges from 0 to 1.

To calculate constraints for the placement of one virtual object, the AR nugget first checks the ranks of the surface the virtual object was placed on for extrema. If the surface's surface area has the largest rank among the surface areas, the constraint is set to 'largest surface'. Otherwise, if its surface area has the smallest rank, the constraint is set to 'smallest surface'. If the surface area has neither the highest nor the lowest rank, the AR nugget checks if the surface has the smallest or largest distance rank and sets the constraints to 'largest distance' or 'smallest distance'.

For virtual objects that are not placed on the largest, smallest, closest, or furthest surface, the AR nugget decides for a surface area constraint if the surface area rank is larger than the distance rank. If the surface area rank is above average, the AR nugget sets the constrait to "at least [surface area value]", and for surface area ranks below average to "at most [surface area value]". Otherwise, if the distance rank is larger than the surface area rank, it decides for a distance constraint with "at least [distance value]" for distance ranks below average and "at most [distance value]" for distance ranks above average. In the case of identical ranks, AR nuggets can use a decision hierarchy. For example, an author is in a room where the physical surfaces have surface areas between four and 0.5 square meters. The surface with the largest surface area has a rank of 1 and the one with the smallest has a rank of 0. The author places a virtual object on a physical desk with a surface area of one square meter. With one square meter surface area, the desk is a rather small surface, but not the smallest one and it has a surface area rank of 0.3. Its distance rank is 0.2 and smaller than its surface area rank. Therefore, the AR nugget decides for a surface constraint. As the surface area rank is smaller than the average, the AR nugget applies "at least 1 m^2 surface area" and surface type desk as constraints.

When the constraints are calculated and users start the application in any environment, the AR nugget places the virtual objects based on the constraints. If there is no suitable surface in the environment that fulfills all constraints, a fallback is needed. Otherwise, if the virtual objects cannot be placed at all, the user may not be able to experience the application as intended by the author. Here, our idea is to set a tolerance threshold and lower it until the AR nugget can place all required virtual objects in the environment. The AR nugget can also use similar categories if required, e.g., it can place a virtual object on the floor if no desk is present.

5.3.2 Implementation

To categorize the surfaces in the real world, we use the scene understanding SDK and implement our prototype with the Game Engine Unity and the MRTK. The scene understanding SDK of the MRTK detects rough surfaces of objects and returns a high-level, abstract representation of a scene. It categorizes them into one of the following categories [Mic]:

- Wall: an (immovable) wall of a room
- Floor: surfaces, on which one can walk (floor of a room, ramps, multi-level floors, etc.)
- Ceiling: the upper surface of a room
- Platform: large, flat surfaces (like tables, countertops, beds)
- Background: an object, which is non of the above (doors, seats and chairs), can be a vertical surface as well as a horizontal surface
- Unknown: not yet classified (can be classified in a later iteration, when more data of this object is present)

Our proof of concept prototype implements an author mode and a user mode. When the user starts the app, it scans its environment and initializes the scene understanding. It highlights all detected surfaces in different colors, each surface category in another color. Then, scene understanding updates every 25 seconds to adapt to changes, like a chair that has been moved, and to improve the quality by gathering further information.

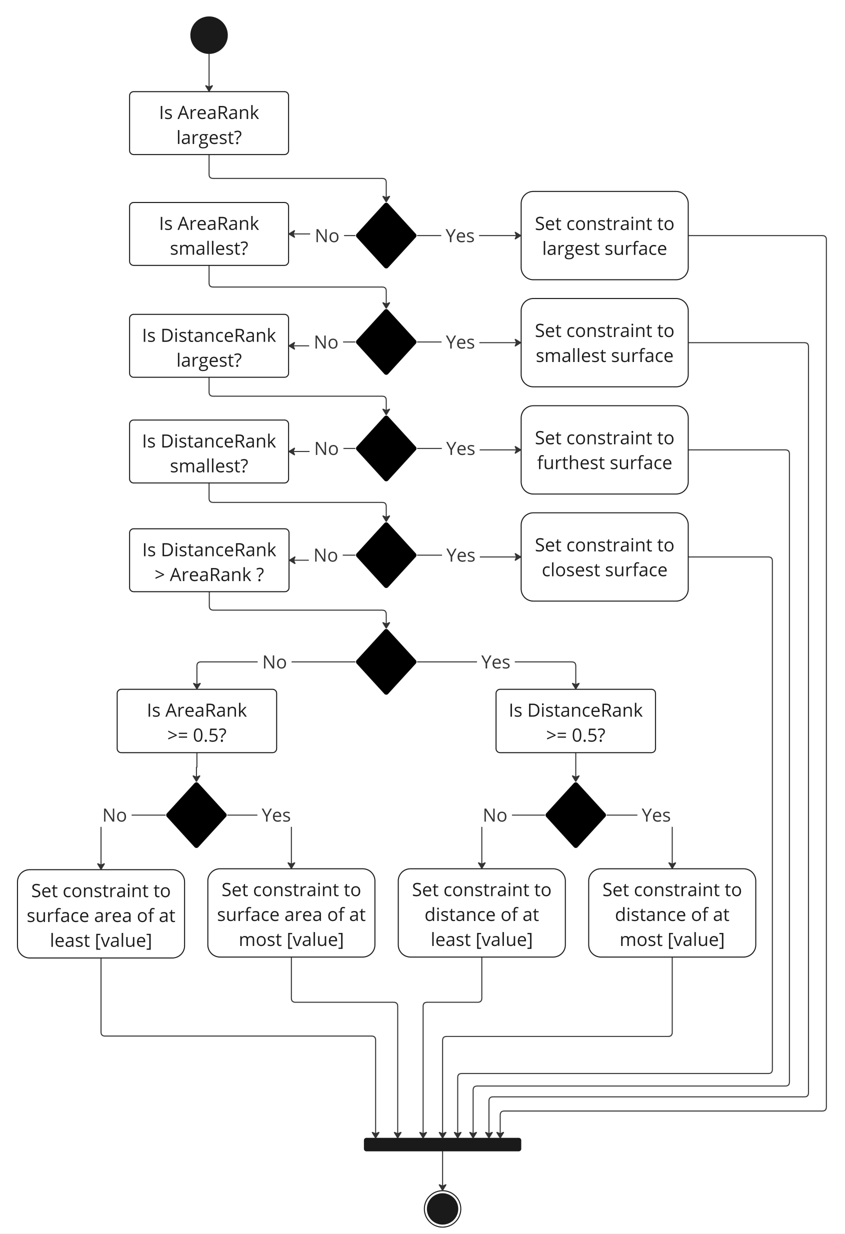


Figure 5.14: Automatic calculation of surface and distance constraints based on rankings.

In the author mode, authors arrange an AR nugget's virtual objects in their current environment. First, the AR nugget places the virtual objects in the scene in front of the author. Then, the author can pick any of them up and place it on any surface by tapping it, moving it with the hand, and letting it go by tapping again. As soon as the user places a virtual object, the AR nugget checks on which category of surfaces it was placed. When the author finished placing the virtual objects and clicks a button, the AR nugget calculates the surfaces' distance and surface area ranks and calculates the constraints. We apply a threshold of 0.05 square meters to consider surfaces for the surface ranks and to exclude smaller surfaces. If the surface is smaller, the application computes no rank and interprets the surface as noise. The AR nugget saves the constraints in form of the virtual objects' surface category, as well as surface area and distance ranks of the surfaces.

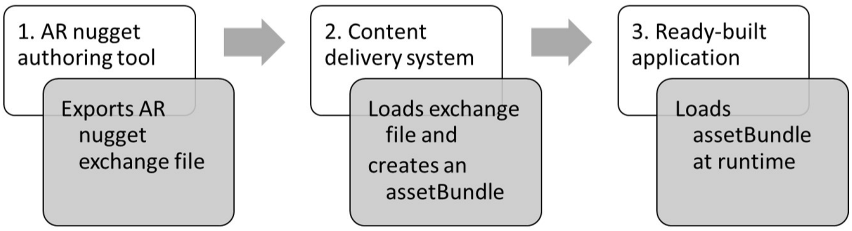


Figure 5.15: [Rau+22b] Toolchain for on-demand AR content delivery to a ready-built application.

Figure 5.14 visualizes how our prototype calculates constraints. If a surface is the largest, smallest, closest, or furthest one, our prototype uses this extrema as a constraint. Otherwise, if the rank is greater than 0.5 on the normalized range, the prototype sets a minimum constraint, i.e., the virtual object must have at least this distance or surface area. If the rank is smaller than 0.5, the value is set as a maximum constraint, i.e., the virtual object's distance or surface area must be smaller than the value.

After the constraints are calculated, the author can go to another room and switch to user mode. Then, the AR nugget places the virtual objects in the room based on the constraints. Our prototype does not implement further tolerances, a fallback, or relative placement.

5.4 Content Delivery

Although AR nuggets are stand-alone AR applications that include a runtime environment, authors may want to integrate AR nuggets into existing applications or combine multiple AR nuggets in one application in some cases. For example, authors could want to integrate AR nuggets that target to enhance an education course within a learning application and a learning management system, similar to work about VR nuggets [Hor+21a]. In such cases, it is essential that authors can update, replace, or add further AR nuggets integrated into existing applications. To target these challenges, we develop and implement a toolchain that we visualize in Figure 5.15.

Our authoring tool from Section 5.1 can export each AR nugget to a separate file using an AR nugget exchange format. The AR nugget stores a unique name, the type of pattern it reflects, its virtual objects, a representation of a path to the real-world anchor, and its nugget-specific parameters in a text-based file. If AR nuggets include constraints to place their virtual objects as described in Section 5.3, it also stores the constraints.

Moreover, we develop a content delivery tool that we integrate in Unity. With the tool, authors can import the AR nuggets from the AR nugget exchange format to Unity. Based on the AR nugget exchange file, the content delivery tool rebuilds a Unity scene. Authors can then use the Unity built-in functions to build an application from the AR nugget. Additionally, authors with programming experience may further adapt the AR nugget using Unity. Using the content delivery tool, the Unity scene can then be exported as an assetBundle. An assetBundle is a Unity-specific file format that applications can load dynamically at runtime and contains assets like 3D models, images, or videos. Finally, the assetBundle with the AR nugget can be loaded into existing ready-built applications at runtime without the need to rebuild and re-install the whole application.

Chapter 6

Evaluation

6.1 AR Nuggets

In an expert user study with physicians and medical experts, we investigate whether the AR nugget concept that we introduced in Section 3.1 and implemented in Section 4.1 is applicable in the context of CME. The study assesses the experts' willingness to engage in short AR experiences. Additionally, we assess how well they accept comparable VR nuggets, allowing us to compare AR and VR nuggets for CME. The subsequent subsection outlines the procedural aspects of the user study. Following that, we show the study's outcomes and subsequently discuss them.

6.1.1 Expert User Study

Our group of experts comprised six unpaid volunteers. Four were physicians, and the other two were medical journalists. Two of the doctors were ear-nose-throat specialists, one was a dermatologist, and one was a neurology specialist. The participants ranged in age from 52 to 61 years old, with Ø 56.33 years and SD 4.03 years. Because of the covid-19 situation then, we conducted the study online via video calls, each lasting between 40 and 60 minutes. We started the video calls by welcoming the experts and informing them about the topic. The latter included a brief explanation of AR and VR and a short introduction to a CME website that gives users access to our nuggets. Following this, we showed a video of a short prototype CME course with AR and VR nuggets. As the video played, we described the actions in the video. The video showed how a prototype CME course educated a doctor about the skin disease actinic keratosis. It was similar to the one described in Section 4.1 and Figure 4.2, but shortened to only include the *show & tell* AR nugget describing the skin's structure and the *progression* AR nugget animating the disease. It also included additional VR nuggets, which were combined to show a tour of a doctor's office of an ear-nose-throat specialist. In this prototype, the first step to start each AR nugget or the VR nuggets is to scan a QR code from a website. The video demonstrates this procedure. After the video, the experts commented on the video or asked questions, which we answered. This part of the interviews took 20 minutes. Finally, we asked the experts to respond to an online survey, which took them 10 to 20 minutes to complete. The survey questionnaire asked general questions about the CME course shown in the video and the participants' assessment of AR and VR for CME.

It included the following questions, where participants could answer Q1 - Q6 on a 7-point scale ranging from 1 (negative answer) to 7 (positive answer). The questionnaire asked questions Q1 - Q4 in two versions, one targeting AR (Q1.A - Q4.A) and one for VR (Q1.V - Q4.V). Q5 and Q6 targeted general aspects regarding the CME course. Next, participants could answer Q7 - Q11 with yes, no, or within free text fields. Then, the participants rated their prior experience with AR/VR using a 4-point scale (Q12.A/V) ranging from 1 (no experience) to 4 (experienced) and listed their CME experiences and preferences. At the end of the survey, the questionnaire asked if there was anything the participants particularly liked, disliked, or had thoughts on before it collected demographic data.

Q1 How do you rate the added value through the AR/VR experience?
Q2 How much can you imagine using AR/VR in your future CME?
Q3 Are you willing to buy a new AR/VR device?
Q4 When you choose your next CME course, how determinant is it if it includes AR/VR content?
Q5 How do you rate the integration of AR/VR into the CME course?
Q6 How clear was the CME course's procedure?
Q7 Do you think some CME courses suit the use of AR/VR better than others?
Q8 Why do you think some CME courses do or do not suit the use of AR/VR better than others?
Q9 What courses or topics can you imagine?
Q10 Would you install an official app to participate in a course with AR/VR elements?
Q11 What obstacles do you see for the use of AR/VR in CME?
Q12 How do you rate your experience with AR/VR?

6.1.2 Expert User Study Analysis

Table 6.1 summarizes the results to the questions Q1 - Q6 and Q12. Additionally, Figure 6.1 visualizes them in a box plot. We hypothesize that the experts evaluate our approach positively with a value greater than four (as four equalizes a neutral value on the 7-point scale). For each of the items Q1 - Q6 rated on a 7-point scale, we performed a Wilcoxon test with a predetermined threshold for statistical significance set at 5%.

With Q1, the participants rated the added value from the nuggets statistically significantly positively, with a probability value of $p = 0.0128$ for AR (Q1.A) and $p = 0.0328$ for VR (Q1.V). Comparing AR and VR in Q1, three of the six experts rated the added value from AR higher than the added value from VR with one, two, and three points more. Two participants rated the added value with the same points for AR and VR, and one rated VR with one point more than AR.

In Q2, all except one expert could imagine using AR/VR for their future CME. However, the Wilcoxon test revealed no statistically significant difference from the results of Q2 to our neutral value of four ($p = 0.0972$ for Q2.A and $p = 0.2887$ for Q2.V). Half of the experts rated Q2 with the same value for AR as for VR, while the other half rated one, two, and three points more

Table 6.1: Expert user study results.

Scale	Question		Ø	SD
7-point	Q1.A	added value	6.17	0.90
7-point	Q2.A	future use	5.33	2.13
7-point	Q3.A	new device	5.67	1.38
7-point	Q4.A	prefer course with AR	4.83	1.34
7-point	Q1.V	added value	5.33	1.11
7-point	Q2.V	future use	4.33	1.80
7-point	Q3.V	new device	4.00	1.83
7-point	Q4.V	prefer course with VR	4.17	1.07
7-point	Q5	integration VR/AR content	6.00	0.58
7-point	Q6	clear procedure	6.83	0.37
4-point	Q12.A	experience with AR	1.50	0.50
4-point	Q12.V	experience with VR	1.83	0.37

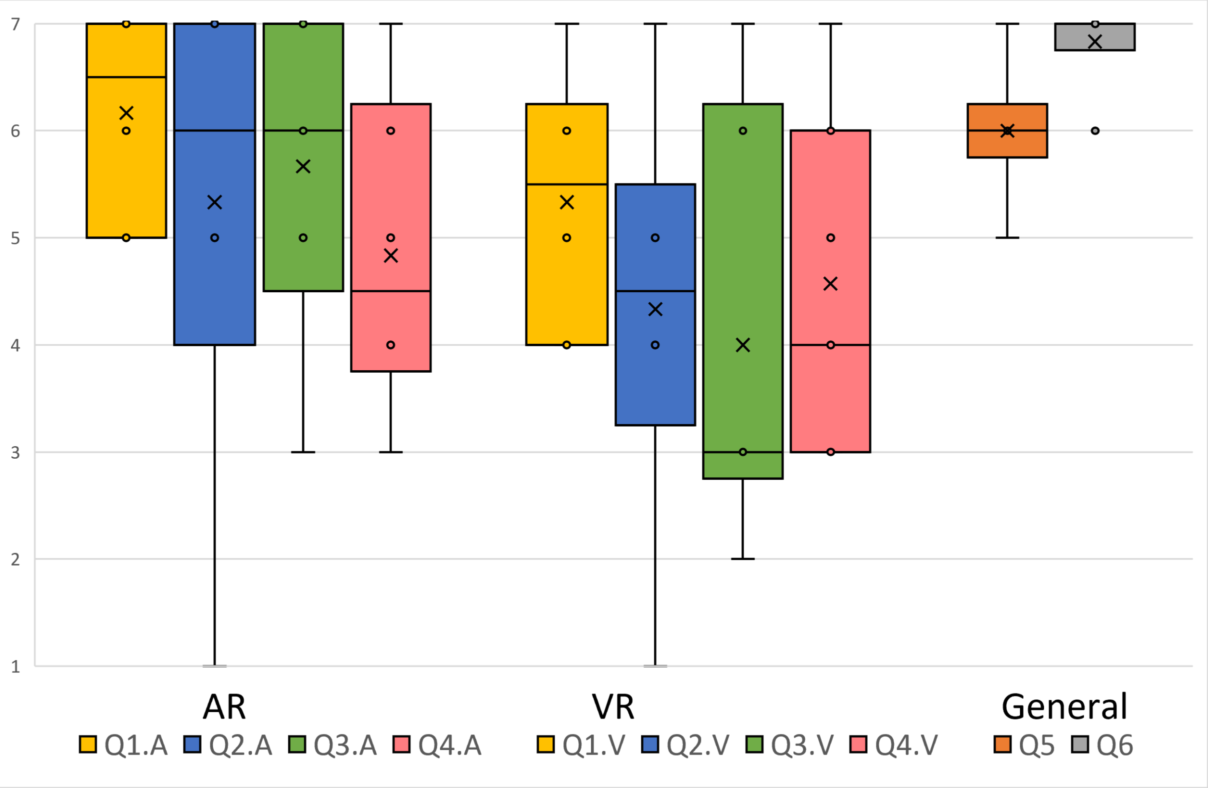


Figure 6.1: Results for Q1 - Q6 and Q12. [Rau+21].

for AR than for VR. Of all participants, one person could not imagine using AR or VR for future CME courses because, as described in a free text field, elderly persons typically have little or no technical expertise and could not accept or not want to use this technology.

One person was not willing to buy a new device to participate in AR-enhanced CME courses (Q3.A), while the other participants were willing and rated Q3.A with a value of five. The Wilcoxon text showed a statistical significance for a positive value in Q3.A with p = 0.0285. For VR, it was not conclusive with p = 0.5. Half of our participants rated Q3.V with three points lower than Q3.A, one participant one point lower, and the other two rated Q3.A and Q3.V similar.

The results for Q4 do not show statistically significant differences to the neutral value of four (Q4.A with p = 0.0987 and Q4.V with p = 0.3527). Comparing AR and VR, half of the participants rate Q4.A with one or two points higher than Q4.V.

In general, the experts rated the nuggets' integration within the CME courses as statistically significant positively (p = 0.0118). Also, they found the procedure clear (Q6), which was significantly rated positively with more than 4 points and p = 0.0098.

All participants in our study thought that specific CME courses are more suitable for taking advantage of AR/VR than other CME courses (Q7). In Q8, they explained that this depends on the course's topic and that the added value is most significant if 3D structures are shown, such as in anatomy, or if the technology lowers risks and costs, such as when laser treatments are demonstrated. As additional course topics or use cases, the experts proposed patient information, product demonstrations, and operations (Q9). The results from Q10 show that all of our participants were willing to install an app on their smartphones to participate in future AR or VR-based CME. In Q11, the participants listed production costs, data privacy, making the course interactive, and the need to purchase VR glasses as barriers. One expert articulated the concern that a course must avoid superficiality and instead offer in-depth knowledge to enhance the expertise of medical specialists.

In Q12.A, five participants reported having minimal experiences, while one has never used AR. Similarly, one of the participants had never used VR, and the others have used VR a few times (Q12.V). The participants' area of expertise is the medical field. They have worked in the medical business for many years and participated in various CME programs. All participants have visited congresses for CME, four have participated in CME through magazines, and five have also used other digital forms (Q13). Regarding the preferred form of CME, half of our participants favored traditional congresses, while the other half leaned more toward online resources (Q14). Finally, two of our participants stated that they see more potential for AR than VR in CME. One participant added that he could not imagine any use case for VR in his discipline (Q15).

6.1.3 Expert User Study Discussion

The results obtained from our questionnaire indicate that the experts interviewed see a higher degree of additional value in AR nuggets for CME compared to VR nuggets, as evidenced by their responses to Q1. Similarly, Q2 was also rated more positively for AR than for VR. This may contribute to the higher willingness to purchase a new AR device than a new device for VR (Q3.A/V). However, smartphones or tablets that can be used as AR devices also serve various other functions in people's daily lives. The fact that the majority of our experts are willing to purchase a new smartphone or tablet to use the AR nuggets (Q3.A) and that they would prefer a course with AR nuggets over a traditional course (Q4.A) demonstrates their interest in our AR nuggets. Based on this, we conclude that our AR nuggets can help a CME course become more attractive. The questions (Q1.A/V) to (Q4.A/V) all received more positive responses for AR nuggets than for VR nuggets. Consequently, we conclude that our experts find AR nuggets more appealing than VR nuggets for the use case of CME.

Incorporating AR nuggets into CME courses received positive ratings (Q5). This suggests that our technique stated in Section 5.4 to provide the AR nuggets to a ready-built app and to launch an AR nugget by scanning a QR code is adequate and clear to our experts (Q6). As highlighted by a participant, an essential challenge occurs in creating AR and VR nuggets to provide accurate and precise data tailored to the needs of medical specialists. Thus, we think it is critical to enable specialists with medical knowledge but few or no programming skills to create their own AR and VR content for CME courses. This can ensure the accuracy of the course's medical content without making the medical experts dependent on IT specialists.

6.2 Utilization of Tangible Interactions

This section evaluates and compares the tangibles that we introduced in Section 3.3 and implemented in Section 4.2 in a user study. It especially analyzes if and why users prefer one tangible over another for specific types of AR nuggets. Here, we implemented *show & tell* and *progression* AR nuggets to use on a Samsung Galaxy S20+. We adapted both AR nugget types to augment both tangible types, realistically shaped and generic, with a 3D model of skin or vertebras. This results in eight AR nuggets that we show in Figure 6.2 and describe in the following.

The AR nuggets A - D are of the type *show & tell* AR nugget A annotates a virtual cross-section of human skin that is augmented to a realistically shaped tangible. AR nugget B annotates two virtual vertebras with a virtual spinal disk that are augmented to a realistically shaped tangible. AR nugget C and D augment the generic tangible with a cross-section of skin and vertebras. With these four *show & tell* AR nuggets, we aim to give users an understanding of the 3D models' structures.

The *progression* AR nuggets show the same virtual objects, but without labels, and instead with an animation showing disease development. With these, we aim to support users in understanding the diseases' developments. AR nugget E shows an animation that visualizes the development of a skin disease called actinic keratosis. The virtual model anchors to a realistically shaped tangible. AR nugget G anchors it to the generic tangible. In AR nuggets F and H, we visualize the development of disk herniation, again once with the virtual object anchored to a realistically shaped tangible and once to the generic tangible.

Additionally to the *show & tell* and *progression* AR nuggets for generic and realistically shaped tangibles, we implemented one AR nugget for our combined tangible that we described in Subsection 3.3.3. When this AR nugget detects the generic tangible without another tangible plugged onto it, it instructs the user to plug the vertebras tangible on top of it. When it detects the vertebras tangible on top of the generic tangible, it augments the vertebras tangible with the corresponding virtual vertebras model.

6.2.1 User Study

Our user study involved 11 voluntary, unpaid participants. The participants were between 22 and 63 with Ø 32.45 and SD 13.44 years old. We ensured sufficient lighting conditions to ensure

Show & Tell AR Nuggets

Realistically Shaped Tangibles — Skin, Vertebras

Generic Tangible — Skin, Vertebras

Sequence AR Nuggets

Realistically Shaped Tangibles — Skin, Vertebras

Generic Tangible — Skin, Vertebras

Figure 6.2: AR nuggets that participants in the user study explored. Top: *show & tell* AR nuggets (A - D). Bottom: *progression* AR nuggets (E - H). Left AR nuggets (A, B, E, F): realistically shaped tangibles for skin and vertebras. Right AR nuggets (C, D, G, H): generic tangible.

good tracking quality for each user. At the beginning of the user study, we welcomed the users and informed them about the topic. We divided the participants into two pseudo-randomized groups. Group A included 6 participants and started with the generic tangible. Group B included 5 participants and started with the realistically shaped tangibles. The rest of the user study was similar for both groups.

Before starting an AR nugget, we informed users about the learning goals of the eight small applications and that they can end each one whenever they think they explored everything. Then, we placed the tangible in front of the user and started the first AR nugget on the smartphone. We visualize the study's order of AR nuggets and questions in Figure 6.3. Additionally, Table 6.2

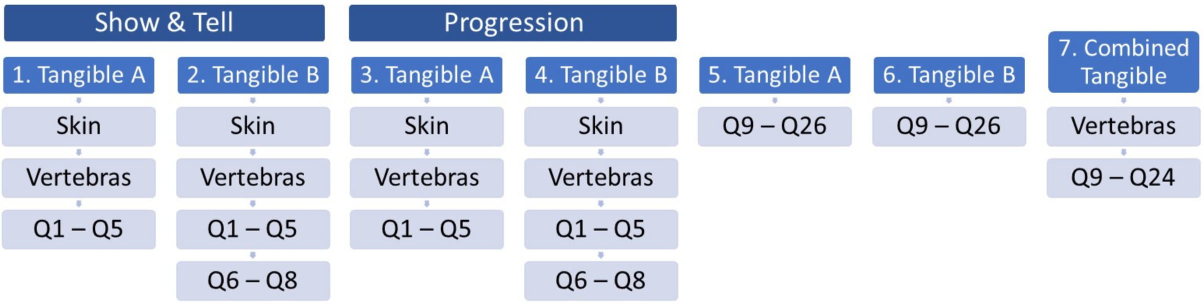


Figure 6.3: Procedure of the user study. Users started with tangible A and the *show & tell* AR nugget about skin, followed by the *show & tell* AR nugget about vertebras and the questions Q1 - Q5. For group A, tangibles A were the realistically shaped tangibles, and tangible B was the generic tangible. This order was switched for group B, so the groups started with different types of tangibles. After using both tangible types and answering Q1 - Q5 for both, the users answered Q6 - Q8. Then, we repeated the procedure with the *progression* AR nugget. Next, users answered Q9 - Q26, which targeted questions about the tangibles without regard to the AR nuggets. Finally, the users tried the combined tangible.

summarizes all questions from the questionnaire. Answers were given as free texts or on a 7-point semantic differential scale with word pairs. One exception was Q8, where users had to choose between the options tangible A or tangible B. Q17 - Q24 were word pairs from the short version of the UEQ questionnaire [SHT17] that measures the tangibles' pragmatic and hedonic quality aspects.

Finally, the questionnaire asked the users about their experience and collected demographic data. On average, the user study took one hour with each user.

6.2.2 User Study Analysis

We summarize the questionnaire's results in Table 6.3 and visualize the questionnaire's results for Q1, Q2, and Q4 as a box plot in Figure 6.4. Additionally, we analyze statistical significance for the questions on a 7-point scale using the Wilcoxson test with a threshold for statistical significance of 5%. The tests revealed no statistically significant differences between the tangible types or the AR nuggets.

For both AR nuggets, users thought they reached the learning goal (Q1) with both tangible types. From our users' point of view, both tangible types contributed to achieving the learning goal (Q2). For the *show & tell* AR nugget, users rated the realistically shaped tangibles' contribution higher than the generic tangible's contribution and the other way round for the *progression* AR nugget (Q2). For the realistically shaped tangibles, five users described that the tangibles' realistic shape supported the understanding and that the haptic feedback was helpful for the learning process (Q3). For the *progression* AR nugget, three users stated that the tangible type does not matter or matters little because the user focuses on the AR device's screen with the animation and not on the tangibles or their shape (Q3).

Table 6.2: Questions and answer possibilities from the questionnaire

Number	Question	Word Pairs
Q1	I think I have reached the learning goal	not - well
Q2	The tangible(s) contributed ... to reaching the learning goal	not - much
Q3	Why did the tangible(s) contribute, or why did it/they not?	
Q4	Handling smartphone and tangible(s) was ...	challenging - easy
Q5	Why was handling the smartphone and tangible(s) challenging - easy?	
Q6	What advantages for tangible A and disadvantages for tangible B do you see?	
Q7	What disadvantages for tangible A and advantages for tangible B do you see?	
Q8	If you were free to choose, what tangible would you decide for?	
Q9	The tangible's/tangibles' weight was ...	too light - too heavy
Q10	The tangible's/tangibles' size was ...	too small - too large
Q11	It was ... to realize that virtual objects anchored to the tangible(s)	impossible - easy
Q12	What did you like about the tangible(s)?	
Q13	What did you dislike about the tangible(s)?	
Q14 - Q21	I find the tangible(s) ...	(UEQ-S [SHT17])

Users found handling the realistically shaped tangible (Q4) easy. Handling the generic tangible (Q4) was also rated as easy. However, two users explained they needed time to adjust to the handling (Q5).

One user rated a value of two for handling the generic tangible and stated that it was impossible to touch what she saw (Q5). Furthermore, she stated that she found moving the tangible easier than moving the smartphone (Q5). She explained that it is easier to understand how to move the tangible to view it as she wants to than to think about how she would need to move the smartphone (Q5). Three users critized that labels are too small or cover parts of the 3D model (Q5). Two users stated that the generic tangible was easy to rotate using its handle (Q5). For the realistically shaped tangibles, three users stated that rotating was challenging at some points (Q5). Additionally, they explained to like the generic tangible's tripod functionality because it was easy to view the augmentations from all sides, including from the bottom (Q5).

In the *show & tell* AR nugget, 10 of our 11 participants described the similarity between realistically shaped tangibles and virtual models as an advantage (Q6/7). Five users stated that the generic tangible was easier to move and rotate than the realistically shaped tangibles (Q6/7). Four users stated that the generic tangible's tripod was especially helpful in viewing the animation because the tangible stood still, and the viewing angle was good (Q6/7). Eight users would use the realistically shaped tangibles for the *show & tell* AR nugget, and three would decide on the generic tangible (Q8). In contrast, for the *progression* AR nugget, seven users would decide to use

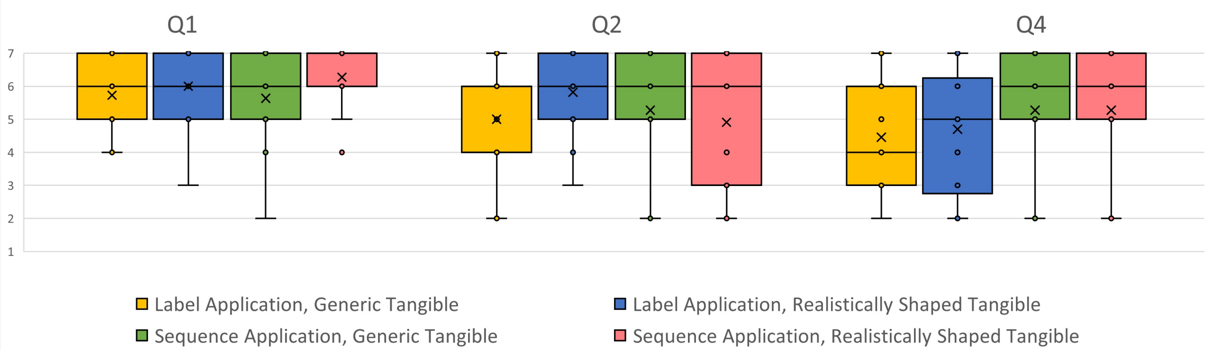


Figure 6.4: Results for Q1, Q2, and Q4 from our questionnaire. Each box plot shows the results for the *show & tell* and *progression* AR nugget with the tangible types realistically shaped and generic.

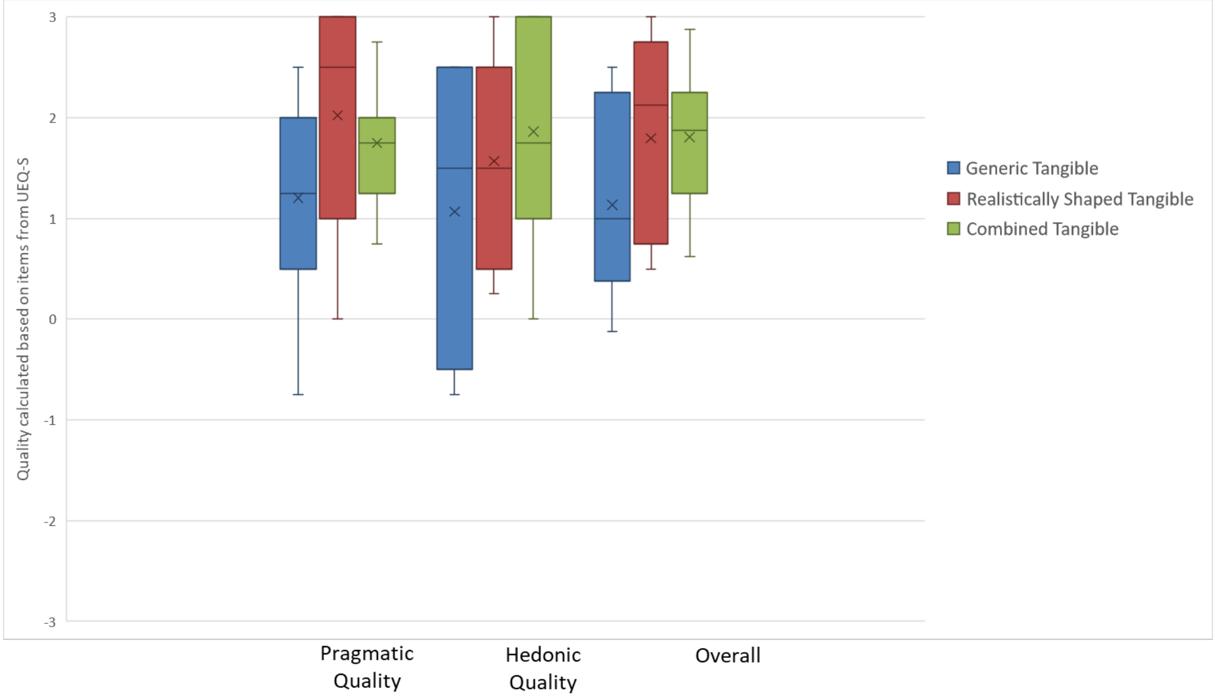


Figure 6.5: Results from the UEQ-S for each tangible type.

the generic tangible and four the realistically shaped tangible (Q8). Both tangible types were, on average, rated as neither too light nor too heavy (Q9) and neither too small nor too large (Q10).

Two users described that they especially liked the combined tangible (Q12). In Q13, two users reported a jerky tracking with the vertebras tangible.

We visualize the results of the UEQ-S (Q14 - Q21) for each tangible type in Figure 6.5. On the answer scale from the UEQ-S, the most negative value is -3, and +3 is the most positive value.

In pragmatic quality, the generic tangible was rated with Ø 1.20, SD 0.98, and the realistically shaped tangibles with Ø 2.02, SD 1.05. Both values show a positive pragmatic quality, meaning our participants perceived both tangible types as supportive, easy, efficient, and clear to use. The combined tangible's pragmatic quality was rated at Ø 1.75, SD 0.63 and thus lies between the generic and the realistically shaped tangibles' pragmatic quality.

Table 6.3: Results from the questionnaire to the 7-point scale questions

Question		AR nugget	tangible type	Ø	SD
Q1	learning goal	*show & tell*	realistically shaped	6.00	1.26
Q1	learning goal	*progression*	realistically shaped	6.27	1.01
Q1	learning goal	*show & tell*	generic	5.73	1.19
Q1	learning goal	*progression*	generic	5.64	1.50
Q2	tangibles' contribution	*show & tell*	realistically shaped	5.82	1.33
Q2	tangibles' contribution	*progression*	realistically shaped	4.91	2.12
Q2	tangible's contribution	*show & tell*	generic	5.00	1.79
Q2	tangible's contribution	*progression*	generic	5.09	2.07
Q4	handling	*show & tell*	realistically shaped	4.73	1.79
Q4	handling	*progression*	realistically shaped	5.27	1.79
Q4	handling	*show & tell*	generic	4.45	1.81
Q4	handling	*progression*	generic	5.27	1.79
Q9	weight		realistically shaped	3.91	0.54
Q9	weight		generic	3.91	0.30
Q9	weight		combined	4.09	0.30
Q10	size		realistically shaped	4.18	0.60
Q10	size		generic	3.64	0.92
Q10	size		combined	3.90	0.30
Q11	connection		realistically shaped	6.36	0.92
Q11	connection		generic	6.18	1.54
Q11	connection		combined	6.27	1.10
Q14 - Q17	UEQ-S: pragmatic quality		realistically shaped	2.02	1.05
Q14 - Q17	UEQ-S: pragmatic quality		generic	1.20	0.98
Q14 - Q17	UEQ-S: pragmatic quality		combined	1.75	0.63
Q17 - Q21	UEQ-S: hedonic quality		realistically shaped	1.57	1.01
Q17 - Q21	UEQ-S: hedonic quality		generic	1.07	1.25
Q17 - Q21	UEQ-S: hedonic quality		combined	1.86	1.07

Differences between both tangible types were smaller for their hedonic qualities. The participants rated the realistically shaped tangibles with a hedonic quality of Ø 1.57, SD 1.01, and the generic tangible with Ø 1.07, SD 1.25. The combined tangible's hedonic quality was rated the highest value of the three tangible types with Ø 1.86, SD 1.07. This means our users perceived the combined tangible as the most exciting, interesting, inventive, and leading-edge one.

6.2.3 User Study Discussion

Our study indicates that each tangible type contributed to the learning experience, but depending on the type of AR nugget one tangible type could be more suitable than the other one (Q2). In the *show & tell* AR nuggets, it is important that users view the virtual object from all sides to reach the learning goal and understand the virtual object's structure. For this, the realistically shaped tangibles are supportive because their realistic shape and realistic haptic feedback support users. For the *progression* AR nuggets, the focus was on the animation displayed on the AR device's screen rather than on any 3D components. Here, the generic tangible contributed slightly more to the learning experience than the realistically shaped tangibles. Users preferred the generic tangible for the *progression* AR nuggets because it is possible to place it still on a surface, and the tangible's shape is of secondary importance. In contrast to placing a realistically shaped tangible on a surface, the generic tangible is not lying directly on the surface. This way, users could still view the virtual content anchored to the generic tangible from all sides, including the bottom. This illustrates that a tripod handle supports one-handed moving and rotating interactions in handheld AR.

In the UEQ-S (Q14 - Q21), the participants rated the realistically shaped tangibles with the highest pragmatic quality, although there is no statistically difference in pragmatic quality between the tangible types. However, this indicates that the tangibles' realistic shape can be helpful for user interactions. The pragmatic quality is lower for the generic tangible, although users stated that they especially liked its handle and tripod functionality. This indicates that a realistic shape contributes more to the pragmatic quality than a handle and tripod do. The pragmatic quality of the combined tangible was rated less than the generic tangible. One reason could be that users usually needed a few seconds to realize how they could plug the tangibles together. However, combined tangible's hedonic quality with Ø 0.30 is the highest hedonic quality of the three tangible types.

The results of our study indicate that requirements for a suitable tangible depend on the type of AR nugget. Our users described that they prefer a realistically shaped tangible if 3D interactions like viewing an object from all sides are central to the AR application. However, using a tangible similar to its virtual augmentation in size and shape is not always possible. For example, if the tangible is too small or has little to no significant features for tracking, the AR nugget cannot detect and track it. Additionally, users can only comfortably grab it if it has a suitable size and weight. Lastly, creating realistically shaped tangibles can be cumbersome and expensive as each use case requires another shape. Our users stated to prefer the generic tangible if the AR application requires users to hold the tangible still, like for watching an animation in our *progression* AR nugget, . Users can hold the tangible on its handle or place it on a surface using the tripod. Our combined tangible joins these advantages and adds interactivity through the plug connection system to an AR nugget. This and the results from the UEQ-S show that the combined tangible is a good compromise between tangibles with realistic shapes and tangibles that support one-handed interactions.

6.3 Integration of User Assistance in AR Nuggets

This section evaluates the AR nuggets' virtual assistance system from Section 3.4 that we implemented in Section 4.3. The evaluation was carried out with a user study conducted as a student project. It analyzes how helpful the assistance functions are from a user's point of view, if they are understandable, and how they contribute to helping users understand how to work with an AR nugget. Additionally, it investigates how users perceive the timing of when the hints appear and disappear.

6.3.1 User Study

Our study involved 10 voluntary, unpaid participants (six male, four female). They are between 21 and 54 years old with Ø 29.30 years, SD 9.89 years. On a scale from 1 (daily use) to 5 (no experience so far), they rate their experience with AR applications with Ø 4.10, SD 0.88. Before the study, we prepared a desk and placed an image target on the desk. We checked the lighting conditions to light the image target and made sure that the image target was clearly visible. At the beginning of the study, we welcomed the participants, explained the study's procedure, and described that the prototype augments an image target and aims to assist in using it. Then, we placed the participants approximately two meters in front of the desk and handed them a smartphone with our prototype started. We advised the participants to test the prototype as they liked it and for how long they liked it. The image target could not be detected and tracked from the participants' initial position because the distance was too far. Thus, participants could decide to approach the desk with the image target and point the AR device to it. If they did not come to the idea of pointing the AR device to the image target and coming closer, the application triggered a hint that explained how to detect the target. The participants continued to use the prototype and triggered hints while doing so. If participants announced that they were done with testing the prototype but had not triggered all hints yet, we advised them to perform actions that triggered the remaining hints so that they could provide feedback for all hints. For example, if participants successfully initiated tracking at the beginning before our prototype displayed any hint, we advised them to point the smartphone away from the image target (so that tracking was lost) and to wait until the hint showed.

After testing the prototype, we asked the participants to fill out a questionnaire with 30 questions that could be answered on a 5-point scale. The first part of the questionnaire asked Q1 - Q4, targeting all implemented hints. Next followed a part that asked Q5 - Q9 for each of the hints for distance, motion, touch, video, and both stages of the tracking hint. The first stage of the tracking hint also included the above questions, except for Q6, because the hint does not have an icon. Instead, the prototype shows a ghost view in the second stage. The part for the video hint only included Q5 and Q7.

Q1: The assistant hints within the application were helpful. [1 agree - disagree 5]
Q2: I understood the textual cues and information. [1 agree - disagree 5]
Q3: I find the combinations of textual cues and icons useful. [1 agree - disagree 5]
Q4: The meaning of the icons was clear. [1 agree - disagree 5]
Q5: The hint, that [description of the hint] was helpful. [1 agree - disagree 5]
Q6: The combination of textual cues and icons was helpful. [1 agree - disagree 5]
Q7: I knew what to do after seeing the hint. [1 agree - disagree 5]
Q8: The hint was visible for too long. [1 agree - disagree 5]
Q9: The point of time when the hint showed was ... (1 too early - too late 5)

The questionnaire's final part asked if the participants had additional comments with the option to write them in a free text field (Q10). Also, it asked for the participants' ages, gender, and experience level with AR applications.

Table 6.4: Results from the user study about the virtual assistant.

	Q1		Q2		Q3		Q4	
Hint Type	Ø	SD	Ø	SD	Ø	SD	Ø	SD
All Hints	1.80	0.92	1.20	0.42	1.20	0.63	2.00	1.05

	Q5		Q6		Q7		Q8		Q9	
Hint Type	Ø	SD	Ø	SD	Ø	SD	Ø	SD	Ø	SD
Distance	1.10	0.92	1.70	0.95	1.20	0.42	3.60	1.35	3.60	1.35
Movement	1.50	0.32	2.10	1.20	1.40	0.97	4.30	1.25	2.80	1.23
Touch	1.60	0.85	1.90	1.20	1.50	1.08	4.40	0.84	3.10	0.57
Video	1.20	1.08	-	-	1.10	0.32	-	-	-	-
Tracking Stage 1	1.30	0.63	-	-	1.00	0.00	4.40	0.70	3.10	1.10
Tracking Stage 2	1.40	0.68	1.10	0.32	1.00	0.00	4.40	0.84	3.60	1.43

6.3.2 User Study Analysis

Table 6.4 shows the outcomes for each of the questions Q1 - Q9.

For this analysis, we divide the users into two groups: one group with no or minimal experience with AR (users who rated their experience with 4 or 5, where value 5 equals no experience) and one group with a bit more experience (users who rated their experience with a value of 3). There were no participants who rated their experience with a value of 1 or 2. For questions Q1 - Q9, a Mann Whitney U test revealed no statistically significant differences between both groups.

Overall, the participants perceived the hints as helpful (Q1) (Ø 1.80, SD 0.92), and participants were able to understand them (Q2) (Ø 1.2, SD 0.42). They found the textual cue and icon combinations helpful (Q3) with Ø 1.20 and SD 0.63. All except one participant, who rated this with a value of 3, rated this with 1. For most participants, the icons' meaning (Q4) was clear (Ø 2.0, SD 1.05).

The individual hints were all rated helpful (Q5) with values of Ø 1.10 to Ø 1.60. All participants rated all hints with value 3 (neutral) or more helpful, except for one participant on the touch hint. In the free text comment (Q10), he states that he found the touch hint rather confusing because if there was no hint, it would have been clear to him that the application did not implement any touch interactions. One more participant also states this in Q10.

The participants rated the combination of textual cues with icons (Q6) as helpful. Here, the combination for the tracking hint at stage 2 was rated as most useful with Ø 1.1, while the others were rated with slightly lower values, ranging from Ø 1.70 to Ø 2.10. Two participants rate the combination for the movement hint with a value of 4, meaning they do not agree to find the combination useful. These two, and also one more participant, describe that they would prefer a stop sign as known from road traffic over the sketched icon in the free text form (Q10).

Our participants stated to know what to do after seeing the hint (Q7) for all hints. For Q7, all participants rated both tracking hints with a value of 1. The video hint is rated with 2 by one participant and with 1 by all other participants. The participant who rated it with 2 explained that it was not intuitive to him how to close the video. The participants rated values of Ø 1.20 to Ø 1.50 for the distance, movement, and touch hint.

The participants did not find the duration during which the application showed a hint too long (Q8). However, one participant fully agreed with the statement that the duration was too long for the distance hint. This participant rated his experience with AR application with 3.

The participants found the timing of the touch hint's appearance (Q9) was on the spot with Ø 3.10, SD 0.57. In contrast, they perceived the timing of the appearance for the other hints (Q9) mixed. For the distance hint, two participants found it too early, two on time, and five too late. The other hints were rated similarly mixed, which results in a larger SD for Q9 than for most other questions. Two participants explicitly describe that they would have preferred to see the second stage of the tracking hint at an earlier point in time (Q10). Except for the movement hint, the more experienced group tended more to rate the hints' timing as too late than the less experienced group did.

6.3.3 User Study Discussion

Overall, the virtual assistant was perceived positively. This suggests that including assistant functions can improve the user experience for AR nuggets. With a critical look, we also see room for improvement. Our touch hint, which was perceived as least helpful, was triggered directly after a single touch interaction when the user tried to use touch input where it was not possible. Instead, another option could be to show the hint only after the user repeatedly tried to use touch interaction, e.g., after 5 attempted touch interactions within 10 seconds. Then, participants would not trigger this hint if it already became clear to them that no touch interactions are implemented.

Our results show that the combination of textual clues and icons is perceived as helpful. Still, there is room for improvement in the icons' design and their connection to the text, specifically for the movement hint and its stop sign icon. The participants, including the ones who never used AR before, understood what to do after seeing the hints. This shows that the hints can communicate how to use an AR application. The hints' duration was rated as a bit too short. To improve this, the virtual assistant can be configured to display the hints longer. Another idea could be to show the hints for a longer time when they are displayed for the first time and for a shorter duration when triggered repeatedly because then the user could be able to remember the hint and need less time to read and understand it. Similarly, the timing could be configured to display hints earlier when the user just started the application, with more time between the hints as the time spent in the application increases.

6.4 Usage of Multiple AR Nuggets in Complex AR Setups

In this section, we evaluate how applicable AR nuggets and our integrated AR nugget authoring tools are without programming from the point of view of two media designers. The media designers used AR Nuggets and our integrated authoring tools from Section 5.2 to create

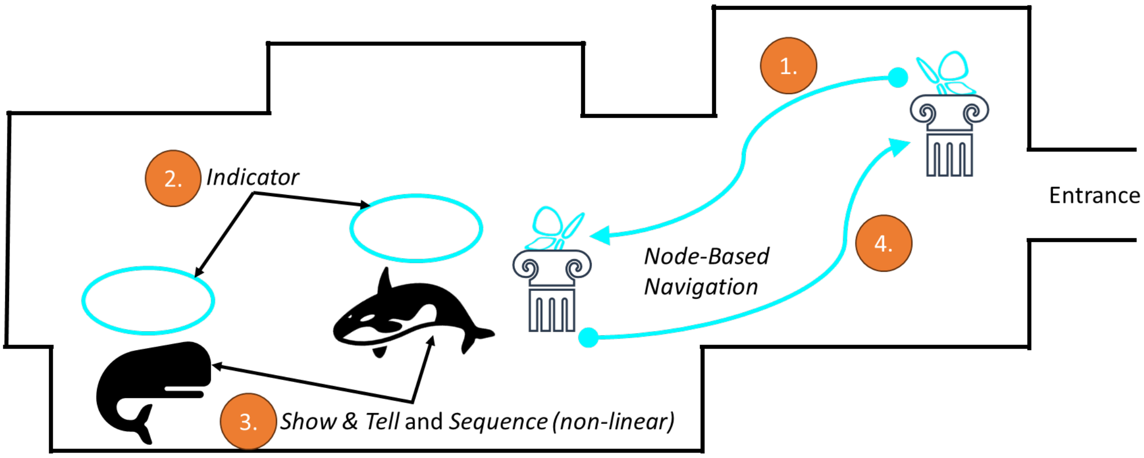


Figure 6.6: The media designers' draft for the museum visitor journey. At the entrance, museum visitors receive the HoloLens 2 and meet a butterfly, which guides them to the whale exhibition with a *node-based navigation* AR nugget. Here, an *indicator* AR nugget shows visitors which exhibits they can explore with the AR application: For both whales, the visitors can experience a *show & tell* and *progression* AR nuggets in any order. Whenever the visitors want to, the butterfly can guide them back to the entrance, again with a *node-based navigation* AR nugget, where they return the HoloLens 2.

a complex AR application for a whale exhibition in a museum for natural history, with the HoloLens 2 as the target device. This shows that AR nuggets are suitable for assembling one larger AR application in a complex setup. In the following, we describe the authoring process from their point of view.

The media designers started by developing a concept for a museum visitor journey and put that onto paper in the form of a storyboard. While less experienced authors could also start with experiencing our default AR nuggets, these two media designers were already familiar with the default AR nuggets. They thus had some knowledge about the possibilities of AR. Figure 6.6 visualizes their idea for the museum visitors' journey. The storyboard plans that museum visitors receive the HMD on a starting point. At the starting point is a pillar, on which a virtual blue butterfly rests. The butterfly serves as a guide and brings the visitors to the entrance of the whale exhibition, where it arrives on another pillar and rests there. After arriving, the application augments two virtual circles to the floor, close to the two whale exhibits the application can augment. A virtual welcome board informs the visitors that they may step into the circles. If they step into one of the circles, one of the AR nuggets that augments the whale exhibits starts. The AR nuggets annotate the physical bones or illustrate what the whales have looked like and how they hunt for food. When the visitors want to leave the exhibition, they can approach the pillar with the butterfly to have the butterfly guide them back to the tour's starting point.

After drafting this plan, the media designers planned what virtual elements were needed, e.g., 3D models of whales, whale sounds, and a 3D model for the butterfly. Furthermore, they checked if the existing collection of default AR nuggets provided AR nuggets that were suitable for their concept. Because we worked closely with the media designers together, the required default AR nuggets were available. The two media designers chose to create two pathways using

AR nuggets for node-based navigation. One navigation AR nugget brings the visitor from the museum's entrance to the whale exhibition, and the other one gets the visitor from the exhibition back to the entrance. Additionally, they chose an indicator AR nugget. For both of the whales, they also chose a *show & tell* and a *progression* AR nugget each.

Then, the media designers dragged the required default AR nuggets into the template Unity scene to start adapting the AR nuggets. Our default template scene includes an authoring menu with buttons to ex- or import spatial anchors and the mode switcher. Next, the media designers replaced the default virtual objects with their self-created 3D models including animations, added their self-created sounds, and adapted label texts. To have a reference for the whales' sizes and distances from each other, they scanned the museum's exhibition room using the HoloLens 2, copied this 3D roomscan to their computer, and added it to the Unity scene. Utilizing the 3D roomscan supported them in adapting the virtual elements' overall positions, rotations, and scales. Additionally, they added a function to toggle the scan on or off to the authoring menu. Finally, they defined the conditional behavior of the AR nuggets by choosing pre- and postconditions for the AR nuggets from a drop-down menu.

Then, they could test the resulting application in Unity's Play Mode, where user input can be simulated. The Play Mode helps to check if the AR application functions as intended and to detect potential logical mistakes in the conditional behavior. When the AR application performed as intended, they used the Mode Switcher Tool to deploy one AR application for visitors and one for authors to the HoloLens 2.

Then, the media designers were ready to start the location-specific authoring phase. They started the application in authoring mode on the HoloLens and positioned the virtual elements so their virtual 3D models augmented the physical whales precisely. For a quick, rough positioning, they grabbed the virtual 3D model of the roomscan and positioned it. All virtual objects moved together with the scan and were placed roughly in the correct place. Then, they used the authoring menu to hide the scan. This allowed fine adjustments for the virtual elements' positions because they could grab and move each virtual element individually using the *grabbable* tool. When satisfied with the positions, they exported the spatial anchors to a file using the spatial anchor ex-/import tool that we had included in the template scene. Then, they started the application in user mode on the same HoloLens 2, where the HoloLens 2 automatically imported the exported spatial anchors. Once the HoloLens 2 completed the import process, they could test the application from a museum visitor's point of view.

Completing the AR application was an iterative process. After the location-specific authoring phase, another process-specific authoring phase follows to finetune the conditional behavior. A location-specific authoring phase followed again, but the spatial anchors from the first iteration could be reused to make the positioning task less effortful. The media designers iteratively repeated the authoring phases until they did not want more iterations.

6.5 Combination of AR Nuggets and VR Nuggets

This section evaluates how transitions can combine and connect AR nuggets with VR nuggets as described in Section 3.6 and implemented in Section 4.5. For this, we carried out a user study. The following subsections describe the user study and analyze and discuss its results.

6.5.1 User Study

We divide the user study's participants into the groups A and B. In total, 20 unpaid volunteers participated, i.e., 10 per group. Group A experienced the AR and VR nuggets, including the implemented transitions. Group B served as a control group and switched HMDs when they liked without us initiating a transition. The participants in group A were aged between 21 and 64 years (Ø 29.80, SD 11.81), while group B was similary aged from 21 to 59 years old (Ø 29.00, SD 10.81). Regarding their experience with AR and VR, the groups were heterogeneous. Group A rated their experience with AR Ø 3.80, SD 2.14, and with VR Ø 4.40, SD 1.69 on a scale from 1 (no experience) to 5 (daily use). On the same scale, group B rated their experience with AR as Ø 2.00, SD 1.61, and with VR as Ø 2.00, SD 1.48. For each participant, the study procedure took about 45 minutes. We conducted the study in an office environment which we prepared by placing miniature skeletons of dinosaurs and a whale in the room. When the participants arrived, we welcomed and informed them about the study's topic. Additionally, we explained that the following experience could also occur in a museum; thus, the miniature skeletons represent real dinosaurs and whale skeletons. Furthermore, we explained how to interact when wearing the HMDs.

Next, we started with the user study's task phase. The participants put the AR glasses on as instructed, and we started the first AR nugget. The first AR nugget did not include an intro transition or a questionnaire. We asked users from group A to inform us before they wanted to switch HMDs. When they informed us, we initiated the first outro transition. In contrast, users from group B switched HMDs as they liked. Then, they took the AR HMD off and switched to the VR HMD.

For group A, we started the VR nugget and triggered an intro transition once the participants had the VR HMD put on. After up to 15 seconds, the intro transition ended, and a questionnaire appeared. Group B directly saw the questionnaire within the VR nugget environment after putting the VR HMD on. Then, participants from both groups explored the VR nugget as they liked and how long they wanted to. Following this, they answered a questionnaire using the controllers of the VR HMD. This questionnaire comprised the short version of the AttrakDiff questionnaire [HBK03] and SPES [Har+16], in total 18 items.

After participants answered the questionnaire, we triggered the next outro transition for group A, and both groups switched the HMDs again. We repeated this process until the participant explored three AR and three VR nuggets, connected with five intro and five outro transitions. In AR, the participants answered the questionnaire using hand gestures.

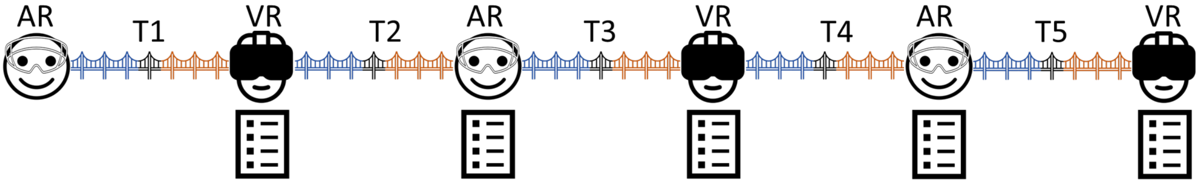


Figure 6.7: Design of the user study for combinations of AR and VR nuggets. After experiencing the first AR nugget, the transition T1 transitioned the user to the next VR nugget. We varied which transition we assigned to T1 - T5. The transitions T1 - T5 each consist of an outro transition, the reality, and a similar intro transition, as indicated by the bridge's colors. Within each AR and VR nugget, except for the first one, the participants answered a questionnaire before the next transition started. [Rau+23].

After experiencing the AR and VR nuggets, the participants completed a post-study questionnaire with the following five items. They could answer the questions on a Likert scale from 1 to 7 or using free text, as indicated in parenthesis. Additionally, the questionnaire collected demographic data and asked the participants if they had any further comments.

Q1: How did you perceive switching the HMDs? [disruptive - pleasantly]
Q2: What made switching disruptive or pleasantly? [free text]
Q3: How did you perceive the transitions? [disruptive - pleasantly]
Q4: Please sort the transitions you experienced by how you liked them. [Question was only included for group A. Each transition was listed, and users could assign positions 1 (liked most) to 5 (liked least) to each transition.]
Q5: Do you have further comments?

As Figure 6.7 shows, we varied with which one of our implemented transitions the participants from group A started to minimize the individual AR and VR nuggets' influence on the participant's perception and answers to the questionnaire. We kept the same order of transitions and only changed with which one the participants started. With five transitions in total and 10 participants in group A, each variation was experienced by two participants.

6.5.2 User Study Results

Table 6.5 summarizes the user study's results. We calculated pragmatic and hedonic quality and attractiveness based on the answers to the AttrakDiff questionnaires. To calculate the perceived presence, we used the results from the SPES. We conduct two-tailed Man Whitney U tests with a significance level of 0.95 to check for statistically significant differences between the groups and transitions. There were no statistically significant differences in the post-study questionnaire (Q1 and Q3). We grouped the results for further analysis, once by AR and VR experiences and once by individual transitions.

Grouped by AR and VR experiences, the test revealed no statistically significant differences between groups A and B from the AttrakDiff questionnaire. For the SPES, it revealed one

	Group A		Group B		Both groups		Answer Scale
	Ø	SD	Ø	SD	Ø	SD	
Q1 (perception switching)	4.80	1.33	4.90	1.04	4.85	1.19	1 - 7
Q3 (perception transitions)	5.50	0.81	4.90	1.22	5.20	1.08	1 - 7
pragmatic quality	3.58	0.14	3.29	0.24	3.44	0.18	1 - 7
hedonic quality	4.33	0.26	4.66	0.24	4.50	0.19	1 - 7
attractiveness	3.94	0.22	3.92	0.09	3.93	0.15	1 - 7
self-location presence	1.82	0.15	2.00	0.14	1.91	0.10	1 - 5
possible actions presence	2.52	0.46	2.36	0.15	2.44	0.26	1 - 5
overall presence	2.17	0.25	2.18	0.09	2.17	0.13	1 - 5

Table 6.5: Results from our user study.

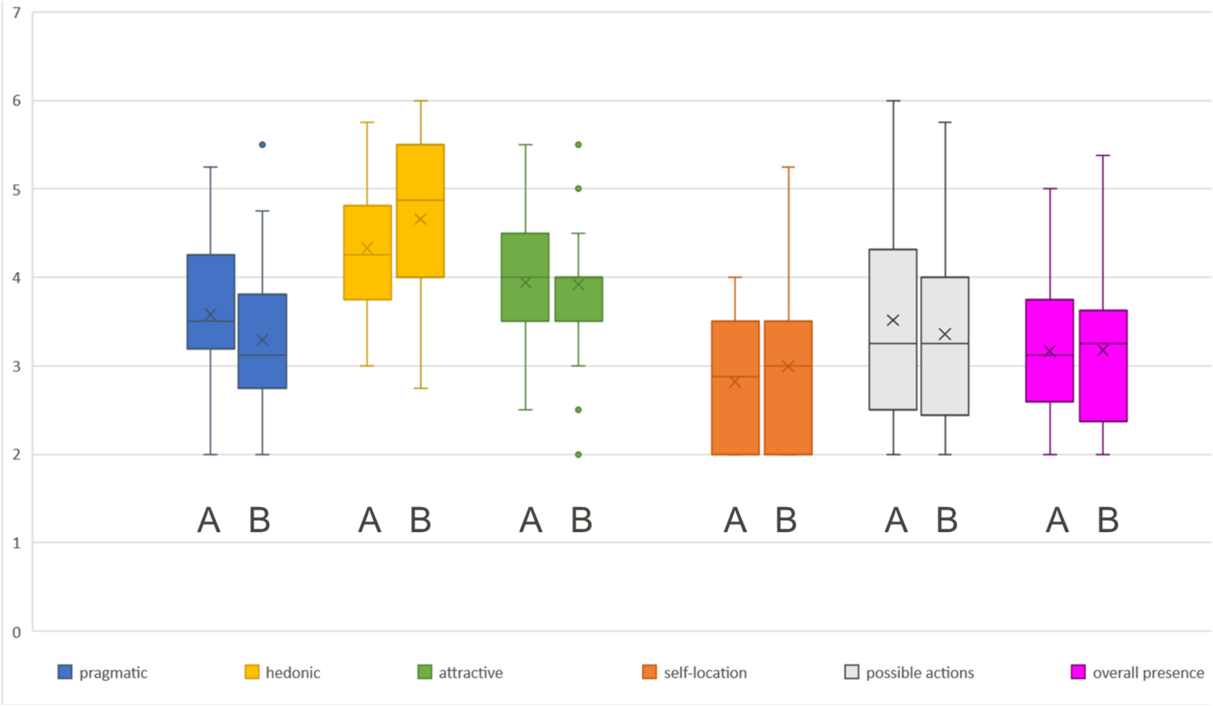


Figure 6.8: Results from the AttrakDiff and SPES.

significant difference regarding the self-location with p = 0.0085 for the third AR nugget. Here, group A rated their self-location presence lower than group B (group A Ø 1.60, SD 0.61, group B Ø 2.15, SD 0.87). For possible actions and overall presence, we found no statistically significant differences.

To group the answers by the five transitions, we averaged the answers from the participants from group B to use as compare values. Again, the Man Whitney U test revealed no statistically significant differences for the AttrakDiff nor possible actions and overall presence on the SPES. However, the test showed one statistically significant difference for self-location presence during the portal transition with p = 0.0340, group A Ø 1.73, SD 0.68, and group B Ø 2.00, SD 0.85). For the portal, two participants explained that they needed to figure out if and how they were supposed to interact with the portal (Q5).

Figure 6.8 visualizes the outcomes from the AttrakDiff and SPES. The results from the AttrakDiff show that the participants rated the transitions' pragmatic quality slightly less than a neutral value of 4 with Ø 3.44, SD 0.18. For the hedonic quality, a value slightly more than the neutral value is reached with Ø 4.50, SD 0.19. The transitions attractiveness is also close to neutral with Ø 3.93, SD 0.15. This is similar for both individual groups. The overall presence is calculated based on self-location and possible actions presence. It is slightly less than the neutral value of 3, with Ø 2.17, SD 0.13, and is similar for both groups.

In Q1, both groups rated switching HMDs rather pleasant than disruptive. Three participants stated that they found the HMD uncomfortable and thus appreciated being able to take it off (Q2). Five explain that they found it cumbersome to adjust the HMD's straps to their heads and that this step took some time (Q2). Two participants explicitly stated that they liked to use both HMDs and experience their differences and individual strengths (Q2). The outcomes from Q3 show that both groups found the transitions pleasant. While group A found them slightly more pleasant than group B, the Man Whitney U test showed no statistically significant difference.

When we asked the participants to rank the transitions (Q4), eight out of ten from group A placed the arrows transition first or second (liked most or second most). They explained that they found this transition easy to understand and instantly knew it was their sign to switch the HMD (Q5). In Q5, three participants explained that they would like a combined transition, e.g., one that combines the portal with the audiovisual information from the indicator and HHD on hand transition.

Overall, our results show that switching HMDs does not necessarily disrupt a combined AR and VR experience. This applied to both groups in our user study, regardless of whether we initiated a transition. Thus, one can draw from the individual strengths of AR and VR HMDs to create experiences mixed with AR and VR nuggets. Based on our results, we also argue that using transitions that provide users with clear instructions for switching HMDs is helpful. For example, a transition that combines audio, animation, and text could support clear instructions, as one of our participants pointed out.

6.6 AR Nugget Authoring Using Different Degrees of Immersion

This section evaluates our stand-alone AR nugget authoring tool ARNAUDDI that we introduced and implemented in Section 6.6. The user study targets to evaluate if authors without programming knowledge find working with AR nuggets supportive. Additionally, it aims to find out how these authors rate ARNAUDDI's pragmatic and hedonic qualities. For this, we conducted a user study that we describe in the next subsection. The following subsection analyzes the user study's results; the last subsection discusses these.

6.6.1 User Study

Our user study incorporates 48 unpaid volunteers between 20 and 34 years (Ø 24.20, SD 3.07 years). No participants are programming experts, and most of them are students in business administration and industrial economics. The participants have little or no experience with AR in general and their experience on a Likert scale from 1 (no experience) to 7 (experienced, using AR at least once a week) with Ø 1.71, SD 1.08). On the same scale, they rate their experience with creating AR applications Ø 1.54, SD 1.02.

Before users agreed to participate, we gave them a demo of ARNAUDDI so that they could decide if they wanted to test it. If they decided to participate, we provided the install files to install ARNAUDDI on a desktop computer and an AR device. Additionally, we provided a selection of image targets that the participants could use as real-world anchors and virtual 3D models. Furthermore, we provided a) instructions that described how to install ARNAUDDI on both devices, b) details about the user study's procedure, c) contact information, and d) a link to an online questionnaire. The instructions also gave an overview of AR nuggets from an author's point of view.

We asked the participants to work with ARNAUDDI independently over three weeks and to contact us with any questions. This task was not bound to any location or time constraints from our side. As also described in the instructions, we asked them to adapt at least one AR nugget per type, i.e., three or more AR nuggets in total. Finally, after three weeks or anytime after they worked with ARNAUDDI, the participants should anonymously answer an online questionnaire. They could also optionally upload their AR nuggets, images, screenshots, and a log file from ARNAUDDI. ARNAUDDI logged a) for much time the participants used it, b) which actions (selecting, replacing, moving, rotating, or scaling objects) they performed, and c) whether actions they performed the actions on the computer or AR device. The log file was saved in a text-based file format so that participants could read it to verify that they did not upload any personal data within the file.

The online questionnaire started with the short version of the AttrakDiff [HBK03] to measure pragmatic and hedonic qualities (10 items). Then, it asked the questions listed in the listing below. The listing indicates the questions' answer options and scales in parathesis behind the questions. The first seven questions refer to all types of AR nuggets. Next, it asked Q8 to Q17 individually for each AR nugget type. Finally, it asked participants if there were further use cases in their daily lives where they would like to use ARNAUDDI (Q18), how much experience with AR applications in general (Q19) and with creating AR applications (Q29) they have, their ages, and their genders.

Q_1: Which device did you prefer for the following actions? [computer or AR device]
  (a) moving labels
  (b) rotating labels
  (c) scaling labels
  (d) move 3D models
  (e) rotate 3D models
  (f) scale 3D models
  (g) look at the preview

Q_2: Switching my focus from computer to AR device was... [1 complicated - straightforward 7]

Q_3: Switching my focus from AR device to computer was... [1 complicated - straightforward 7]

Q_4: If I had only the computer but no AR device available, I would have needed ... time to create the AR nuggets [less 1 - 7 more]

Q_5: Which additional edit functionalities would you like on the computer? [free text]

Q_6: Which additional edit functionalities would you like on the AR device? [free text]

Q_7: Placing and moving objects (3D models, labels) in the 3D room was... [1 easy - complicated 7]

- (a) by pressing keys on the computer device
- (b) by clicking the buttons on the computer's UI
- (c) by tapping the buttons on the AR device's UI
- (d) by tapping and holding the object on the AR device

Q_8: How do you rate the workload for one *show & tell / quiz / semantic zoom* AR nugget? [1 high - low 7]

Q_9: How satisfied are you with your final *show & tell / quiz / semantic zoom* AR nuggets? [1 not satisfied - satisfied 7]

Q_{10}: How do you rate the workflow for *show & tell / quiz / semantic zoom* AR nuggets? [1 complicated - straightforward 7]

Q_{11}: What contributed to a complicated workflow? [free text]

Q_{12}: What contributed to a straightforward workflow? [free text]

Q_{13}: When you tried the *show & tell / quiz / semantic zoom* AR nugget with the default objects, did you get an impression of how your own AR nugget could look like? [1 No - Yes 7]

Q_{14}: Why were you able to get this impression? [free text]

Q_{15}: What did hinder you from getting this impression? [free text]

Q_{16}: How important was it to be able to try out the *show & tell / quiz / semantic zoom* AR nuggets directly on the AR device in the first step and at any time as it progressed? [1 not important - important 7]

Q_{17}: What did you especially like or dislike about the *show & tell / quiz / semantic zoom* AR nuggets? [free text]

6.6.2 User Study Analysis

To distinguish to which type of AR nugget Q8 - Q17 refer, we label them Q8.sat - Q17.sat when referring to the *show & tell* AR nugget type, Q8.q - Q17.q when referring to the *quiz* AR nugget type, and Q8.sz - Q17.sz when referring to the *semantic zoom* AR nugget type.

Figure 6.9 visualizes the outcomes from the AttrakDiff questionnaire. Its left side shows the mean scores for the word pairs, with 0 being the lowest, 3 a neutral, and 6 the highest rating. Except for the word pair "unimaginative - creative," which the participants rated with the most positive value, they rated all other word pairs with a value between 2 and 4. Aside from this word pair, all word pairs regarding the hedonic quality are rated with a value close to neutral but slightly negative. For pragmatic quality, the word pair "impractical - practical" is rated slightly positively, while the others are rated slightly negatively. The figure's right shows the AttrakDiff

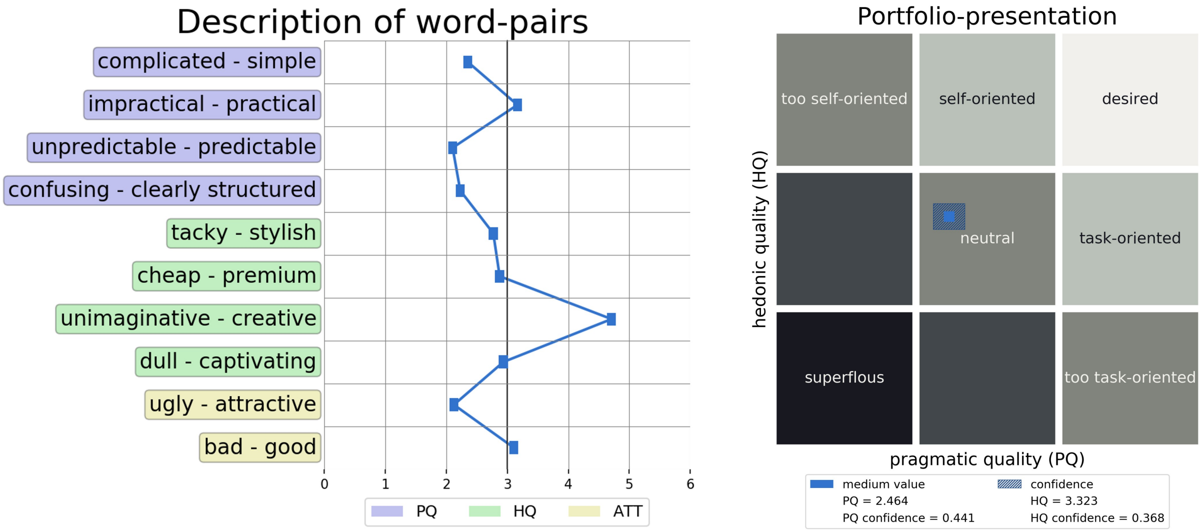


Figure 6.9: Outcomes from the AttrakDiff questionnaire [HBK03] Left: Description of word pairs. Right: Portfolio presentation. [Rau+22b].

portfolio presentation, which places ARNAUDDI's pragmatic and hedonic qualities in the neutral area with similar confidence intervals.

During adapting the AR nuggets, most participants (40 to 44) preferred to work with the desktop device over the AR device (Q1a - Q1f). However, for previewing the AR nuggets, 34 participants (70.83 %) preferred working with the AR device over the desktop device. Switching focus between the devices was perceived as straightforward with Ø 5.10, SD 2.11 to switch from desktop to AR device (Q2) and Ø 5.44, SD 1.74 to switch from AR to desktop device (Q3). Our participants believe that adapting the AR nuggets would have been neither slower nor faster if they had not used an AR device (Q4, Ø 4.13, SD 1.39).

Additional authoring functionalities that the participants would like to use on the desktop device (Q5) are a 3D navigation control, similar to established 3D viewing and editing tools, and usability improvements (e.g., connecting devices using a QR code instead of typing the server id manually, support of drag & drop, scaling using the mouse wheel, selection objects from a list instead of the 3D view). For the AR device (Q6), five users would like if the device supported touch input to edit the 3D models. The interactions that allow to placement and movement of virtual objects in 3D were rated mixed (Q7). Clicking buttons was perceived as the most complicated, both on the desktop device (Q7b, Ø 5.41, SD 1.64) and on the AR device (Q7c, Ø 5.85, SD 1.38). On the desktop device, participants found it easier to place and move virtual objects by pressing keys on the keyboard (Q7a, Ø 4.74, SD 1.76). The participants found the easiest interaction to place and move the virtual objects was to tap and hold the object on the AR device's screen (Q7d, Ø 4.50, SD 2.10). However, half of the participants (24 persons) did not use the AR device's touch functionality and are not included in the calculation for Q7d. All values for Q7 are larger than a neutral value of 4 on the 7-point scale. Thus, participants perceived all of these four options as rather complicated than easy.

For the part of the questionnaire we asked on a 7-point scale for each type of AR nugget individually, we conducted Friedmann tests with a significance level of 0.95 to find differences

Q	.sat Ø	.sat SD	.q Ø	.q SD	.sz Ø	.sz SD
Q8 (workload)	4.21	1.62	4.40	1.74	3.60	1.59
Q9 (satisfaction)	4.17	1.68	4.15	1.81	3.27	1.79
Q10 (workflow)	3.90	1.54	4.96	1.35	3.54	1.55
Q13 (first impression)	5.10	1.48	5.35	1.42	4.02	1.91
Q16 (try anytime)	4.90	1.93	4.21	1.88	4.83	1.77

Table 6.6: Outcomes from the questionnaire for questions that we asked for each type of AR nugget (*show & tell* (.sat), *quiz* (.q), *semantic zoom* (.sz)) and answered on a scale from 1 to 7.

between the AR nugget types. Table 6.6 shows the outcomes from this part of the questionnaire. The participants rate the workload (Q8) neutrally for the *show & tell* and *quiz* AR nuggets and as a little higher for the *semantic zoom* AR nugget. Their satisfaction with their own AR nuggets is also on a neutral level (Q9). Similar to Q9, the *show & tell* and *quiz* AR nuggets are rated with values close to each other, while the participants are a little less satisfied with their *semantic zoom* AR nuggets.

For the workflow (Q10), the Friedmann test found statistically significant differences between the three types of AR nuggets. The workflow for the *quiz* AR nugget was rated statistically significantly more positively than the one for the other two AR nuggets with a p-value of 0.0001. For the *show & tell* AR nuggets, 13 participants criticized the workflow when connecting labels with the 3D models (Q11.sat). For all types of AR nuggets, but at most for the *show& tell* AR nugget, several participants described unexpected behavior (bugs) from ARNAUDDI (Q11). In the case of the *quiz* AR nugget, three users stated that they had problems with the menu's arrangement as some menus occluded other menus elements (Q11.q). For all three AR nugget types, our participants found that the order of the menu contributed to a straightforward workflow (Q12) and explicitly stated that they found it helpful to start with a ready-to-use default AR nugget (Q12).

For the *show & tell* and *quiz* AR nuggets, participants explain that the default AR nuggets are good examples to understand the application (Q14.sat, Q14.q). Our participants found it easiest to get an impression of how their adapted AR nuggets could look like for the *show & tell* and *quiz* AR nuggets, but significant ($p = 0.0230$) more difficult for the *semantic zoom* AR nugget (Q13). They explain that the magnifying glass of the *semantic zoom* AR nugget was not correctly scaled to their AR device and was placed above the whole AR device's screen (Q15). Thus, they could not understand that the semantic zoom objects were only visible through the magnifying glass.

For all AR nuggets, our participants found it important to be able to test and experience the adapted AR nuggets at any time (Q16).

When the questionnaire asked the participants what they liked or disliked specifically about the *show & tell* AR nugget (Q17.sat), they stated to appreciate the default AR nugget's simplicity and that they could create a complex AR application based on it. They disliked how to connect labels to virtual objects, and five users stated that this did not work at all for them. Some participants wanted to add more than one 3D object to the *show & tell* AR nugget. For the *quiz* AR nugget, the default AR nugget's simplicity was similarly perceived positively (Q17.q). The participants also liked that the *quiz* AR nugget reacted to correct and wrong answers through blinking. Nine users described that correct answers were detected as wrong, which can happen if the 3D model is not configured with a fitting collider. On the *semantic zoom* AR nugget, our participants liked that they could include multiple semantic zoom objects (Q17.sz). They also liked the idea of the magnifying glass and pointed out that this could help to make complex 3D models more clear to users. However, the magnifying glass also caused usability issues, which the participants disliked. One participant suggested enhancing the *semantic zoom* AR nugget with labels.

Further use cases where our participants would like to use ARNAUDDI are tabletop and board games, education, navigation, guided tours, or care at home (Q18).

6.6.3 User Study Discussion

Our participants found the concept of AR nuggets and starting with a default AR nugget as an example application helpful to get started. Although the default AR nuggets only include basic geometric objects as placeholders, these are sufficient to give authors an idea of the AR application and to get started (Q13). Despite this simplicity, the participants rated ARNAUDDI as creative in the AttrakDiff's word pairs.

However, some users stated that ARNAUDDI had unexpected behaviors or that interactions were unclear. For example, ARNAUDDI supported touch input in the AR device, but some authors listed this as a feature they would like implemented in the future (Q6). This indicates that they were either unable to use touch input or did not realize that ARNAUDDI supported it. Furthermore, this could contribute to participants preferring the desktop device over the AR device for adapting the AR nuggets. The usability issues also have a negative impact on ARNAUDDI's pragmatic qualities, as reflected by the AttrakDiff's outcomes. Besides fixing unexpected behaviors or bugs, one could provide a manual or tutorial to ensure that authors can be aware of all authoring functionalities, including touch input on the AR device. However, AR nuggets or ARNAUDDI could also implement smart assistant functionalities that can support authors pro-actively, similar to our virtual assistant from Section 3.4, Section 4.3, and Section 6.3.

Further examples related to usability are scaling issues due to different screen sizes (Q11). For example, some participants stated that the magnifying glas in the *semantic zoom* AR nugget was taking up the whole screen size so that they could hardly see virtual objects without looking through the magnifying glas. This could have contributed to lower scores for the *semantic zoom* AR nugget compared to the other two types of AR nuggets. The scaling issues can have a negative impact on hedonic qualities and, if users cannot see or click occluded menu elements, also on pragmatic qualities. ARNAUDDI and its AR nuggets should accommodate different hardware and screen sizes to improve this.

For some 3D models used in the *quiz* AR nugget, participants reported that the AR nugget did not provide the correct feedback for an answer. If the 3D models' colliders do not fit the 3D models, clicking or tapping on the 3D models cannot be recognized correctly, which leads to unexpected behavior. The outcomes from the AttrakDiff and the word pair "unpredictable -

predictable" also reflect this. While some participants realized and disliked this for the *quiz* AR nugget, this could also have negatively impacted the process of placing labels for a *show & tell* AR nugget. Our authoring process for adapting AR nuggets does not include creating and preparing 3D models; we see this as another challenging task. However, ARNAUDDI could be extended with further smart functionalities and check the 3D models' colliders upon importing them. With this, ARNAUDDI might become more predictable and improve its pragmatic qualities.

ARNAUDDI sets limits regarding the real-world anchors for the AR nuggets as it only supports Vuforia image targets. It could be helpful to extend it to also support augmenting physical 3D objects, surfaces, or any PoI in a room.

ARNAUDDI implemented three exemplary types of AR nuggets, which also limited our participants in options for how their own AR application could look like. Thus, some participants wanted to combine AR nuggets or had other ideas that ARNAUDDI did not support. Here, a more comprehensive selection of implemented AR nuggets can increase the available options for the authors. However, if the participants want to adapt AR nuggets for more complex scenarios, the complexity of the whole authoring tool might need to increase to provide more functionalities. This can make adapting AR nuggets more complicated for first-time authors. Additionally, extending the ARNAUDDI's functionalities and adding more default AR nuggets requires programming knowledge. It typically involves interdisciplinary communication and cooperation from developers (programming experts who implement the functions and AR nuggets) and AR nugget authors (who identify the patterns). Some authoring functionalities can be perceived as supportive or unwanted restrictions and limitations. For example, we implemented the *show & tell* AR nugget's labels to adjust the text size to fit the labels' size to support authors. However, one participant stated that he would have liked to be able to adjust the text size on his own. To accomodate users' different levels of experience, it could be an option to allow more experienced authors to enable extended authoring functionalities. Thus, the following section evaluates our alternative approach to a stand-alone authoring tool which includes more options for authors and targets to support creating more complex AR applications.

6.7 Integrated AR Nugget Authoring Tools

The museum application described in Section 6.4 was authored with the authoring tools introduced in Section 5.2. To further evaluate our authoring tools from Section 5.2, we conducted an authoring workshop targeted to the context of a museum. The following subsection describes how we realized the authoring workshop and analyzes and discusses its outcomes.

6.7.1 Authoring Workshop

We invited 14 voluntary, unpaid participants (5 female, 7 male, 2 genders not specified) to our authoring workshop. They were between 22 and 49 years old (Ø 29.75, SD 8.32). Because our authoring tools target persons who want to make further adaptions to the default AR nuggets than a stand-alone authoring tool supports, we invited persons with a background in design, media didactics, or other areas that involve technical implementations. On a scale from 1 (no experience) to 7 (usage almost every day), they rated their experience with AR Ø 5.07, SD 1.16, with Unity or other game engines Ø 4.21, SD 2.04, and with authoring tools in general Ø 2.93, SD 2.12. We provided each participant with a laptop with our tools installed and a HoloLens 2. Because of limited laptops and HoloLenses, we divided the authoring workshop into two groups that participated on different dates. One group, in the following called group A, participated in an office environment, where we placed miniature skeletons to represent skeletons from a museum. The others, group B, visited a museum for natural history, where the authoring workshop took place. All volunteers were free to choose their preferred date and location to participate. Eight persons decided to participate in the office environment (group A) and six participants in the museum (group B).

Before the participants arrived, we prepared a workplace for each, where we placed a HoloLens 2 and a laptop. We also started the laptops and opened Unity. Additionally, we prepared and placed a manual that, in bullet point form and with screenshots, explained how to use each tool. The manual also included a task list. First, the task list asked participants to adapt a *superimposition with interactive transparency control* AR nugget to show how the skin of a whale or dinosaur lies over the animal's bones. Next, it asked them to use a *show & tell* AR nugget to explain the skeleton of an animal. Finally, it invited participants to realize their own ideas by adapting further AR nuggets and using the default AR nuggets as inspiration. Based on feedback from group A, we realized that our instructions might not be detailed enough for persons who have never worked with Unity before. Thus, we added some more detailed instructions for the adaption process. The manual for both groups are available in Appendix D.

When the participants arrived, we welcomed them, and they sat down in one of the prepared workplaces. We introduced them to the topic with a presentation where we explained AR nuggets, interactions with the HoloLens 2, challenges of placing virtual objects in a physical room, and gave an overview of interactions within Unity. Then, we gave them a brief introduction to our tools by showing them where to find and how to use them in the Unity project. This took about 45 minutes. Next, we asked the participants to work with the authoring tools independently. While they did so, we answered individually arising questions. The participants individually switched between adapting AR nuggets and testing them directly in Unity on the laptops. When they wanted to test their adapted AR nuggets on the HoloLens 2, they used the mode switcher tool to build the application. This took roughly two hours, during which the participants took breaks as they needed individually. After adapting and testing, we asked the participants to complete an anonymous online questionnaire. The questionnaire included the following questions, where it repeated Q7 - Q15 for each type of AR nugget the participants worked with.

Q_1: How satisfied are you with your own AR application? [1 not satisfied - satisfied 7]

Q_2: How would you rate the difficulty of... [1 very simple - very difficult 7]

  (a) replacing a standard object (e.g., a cube) with your own object?
  (b) switching between visitor and author modes using the Mode Switcher tool?
  (c) blocking certain objects for visitor or author modes (Select Parent Mode Tool in combination with Mode Switcher Tool)?

(d) setting start and end conditions for AR Nuggets?
(e) using the Rotate Parent To Gameobject tool?
(f) adding new labels?
(g) using the Grabbable Parent Tool?
(h) using the Anchor My Parent tool?

Q_3: What did you find particularly difficult when using the tools? [free text]

Q_4: What did you find particularly easy when using the tools? [free text]

Q_5: What would you like to see improved in the tools? [free text]

Q_6: What tools or authoring functionality were you missing? [free text]

Q_7: How much work do you think is required to adapt this AR Nugget type? [1 very little - very much 7]

Q_8: How important was it to be able to try out this type of AR Nugget in the first step and then, as the authoring process progressed, again and again? [1 not important at all - very important 7]

Q_9: What would you like to have improved about this type of AR Nugget? [free text]

Q_{10}: The standard AR nugget of this type inspired me to develop my own ideas for AR applications. [1 disagree - agree 7]

Q_{11}: By experiencing a standard AR nugget of this type, I could imagine what my AR application might look like. [1 disagree - agree 7]

Q_{12}: The AR nuggets of this type that I adapted look like I imagined they would. [1 disagree - agree 7]

Q_{13}: If the AR Nugget doesn't look like you imagined, what looks different? [free text]

Q_{14}: Did you have ideas you wanted to implement but couldn't with the given authoring tools? [free text]

Q_{15}: If you answered yes in the previous question, please describe the ideas you wanted to implement but could not with the given authoring tools. [free text]

Additionally, the questionnaire collected the participants' experience level, age, gender, profession, and further comments. Finally, we held a 25-minute open discussion round.

6.7.2 User Study Analysis

First, we checked for statistically significant differences between the two groups and the five types of AR nuggets. With a Man Whitney U test, we found only one statistically significant difference between the two groups for Q12. A Kruskal-Wallis test revealed no statistically significant differences between the AR nugget types.

For Q1 and Q2, we summarize the results in Table 6.7. Our participants rate their satisfaction with their own AR application (Q1) in a neutral area. They perceived replacing the AR nuggets' placeholder objects as very simple (Q2a). The participants also found using the mode switcher tool (Q2b) was simple. Except for Q2f, the participants rated the then following tasks Q2c - Q2h between simple and neutral with values between 3 and 4, where 4 was a neutral value. On Q2f,

		Q1	Q2a	Q2b	Q2c	Q2d	Q2e	Q2f	Q2g	Q2h
Group A	Ø	4.38	1.75	1.86	3.57	3.13	3.86	2.00	3.29	3.71
	SD	1.58	1.30	1.46	1.68	1.62	1.46	1.58	2.05	1.58
Group B	Ø	3.50	2.00	2.80	4.00	3.40	2.80	1.60	2.60	3.20
	SD	1.61	1.00	2.40	1.67	1.36	1.47	0.80	1.02	1.83
Overall	Ø	4.00	1.86	2.25	3.75	3.23	3.42	1.85	3.00	3.50
	SD	1.65	1.19	1.96	1.69	1.53	1.55	1.35	1.73	1.71

Table 6.7: Outcomes for Q1 (answered on scale 1 satisfied - not satisfied 7) and Q2 (answered on scale 1 very simple - very difficult 7)

they rated adding new labels as very simple. One participant notified us that she did not use the tools from Q2b, c, e, g, and h.

In the free text field, four participants explained that they found it difficult to figure out what to do if a tool did not work as expected, either due to a bug or wrong usage (Q3). Two persons stated that they found it challenging to find the correct settings in Unity's inspector window (Q3). Also regarding Unity, two persons described that they only sometimes knew where exactly to drag & drop the tools (Q3). Two participants mentioned usability-related challenges on the HoloLens 2 or in Unity's play mode, e.g., selecting a virtual object when there are multiple overlapping objects (Q3). One participant pointed out that the main challenge is not to use the tools but to apply one's own ideas to the tool (Q3). Seven participants stated that they found it particularly easy to adapt the AR nuggets, and one found it easy to edit the labels' texts (Q4). Two described that placing AR nuggets or labels in the room was easy (Q4). One participant pointed out that the drag & drop technique was intuitive and easy to use (Q4). As an improvement (Q5), one participant suggested more detailed documentation that provides a clear order of authoring tasks. One participant pointed out that it needed to be clarified whether a tool was part of Unity or provided by us (Q5). Other suggestions were adding a tool that supports animating 3D models, automatically attaching *grabbable* tools to all virtual objects, and allowing to combine AR nuggets with each other to create new AR nuggets (Q5). Two participants missed authoring tools to work with audio, e.g., to create a "spatial sound zone" that plays a sound when the user enters the zone (Q6). Two participants missed an undo function and one would have liked more interactivity within the AR nuggets (Q6).

The participants found that only a little work is required to adapt the AR nuggets (Q7). From all AR nuggets, they rated the *semantic zoom* AR nugget as the one that requires the most work, but this value is with Ø 3.17, SD 1.46 still between "little work" and neutral. The participants rate the importance of being able to try their AR nuggets at any time mainly in a neutral area (Q8). There are slight differences between the types of AR nuggets; the importance was rated lowest for the *compare* AR nugget, close to neutral for *show & tell* and *quiz*, and higher for the *semantic zoom* and *superimposition with interactive transparency control* AR nuggets.

Things the participants would like to have improved about *show & tell* AR nuggets are (Q9): a) labels automatically rotating in relation to the virtual objects, b) move the start point of the labels' lines a bit away from the labels' text, c) adjustable label text size, and d) a button to automatically create multiple labels. For the *compare* AR nugget, only one participant answered Q9 and stated

Nugget		Q7		Q8		Q10		Q11		Q12	
Type	Group	Ø	SD	Ø	SD	Ø	SD	Ø	SD	Ø	SD
Show & Tell	A	1.71	1.03	4.00	1.77	3.00	1.60	5.71	1.28	5.86	2.10
	B	3.00	0.89	4.20	0.40	4.40	1.96	5.60	1.50	4.60	2.06
	Both	2.25	1.16	4.08	1.38	3.58	1.89	5.67	1.37	5.33	2.17
Compare	A	1.50	0.50	3.25	1.79	3.25	1.79	2.75	1.92	5.25	1.79
	B	-	-	-	-	-	-	-	-	-	-
	Both	-	-	-	-	-	-	-	-	-	-
Quiz	A	2.00	1.10	4.40	2.15	5.00	2.10	6.60	0.80	6.00	1.55
	B	-	-	-	-	-	-	-	-	-	-
	Both	-	-	-	-	-	-	-	-	-	-
Semantic Zoom	A	3.17	1.46	5.33	1.80	5.83	1.77	6.33	1.11	5.50	1.38
	B	-	-	-	-	-	-	-	-	-	-
	Both	-	-	-	-	-	-	-	-	-	-
Superimposition with Interactive Transparency Control	A	2.63	1.11	5.00	1.73	4.38	1.73	6.00	1.00	5.50	1.87
	B	3.60	1.50	3.80	0.98	3.80	2.04	5.20	1.60	4.20	0.40
	Both	3.00	1.36	4.54	1.60	4.15	1.87	5.69	1.32	5.00	1.62

Table 6.8: Outcomes for questions asked for each type of AR nugget. If no participants from a group adapted a type of AR nugget, no data is available.

that she needed help understanding the added value of the *compare* AR nugget as the *show & tell* AR nugget was quite similar. She would have liked a more understandable example. Suggestions to improve the *quiz* AR nugget (Q9) are adding multiple choice and Likert scale questions. For the *semantic zoom* AR nugget, a participant suggested automatically placing the two virtual 3D objects in correct relation to each other when replacing the placeholder objects (Q9). Another participant wanted multiple magnifying glasses in one *semantic zoom* AR nugget (Q9). For the *superimposition with interactive transparency control* AR nugget, participants suggested improving the slider's handling (Q9).

Our participants' agreement to the statement that the default AR nugget inspires to come up with own ideas for AR applications (Q10) is on a neutral or slightly negative level for the AR nugget types *show & tell*, *compare*, and *superimposition with interactive transparency control* (Ø 3.00 - 4.38, SD 1.60 - 2.04). For the types *quiz* and *semantic zoom*, they agree more with this statement (Ø 5.00 and 5.83, SD 2.10 and 1.77). Yet, in Q11, the participants stated that experiencing the default AR nugget helped them to imagine what their own AR applications could look like for all AR nugget types (Ø 5.20 - 6.60, SD 0.80 - 1.60), except for *compare* (Ø 2.75, SD 1.92).

The participants found that their adapted AR nuggets mostly looked like they imagined (Q12). Here, the Man Whitney U test found a statistically significant difference between both groups for the *superimposition with interactive transparency control* AR nugget with $p = 0.0477$. Group A agreed more with the statement that their adapted AR nuggets looked like they imagined it, while participants from group B positioned themselves neutrally to the statement. In Q13, three participants from group B described that parts of the objects did not become as transparent as expected or not transparent at all (Q13). For the *show & tell* AR nugget, two participants elaborated that the labels' position and rotation were not as expected, and one participant described that one out of her three *show & tell* AR nugget did not show up on the HoloLens 2 (Q13). No participant answered Q13 for the *compare* AR nugget. For the *quiz* AR nugget, one participant explained that more time would be needed to find questions that make use of the 3D room (Q13). For the *semantic zoom* AR nugget, one participant explained that there was a disturbing zoom factor on the magnifying glass (Q13).

Four participants had ideas for the *show & tell* AR nugget they wanted to implement but could not because the tools limited them (Q14). They would have liked to a) place the text at different angles and distances to the object, b) play audio when moving closer to an object, and c) show (parts of) a virtual object transparent (Q15). No participants had or described ideas that they could not implement for the *compare* or *semantic zoom* AR nuggets (Q14, Q15). For the *quiz* AR nugget, one participant would like to be able to define single parts of virtual objects instead of whole objects as right or wrong answers (Q14, Q15). In Q14 and Q15, for the *superimposition with interactive transparency control* AR nugget, a participant would like to define elements that do not become transparent when the slider is used. Another participant would like to add animations and sound.

In the discussion round, participants described that having a ready-to-use application and replacing virtual objects was a good start for creating an AR application. Three participants who had experience with other AR authoring tools found using the AR nuggets as building blocks saved effort compared to those other authoring tools. One participant described that she would like to define types of surfaces, e.g., a chair or couch, where virtual objects are automatically placed. This function was not included in the authoring tools that she tested, but in our authoring approach from Section 5.3.

Most challenges the participants described referred to usability issues with the HoloLens 2 (e.g., grabbing virtual objects) or the provided 3D models. For example, the 3D models had quite different scales and initially did not fit well when loaded. The participants first had to scale the 3D models. Here, one participant suggested having a reference in Unity to better reckon the correct scale, e.g., a room scan where authors can place the virtual objects. Additionally, the colliders for the 3D models were not always correct. One participant suggested that our tools could make colliders visible so authors can check and correct them.

Another challenge that the participants described in the discussion round was that they did not know which tool to use for which purpose because they were not familiar with the tools' names or the names were not clear to them. Our participants explained that they deactivated the AR nugget manager initially because they wanted to first test the AR nuggets without regard to when they started or ended. Later, they forgot about the AR nugget manager and did not use it. Similarly, they did not use the condition manager tool and only worked in the authoring mode. They also mainly focussed on the first page of the provided manual, which described the authoring task, and did not see the further pages, which explained the tools in more detail.

6.7.3 User Study Discussion

Our participants rated their satisfaction with their adapted AR nuggets neutrally (Q1) but stated that they looked like they imagined (Q12). The participants' satisfaction was lower for the participants from group B, who participated in a museum. Although our participants had some experience with AR or authoring, they found it supportive and effort-saving to start with a ready-to-use application with placeholder objects. This might be more important if authors have no experience with AR as default AR nuggets could help to develop ideas of what is possible with AR. Integrating the tools in the Game Engine Unity seemed suitable as our participants found the tools generally easy to use (Q2, discussion round). However, our authoring tools could further support authors by guiding them to the settings where they can edit values, e.g., changing a label text in the shoe & tell AR nugget. This could also help if authors less experienced authors use the authoring tools. Our authoring workshop also revealed room for improvement regarding the usability of the HoloLens 2. For example, one participant suggested making colliders on the HoloLens 2 visible. The most current version of MRTK implements this, but this feature was not available when we started to develop our authoring tools.

As one participant pointed out, one challenge besides using the authoring tool is conceptualizing one's idea and identifying how to apply the tools to implement one's idea (Q3). It might be more challenging to apply the tools if an author already has a complex idea for an own AR application than using the tools to develop an idea. While our tools do not directly target this challenge, they can support authors in developing their own ideas by providing example applications. As another participant pointed out, the default AR nuggets provide a starting point to develop own creative ideas (discussion round). However, as results from Q10 indicate, more than the default AR nuggets alone might be required to inspire authors to own ideas for AR applications. Authors might additionally need a topic or use case (e.g., a whale exhibition in a museum), 3D models, or more examples.

Nearly all ideas the participants wanted to implement but felt limited to by the authoring tools (Q6, Q15) could actually be implemented with the provided authoring tools. However, authors might need more time to familiarize themselves with all tools before realizing so. For example, adding a sound when a user moves closer to an object could be implemented with our condition manager. This illustrates that it can be challenging to apply the authoring tools to one's own ideas. Still, all participants used the available time to work with the AR nuggets and other tools, so they did not have enough time to use the condition manager. From the limitations that the participants described, only creating 3D animations is not implemented by our authoring tools. We believe creating 3D animations is, similar to creating 3D models, an authoring challenge on its own.

For participants, it was more important to be able to test the *semantic zoom* and *superimposition with interactive transparency control* AR nuggets at any time than it was for other AR nugget types (Q8). One reason could be that labeling objects as in the *show & tell* and *compare* AR nuggets or the concept of a quiz is familiar to the authors. In contrast, using a *semantic zoom* or controlling an object's transparency is not common in other technologies or everyday lives.

6.8 Constraint-based Authoring with AR Nuggets

We evaluated selected functions of constraint-based authoring that can be applied to AR Nuggets. For this, we conducted a user study together with computer science students. The study evaluates our concepts for constraints and if the calculation of constraints works as users expect. Seven unpaid volunteers (3 female, 4 male) aged between 21 and 33 years (Ø 25.57, SD 3.70) participated. They classified themselves primarily as novice AR users on a scale from 1 (no experience) to 5 (daily usage) with Ø 1.71, SD 0.88.

6.8.1 User Study

With each participant, our user study took place in two different rooms. For example, room A could be an office environment, and room B could be a living room. In room A, we welcomed our participants and explained the topic of the study. Then, the participants used an introductory application on the HoloLens 2. This aimed to help them learn how to interact with the virtual content and use the tap2place gesture. When users felt comfortable, we started the prototype and advised the participants to walk around the room with the HoloLens 2 so that the HoloLens 2 could capture the whole room. When the participants had walked around the room, we asked them to place six objects in the room on any surfaces they liked. They could use the tap2gestures from the introductory application to do so. The provided virtual objects were: a Christmas tree, a blue and a yellow present box, a garland, a wrapped candy, and a lantern. Once participants expressed that they had everything placed as they liked, we advised them to use the hand menu to start the process of automatically calculating constraints. Then, we switched to room B, where we again asked the participants to move around to have the HoloLens 2 capture the room. Next, we asked them to use the hand menu to automatically place the virtual objects in the room based on their calculated constraints from room A. After the automatic placement, the participants walked around to view if and where the application placed the virtual object. Finally, they took the HoloLens 2 off and answered a questionnaire.

The questionnaire consisted of one part for the template scene creation in room A and a second part for the automatic placement in room B. It asked the following questions and gave the following answer options where it asked Q1.1, Q1.2, Q2.3, and Q2.4 for each of the six virtual objects. Finally, the questionnaire collected demographic data and the participants' experience level regarding using AR.

$Q_{1.1}$: Which surface did you place the [virtual object] on? [Answer options: wall, ceiling, floor, platform, background]

$Q_{1.2}$: Where did you place the [virtual object]? [Answer on a scale: 1 smallest distance - largest distance 5]

$Q_{1.3}$: Where did you place the [virtual object]? [Answer on a scale: 1 smallest surface area - largest surface area 5]

$Q_{1.4}$: Were there any surfaces that were not detected and highlighted? [Answer options: yes or no]

$Q_{1.5}$: If there were surfaces that were not detected and highlighted, please describe them. [Answer as free text]

$Q_{2.1}$: Were there any surfaces that were not detected and highlighted? [answer options: yes or no]

$Q_{2.2}$: If there were surfaces that were not detected and highlighted, please describe them. [Answer as free text]

$Q_{2.3}$: Was the [virtual object] placed as you expected? [Answer on a scale: 1 placed as expected - placement not understandable 5]

$Q_{2.4}$: How did you expect the placement of the [virtual object]? [Answer as free text]

$Q_{2.5}$: What further options to design a scene would you expect or can you imagine? [Answer as free text]

$Q_{2.6}$: Were there any problems using the prototype? [Answer options: yes or no]

$Q_{2.7}$: If there were problems regarding using the prototype, please describe them. [Answer as free text]

6.8.2 User Study Analysis

Five participants stated that all surfaces were successfully detected (Q1.4). One of the other two participants listed a chair's sitting area as a surface that was not detected (Q1.5), and the other one explained that the detection improved over time. In the second room, four participants listed surfaces that were not detected (Q2.1): parts of a wardrobe, the window sill, some windows and glass surfaces, and a coffee table (Q2.2).

Virtual objects were mainly automatically placed as expected (Q2.3) when the participants put them on the floor (Ø 1.60, SD 0.80) or the ceiling (Ø 1.0, SD 0.0). Also placed as expected (Q2.3) were objects that were placed on a platform (Ø 2.00, SD 1.41) or background surfaces (Ø 2.33, SD 1.33). In contrast, the application did not place virtual objects that the user intended to place on a wall as expected (Q2.3, Ø 3.50, SD 1.43). However, the participants did not find the placement completely incomprehensive. It calculated a distance constraint for most virtual objects placed on a wall, but the participant expected a surface constraint. For example, for the garland object, six out of seven participants intended to place it on a large or very large wall. (Q1.1 - Q1.3)

When the participants moved through the room, the virtual objects changed their position to keep their distance relative to the user, e.g., to stay at the surface furthest away from the user. Two participants describe this change as unexpected in Q2.4.

When asked for further ideas for automatic placement (Q2.5), one participant suggested including the user's orientation. For example, if the users place a virtual object behind themselves, the virtual object should automatically be placed behind the user again. Additionally, this participant and another one suggested using the user's eye level as a reference for the virtual objects' height. One user suggests that the distance of the virtual objects to each other should stay constant. Two users state that virtual objects manually placed in a corner of the rooms should automatically be placed in a corner again. One user expressed that she expected objects to be mirror-inverted when she turned around.

In Q2.6, two participants reported that it took a while to complete the automatic placement. One participant explained that the tap2place gesture was not working fluently. Another participant suggests starting the manual placement in a smaller room than the room where the automatic placement takes place because there are more options for placements in the larger room. Two participants had no difficulties.

In total, 14 virtual objects were placed precisely as expected, six nearly as expected, four were rated with the middle value, four times the placement was not understandable, five times not understandable at all, and nine times the participants could not find the virtual object in the second room.

6.8.3 User Study Discussion

Virtual objects placed on the ceiling or floor were most likely to be automatically placed as expected by the participants, while objects placed on the wall were least likely to be so (Q2.3). Notably, rooms typically have a single ceiling surface and a single floor surface but multiple walls or other surface types. Our prototype placed virtual objects automatically on the same surface type as the user placed them. If only one or a few surfaces of this type are available, the application can match the user's expectations more easily than when many options are available. For example, if there is only one ceiling in each room, and the user manually places a virtual object on the ceiling, the virtual object will also be placed on the ceiling in the second room, as a user typically expects. With our additional two constraints of surface area size and distance, the system did not always choose the constraint the user expected.

The constraints were calculated based on the surfaces' ranks, where a rank greater than 0.5 resulted in a minimum constraint (at least the given surface's distance or surface area) and smaller ranks in a maximum constraint (maximally the given surface's distance or surface area). Because users have no list with all distances and surface areas, they might think that a surface is ranked higher than 0.5 on the normalized scale, although it is not. Then, the application would place the virtual object on a surface with a surface area or distance less than the surface where the user placed it, although the user expects it to be placed on a similar-sized or larger surface. Furthermore, it is difficult to decide between different constraints. Here, it could be helpful to use a surface area and distance range instead of the ranks, e.g., the surface area or distance should be a maximum of 20% more or less of the original area or distance.

The users' distance to the virtual objects varies as they move around in the room. Therefore, the constraints depend on the users' position at the point of time when the constraints are calculated. Some users did not expect this. One option could be informing users before the constraints are applied so they can go to a position of their choice. Then, in the automatic placement process, the AR nugget could place the virtual objects once with the matching distance constraint and then keep their position to avoid changes the user does not expect. However, this would make it more important that the AR nugget has fully scanned the room before the automatic placement process starts. Otherwise, surfaces that were unknown to the AR nugget it placed the virtual objects would be disregarded. Another option could be to only use constraints that are independent from the user's position. Also, it could be an option to allow users to set weights for distance and surface constraints, i.e., the users could decide which constraints have more weight.

The additional constraints that our users suggested (Q2.5) can help to describe a scene in more detail. However, then the calculation of the constraints would become more complex. Further tests are required to evaluate if a complex calculation of constraints can match the majority of users or if users expect other constraints than the system would calculate.

Chapter 7

Discussion

This chapter discusses where this thesis contributed to answering the research questions we proposed in Section 1.2.

RQ1: How can a nugget concept be applied to AR?

1. How can AR nuggets be defined?
2. What universal patterns for developing an AR application can be identified (and applied)?

We applied the concept of nuggets, based on microlearning and VR nuggets, to AR and described our concept in Section 3.1. We defined AR nuggets as ready-to-use, stand-alone, and self-contained AR applications. Each AR nugget is based on one pattern and implements all interactions that the pattern requires. Moreover, AR nuggets include placeholder objects and default parameters. Because AR nuggets include all functions and objects required to execute them, authors can experience an AR nugget before making adaptions. No third application is necessary to experience an AR nugget as each AR nugget is a stand-alone application and self-contained.

We identified eight patterns with different variations and implemented them in AR nuggets. Section 3.2 lists these and shows example scenarios from cultural heritage and educational settings where our patterns and AR nuggets were applied.

Overall, we showed that a nugget concept can be applied to AR and that suitable patterns for AR nuggets exist. With this, we contribute to filling two research gaps: First, we contribute to findings on how AR can be applied to microlearning. Moreover, we contribute to exploring how patterns can support authors and users of AR applications.

RQ2: What features are included in AR nuggets?

1. How can AR nuggets support users in their AR experience?
2. How can AR nuggets be combined with VR nuggets?

3. How can multiple AR nuggets comprise one larger AR experience?

We developed multiple features that target to support complex AR scenarios. First, we introduced (Section 3.4) and implemented (Section 4.3) a virtual assistant targeting to support AR users. We identified that users, especially novice users, are in need of support or advice to ensure sufficient lighting conditions, hold the AR device stable and at an appropriate distance, use gestures, touch input, or other interactions. Yet, there can be more situations where users need support to use AR nuggets or AR applications in general. Our virtual assistant integrates support to help users overcome our identified challenges into AR nuggets. In contrast to traditional tutorials, our virtual assistant provides support and hints automatically in case of need. This has the advantage that users receive the support at the point of time when they need it and do not have to remember several instructions from a tutorial or manual. However, preparing for different scenarios becomes more challenging, and some users might prefer traditional tutorials or manuals. One option can be to provide a manual along with the virtual assistant. The virtual assistant monitors data from the AR device's sensors to detect if users face challenges. For example, the virtual assistant monitors the AR device's acceleration and camera sensors to calculate its acceleration and surrounding lighting conditions. Based on these, it can detect if a user is holding the device too shaky or too dark or bright. Then, it can inform the user to hold the AR device more steadily or to change the lighting conditions. The detection of challenges during runtime allows the virtual assistant to provide individual support and to adapt to the different levels of experience or required assistance from users. In our user study, the participants found the virtual assistant overall helpful. However, the hints' timing and duration can be improved.

Next, we described how we can combine AR nuggets with VR nuggets using transitions in Section 3.6. We target to use AR where it can add value and use other technologies where others can add more value. Thus, we researched how AR and VR nuggets can connect to one mixed AR/VR experience. Section 4.5 implemented a prototype that employed AR nuggets to be experienced on AR HMDs and VR nuggets on VR HMDs. In a user study (Section 6.5), we showed that users did not feel disturbed by switching HMDs between AR and VR nuggets. While this shows that it can be suitable to draw from the individual strengths of AR and VR nuggets and devices, our user study did not find a statistically significant impact from the transitions to the users' perception of presence. Moreover, the transitions should provide clear instructions, e.g., combined with audio, animation, and text, to the users.

Finally, Section 3.5 and Section 4.4 introduced and implemented an AR nugget manager that allows combining multiple AR nuggets into a single, more complex AR application. This way, users do not need to start another AR application at each PoI where they want to experience an AR application. Instead, they only start one overall AR application, and the AR nugget manager automatically starts the individual AR nuggets based on predefined pre- and postconditions. Section 6.4 showed that two media designers could create a complex AR application using AR nuggets and the AR nugget manager.

Additionally, the Section 3.5 and Section 4.4 described how AR nuggets can guide users from an AR nugget at one location to one at another location. This is especially relevant for location-specific AR nuggets that can be experienced in a large or complex location, environment, or building where users cannot see all AR nuggets. We realized this by introducing three types of navigation AR nuggets: a) node-based, b) based on pre-processed scan data, and c) based on spatial mapping information. Using only spatial mapping information requires the least preliminary work; authors only need to define the pathway's goal. Moreover, it can react to dynamically changing environments, e.g., if persons or physical objects get in the way. However, the limited knowledge of spatial mapping information might be insufficient so that users get stuck in a corridor's dead end or take a route that is not the shortest one. Authors can add nodes to a pathway to avoid this. Then, the AR nugget calculates the pathway based on the nodes. However, it can be cumbersome if multiple pathways have lots of nodes to add. In such a case, using a pre-processed scan as a basis to calculate the pathway can be suitable. Then, authors need to create and pre-process the scan instead of placing nodes. Whether it is preferable to use option a) and to add nodes or to use option b) and to create and pre-process a scan depends on the number of pathways and nodes. Both, options a) and b) can also be combined with spatial mapping information (option c).

Overall, our answers to this RQ contribute to the development of sophisticated AR applications by filling the research gaps pointed out in Section 2.4 and Section 2.5. Additionally, it contributes to filling the research gap about how small applications can contribute to a larger one.

RQ3: How can AR nuggets address location-specific content and tangible interactions?

1. What tangibles are suitable for being used in AR nuggets?
2. How can AR nuggets include location-specific content?

Section 3.3 described how realistically shaped tangibles and a universal tangible can be used for AR nuggets. We created instances for both tangible types in Section 4.2. However, our instance of a universal tangible shows only one way a universal tangible can look, and there can be more alternative versions for universal tangibles. In Section 6.2, we showed that realistically shaped tangibles are suitable for AR nuggets with a focus on 3D interactions, e.g., a *show & tell* AR nugget where users want to view a virtually labeled tangible from all sides. For AR nuggets with a focus on the AR device, e.g., a *progression* AR nugget where users focus on watching a virtual animation, our universal tangible was more suitable.

We used Vuforia's object tracking within AR nuggets to track the tangibles. To include new tangibles in an AR nugget, a virtual 3D model with the exact shape as the tangible is required. Alternatively, the tangible can be scanned. For robust recognition and tracking, the tangibles should be rigid, have sufficient geometric details, and have a colored or patterned surface. These requirements can limit what objects are suitable as tangibles, so not every 3D model is suitable as a realistically shaped tangible. For one realistically shaped tangible that was challenging to track, we improved the tracking quality by printing texture on paper and gluing it to the tangible.

AR nuggets need to track the user's physical environment to include location-specific content. Here, we utilized Vuforia's area targets and spatial anchors from the MRTK. We conceptualized and implemented three different ways how authors can anchor AR nuggets in the physical world (Section 5.2): a) grabbing and placing virtual elements individually, b) teleporting AR nuggets, and c) moving all virtual elements using a reference scan of the physical environment. These can also be combined, e.g., multiple virtual elements can be moved using option c) before they are individually placed more precisely using option a). By this, authors can make use of all options to make the placement process easier and faster.

The manual placement with one of these options utilizes spatial anchors from the MRTK to anchor virtual objects in the physical environment. To store the spatial anchors permanently across multiple applications and devices, we introduced tools that allow exporting and importing the spatial anchors. When authors export the spatial anchors, they can import them again, e.g., at the next start of the application, to automatically place all virtual objects. This can save time and effort because authors do not need to re-position virtual objects if they execute the AR application on another device or if an AR application shares similar spatial anchors with another AR application. However, authors need to manually copy and share the spatial anchors, e.g., using a USB stick, to synchronize spatial anchors between multiple devices. It is possible to support cloud-based solutions to share spatial anchors across devices to avoid using USB sticks. Yet, this requires an internet connection, and we built our prototypes to run without an internet connection.

For option c), AR nuggets can also automatically place virtual objects using Vuforia's Area Targets. However, the automatic placement requires a pre-processed scan of the environment in a format supported by Vuforia. Creating and pre-processing this scan requires a labor force and suitable scanning equipment. During runtime, the automatic placement requires more processing power than using spatial anchors, which can overload the AR device's processing power.

Overall, our answers to this RQ show that tangibles can be utilized with AR nuggets. Moreover, we contributed to filling the research gaps about how generic tangibles can be designed and how realistically shaped tangibles can robustly be tracked in combination with a generic tangible. Our answers to RQ3 also contribute to providing authors with suitable tools to place virtual objects in a physical environment.

RQ4: How can AR nuggets support authors in developing qualitative valuable AR applications?

1. What can the workflow for the authors look like?
2. What tools can support authors in adapting AR nuggets?

3. How can AR nuggets be delivered and executed on an AR device?

We contributed to answering RQ4 with Chapter 5. First, we developed an AR nugget authoring tool that we called ARNAUDDI (short for AR Nugget Authoring Using Different Degrees of Immersion) in Section 5.1. We implemented two versions of ARNAUDDI; one that targets desktop computers and one for HHDs. We implemented ARNAUDDI as a stand-alone authoring tool targeting desktop computers. Using ARNAUDDI, authors can select default AR nuggets and adapt them to create custom AR applications without programming. Adapting AR nuggets includes to a) add, remove, or replace 3D objects, b) add, remove, or replace targets where the 3D objects are augmented to, c) add, remove, or edit text labels, and d) move, rotate, and scale virtual objects. ARNAUDDI ensures that the AR nuggets always remain in an executable state. It only allows removing virtual objects that are not mandatory for the AR nugget. For example, a default AR nugget of the type *show & tell* incorporates one mandatory 3D object and one mandatory label. Authors can replace both or add more, but can only delete 3D objects as long as at least one 3D object remains. Moreover, ARNAUDDI provides a preview function that allows authors to experience their AR nuggets. In addition to the version targeting desktop computers, we also developed a version that targets HHDs. This version can extend and connect with the authoring tool executed on a desktop computer. It allows authors to experience AR nuggets immersively on a HHD. The HHD version of ARNAUDDI also includes selected authoring functionalities that adapt parameter values using 3D interactions, e.g., adapting positions or rotations. If authors adapt an AR nugget using one device, the two versions synchronize to make the changes visible on both devices, the HHD and the desktop computer.

We also introduced a workflow for adapting AR nuggets in Section 5.1. First, authors choose a default AR nugget they want to experience or adapt. As an optional second step, the authors can adapt the AR nugget. ARNAUDDI allows authors to perform the adaptions to the AR nugget in any order, but its order of UI elements suggests starting with adapting the 3D object(s) and anchor(s) before working on positions, rotations, scale, or text. Third, the authors can experience the AR nuggets. Because the AR nuggets remain in an executable state, authors can switch between adapting and experiencing as they like.

We evaluated ARNAUDDI and our authoring workflow in Section 6.6 with a user study. The participants rated the workflow in a neutral area on a scale from complicated to straightforward. For all three provided types of AR nuggets, the participants found that the order of the menus in the UI contributed to a straightforward workflow. The participants found starting with the provided default AR nuggets helpful. Moreover, they could get an impression of how their adapted AR nuggets could look like by experiencing the default AR nuggets for the AR nugget types *show & tell* and *quiz*. For *semantic zoom*, they rated this neutrally. The participants also found it important to be able to experience their AR nuggets at any time. However, some participants described unexpected behavior (bugs) from ARNAUDDI or that interactions were unclear to them. For example, ARNAUDDI supported touch input on the HHD, yet some authors described touch input as a new feature they would like. Here, our virtual assistant from Section 3.4 and Section 4.3

could be extended to support not only users of AR nuggets but also authors by providing hints to available authoring functionalities.

Next, Section 5.2 introduced authoring tools integrated into the Game Engine Unity. As Section 6.4 and Section 6.7 showed, these tools can support authors in a) placing and anchoring location-specific virtual elements in the physical world, b) deploying an author application with authoring functionalities and a user application without these in one step (*mode switcher* tool), c) saving and loading assets and parameter values during runtime, d) rotating virtual objects to any other object or the user, e) defining conditions and events. Our participants used all these tools without the need for programming.

In our user study from Section 6.7, participants could get an impression of how their adapted AR nuggets could look for all provided types of AR nuggets (*show & tell*, *quiz*, *semantic zoom*, *superimposition with interactive transparency control*), except for *compare*. Similar to participants for our evaluation of ARNAUDDI, the participants found starting with the provided default AR nuggets helpful. This shows that providing AR nuggets as ready-to-use, executable AR applications can support authors. Also, the participants could get an impression of how their adapted AR nuggets could look like by experiencing the default AR nuggets. However, they rated the importance of being able to experience the AR nuggets at any time in a neutral area, but slightly higher for the more interactive AR nuggets *semantic zoom* and *superimposition with interactive transparency control*. The participants rated the difficulty of authoring tasks as very simple for replacing placeholder objects and adding new labels. For other authoring tasks, they rated the difficulty as simple to neutral. However, one participant pointed out that the main challenge is not to use the authoring tool but to apply one's own ideas to the tools. To support authors in doing so, the tools could have more evident names and descriptions of how they work. One idea could be to not only extend ARNAUDDI but also to extend the integrated authoring tools with a virtual assistant as described in Section 3.4.

One participant from our user study in Section 6.7 pointed out that she would like to be able to define specific types of surfaces where virtual 3D objects are automatically placed, e.g., a chair or a couch. ARNAUDDI is limited to placing virtual objects on Vuforia image targets and our integrated tools can place virtual objects to any anchor in a room, but both authoring approaches do not support defining types of surfaces as a placement constraint. We explored an alternative authoring approach to place virtual objects in a yet unknown physical environment based on constraints in Section 5.3. With this, AR nuggets can categorize the surfaces present in the user's environment and re-construct a virtual scene based on this. Then, the AR nuggets can automatically place virtual objects on specific category types of surfaces. To define the constraints, we developed an authoring mode where an author can interactively place the virtual 3D objects in the room and have the AR nugget automatically calculate constraints. Besides the surface type, we introduced distance to the author and surface area as constraints. Our user study from Section 6.8 showed that AR nuggets mainly placed the virtual objects as the author expected, i.e., the AR nuggets calculated constraints as the author intended. However, the distance constraints were confusing for some participants. One option could be to allow authors to view and change the constraints, e.g., by providing a list with all available constraints. Moreover, constraints other than surface type, distance, and the surface area should also be considered, e.g., relation to other placed objects, placement in a corner, and fallback options if no suitable surface is available should be implemented.

Finally, this work presented a novel solution that is able to successfully deploy AR nuggets to AR devices in Section 5.4. The solution allows authors without programming knowledge to adapt AR nuggets using authoring tools like ARNAUDDI (Section 5.1) or integrated authoring tools (Section 5.2) and simultaneously allows programmers to adapt AR nuggets further using a game engine as a familiar environment. Our solution exports each AR nugget to an exchange file that can be loaded into the game engine Unity. From there, the AR nuggets can be exported as assetBundle and loaded into existing Unity-based applications, e.g., a CME learning application. Alternatively, the AR nuggets can be deployed as stand-alone applications to any platform that Unity supports. While this process also allows authors to adapt AR nuggets using Unity, it might be a challenge for persons who have never used Unity to import the AR nuggets into Unity and build an application from there.

Overall, our answers to this RQ contribute to lowering barriers to authoring applications and overcoming authoring challenges that we identified in Subsection 2.6.1. We contribute to filling the research gaps of how the challenges identified in Subsection 2.6.1 can be approached and overcome. Our contributions support authors in creating their own AR applications without requiring programming knowledge or expertise with AR.

Chapter 8

Conclusion and Future Work

This work introduced pattern-based components, called AR nuggets, for using and authoring AR experiences. It presented concepts for tangible interaction, user assistance, complex AR applications, and authoring. Our concepts and findings contribute to lowering barriers to using and authoring AR applications as follows. In total, we developed and evaluated a concept for AR nuggets as well as six protoypes (Utilization of Tangible Interactions in AR Nuggets, Integration of User Assistance in AR Nuggets, Combination of AR Nuggets and VR Nuggets, AR Nugget Authoring Using Different Degrees of Immersion, Integrated AR Nugget Authoring Tools, and Constraint-based Authoring with AR Nuggets). More precisely, the following list shows ten selected highlights of the contributions and findings from this thesis.

1. We developed a concept for AR nuggets that derives from concepts of microlearning and VR nuggets. The concept of AR nuggets expands existing nugget concepts by targeting challenges that the incorporation of the users' physical environment brings. For example, AR nuggets implement functions to detect and track physical objects in the user's environment or the environment itself. They also implement AR-specific interaction techniques for HMDs and HHDs.
2. We identified eight patterns with different variations and implemented AR nuggets from example scenarios from cultural heritage and educational settings. This shows that suitable patterns for AR nuggets exist.
3. We introduced a proactive virtual assistant that aids in using AR nuggets by identifying typical challenges and providing hints when the user faces one of the challenges. With the virtual assistant, authors do not need the knowledge to implement assistance functions on their own.
4. AR nuggets can contribute to complex applications consisting of multiple AR nuggets and other forms of applications, e.g., VR nuggets. Here, we developed transitions and introduced a conceptual model for transitioning between AR and VR. Our user study showed that participants do not feel disturbed by switching between AR and VR HMDs.

5. We introduced an AR nugget manager that controls the application flow when authors combine multiple AR nuggets into one larger non-linear AR experience. We also implemented three types of guiding methods within AR nuggets that target to navigate users from one physical location to another. The AR nugget manager and our navigation AR nuggets allow using AR nuggets in large, complex buildings and environments where multiple AR nuggets are widely distributed, e.g., on different floors and exhibits in a large museum.

6. This work integrated interactions with tangible objects within AR nuggets and developed different types of tangibles: shaped similar to a 3D model that augments them (realistically shaped), generic, and a combination of both. Our user study indicated that realistically shaped tangibles are especially supportive for AR nuggets that focus on 3D interactions, e.g., viewing a tangible in a *show & tell* AR nugget from all sides. However, realistically shaped tangibles are not always available, their production can be resources- and time-consuming, and some can be difficult to detect and track. Therefore, this work also developed a generic tangible. To enhance its handling, the generic tangible provides a handle that can support users in comfortably grabbing and holding it. In our study, users described that this tangible type is more suitable for the *sequence* AR nuggets because they focussed on the AR device's screen. Our combined tangible combined both tangible types to allow utilizing realistically shaped tangibles even if they are difficult to track and to simoultanesouly take advantage of the generic tangible's handle.

7. To support location-specific AR applications, we developed a *grabbable*, *teleporting*, and *place with scan* tool that help authors to place AR nuggets and virtual content in the physical world. Moreover, we introduced a *mode switcher* tool that supports authors in creating two separate applications: one application in authoring mode, where authors can place the AR nuggets in the real world, and one in user mode, where moving AR nuggets is restricted. Authors can use the *mode switcher* tool to switch between both modes with one click. Additionally, the *mode switcher* tool can automatically activate and deactivate other authoring functionalities or 3D objects that the author defined. To exchange real-world anchors between the applications and also between different AR devices, we introduced ex- and import tools for the real-world anchors.

8. This work developed and evaluated one stand-alone authoring tool (ARNAUDDI) for adapting AR nuggets. It can be executed on desktop computers and HHDs to supports different degrees of immersion. In our user study, participants primarily authored using their desktop devices. However, they found it helpful to use their AR device at any time during the authoring process to experience their authored application instantly.

9. For authors who want to make use of more authoring functionalities than ARNAUDDI can provide, we introduced and implemented multiple additional authoring tools integrated into a Game Engine. Similar to our stand-alone authoring tool ARNAUDDI, authors can use these integrated authoring tools to adapt AR nuggets without programming or scripting.

10. This work applied constraint-based authoring to AR nuggets. The constraints allow authors to control where AR nuggets place virtual objects without knowledge of the actual environment. We developed and implemented algorithms and decisions that use the positions of virtual objects in a room to calculate surface type, distance and surface area constraints. Moreover, we developed and implemented algorithms that automatically place virtual objects in a room based on the calculated constraints.

The individual discussion sections in Chapter 6 already pointed to some directions for future work. Overall, we see the following tasks and challenges that future work can cover.

Our authoring tools can be further refined based on the outcomes of our user studies. While in our expert study from Section 6.1, the medical experts support the idea of AR nuggets and saw potential for using AR nuggets in the future in their continuing education, the study did not evaluate the AR nuggets' impact on learning. It can be interesting to evaluate if and how AR nuggets can contribute to an enjoyable and successful learning experience. This can be evaluated not only for CME but also for education about other subjects. Similarly, it can be interesting to investigate how AR nuggets affect other aspects in any scenario, not only educational ones. For example, can AR nuggets contribute to making a museum visit more enjoyable?

This work applied AR nuggets to CME and the exhibition domain, but applying AR nuggets to further domains and evaluating their impact there remains open. There are likely further patterns that are typical for other domains and that can be transferred to other domains and implemented in AR nuggets. In future work, researchers can use our methods to identify and implement novel patterns as AR nuggets. Furthermore, researchers can build upon our work, add novel AR nuggets to our authoring tools, and evaluate them similarly to our evaluations.

To allow a broader community to access such novel default AR nuggets, one can make the AR nuggets accessible online. Here, developers and authors could not only share novel default AR nuggets but also custom adapted AR nuggets. Establishing a standardized AR nugget format as a common format for AR experiences can also be beneficial. Then, other AR authoring tools not specialized for AR nuggets, e.g., commercial authoring tools, can implement import and export functions to work with AR nuggets. For example, we developed a content delivery system to import AR nuggets into the Game Engine Unity. In future, other authoring tools or domain-specific software, e.g., PowerPoint or Learning Management Systems, can also be extended to import and include AR nuggets.

Appendix A

References

[AIS77] Christopher Alexander, Sara Ishikawa, and Murray Silverstein. *A pattern language: Towns, buildings, construction*. Vol. 2. Center for Environmental Structure series. New York, NY: Oxford University Press, 1977. ISBN: 0-19-501919-9. URL: http://www.loc.gov/catdir/enhancements/fy0603/74022874-d.html.

[arz] arztCME. *Augmented Reality als unterstützendes Element in einer CME-Fortbildung*. Online, accessed: 17.04.2024. URL: https://www.arztcme.de/kurse/endometriose-update-2022/.

[Ash+20] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. "Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities." In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. CHI '20. Honolulu, HI, USA: Association for Computing Machinery, 2020, 1–13. ISBN: 9781450367080. DOI: 10.1145/3313831.3376722.

[AYPVCD42] Yuliana Apaza-Yllachura, Alfredo Paz-Valderrama, and Carlo Corrales-Delgado. "SimpleAR: Augmented Reality high-level content design framework using visual programming." In: *2019 38th International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 11/4/2019 - 11/9/2019, pp. 1–7. ISBN: 978-1-7281-5613-2. DOI: 10.1109/SCCC49216.2019.8966427.

[Azu97] Ronald T. Azuma. "A Survey of Augmented Reality." In: *Presence: Teleoperators and Virtual Environments*. Ed. by Eugene Ch'ng and Albert Rizzo. Vol. 6. 1997, pp. 355–385.

[Bai+06] Chris Bailey, Mohd T Zalfan, Hugh C Davis, Karen Fill, and Gráinne Conole. "Panning for gold: Designing pedagogically inspired learning nuggets." In: *Journal of Educational Technology and Society* 9, No. 1 (2006), pp. 113–122.

[BD12] Mark Billinghurst and Andreas Duenser. "Augmented Reality in the Classroom." In: *Computer* 45.7 (2012), pp. 56–63. ISSN: 0018-9162. DOI: 10.1109/MC.2012.111.

[Bea+07] Jennifer S. Beaudin, Stephen S. Intille, Emmanuel Munguia Tapia, Randy Rockinson, and Margaret E. Morris. “Context-Sensitive Microlearning of Foreign Language Vocabulary on a Mobile Device.” In: *Ambient intelligence*. Ed. by Bernt Schiele. Vol. 4794. Lecture Notes in Computer Science. Berlin: Springer, 2007, pp. 55–72. ISBN: 978-3-540-76651-3. DOI: 10.1007/978-3-540-76652-0_4.

[Ber+12] Joseph Bergin, Jutta Eckstein, Markus Volter, Marianna Sipos, Eugene Wallingford, Klaus Marquardt, Jane Chandler, Helen Sharp, and Mary Lynn Manns, eds. *Pedagogical Patterns: Advice For Educators*. Joseph Bergin Software Tools, 2012. ISBN: 978-0-9851543-5-6.

[Ber18] Lucas Bertolini. *Hands-On Game Development without Coding: Create 2D and 3D games with Visual Scripting in Unity*. 1st ed. Birmingham: Packt Publishing Limited, 2018. ISBN: 9781789537987.

[BH99] Hugh Beyer and Karen Holtzblatt. “Contextual design.” In: *Interactions* 6.1 (1999), pp. 32–42. ISSN: 10725520. DOI: 10.1145/291224.291229.

[Bit+22] Jessica L. Bitter, Ralf Dörner, Yu Liu, Linda Rau, and Ulrike Spierling. “Follow the Blue Butterfly – An Immersive Augmented Reality Museum Guide.” In: *HCI International 2022 Posters*. Ed. by Constantine Stephanidis, Margherita Antona, and Stavroula Ntoa. Vol. 1582. Springer eBook Collection. Cham: Springer International Publishing and Imprint Springer, 2022, pp. 171–178. ISBN: 978-3-031-06390-9. DOI: 10.1007/978-3-031-06391-6_23.

[BK] Bernd Bruegge and Gudrun Klinker. *DWARF Distributed Wearable Augmented Reality Framework*. Online, accessed: 17.01.2020. Technische Universität München. URL: http://campar.in.tum.de/twiki/pub/Chair/DwarfWhitePaper/dwarf.pdf.

[BKP08] Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. “Tangible Augmented Reality.” In: *SIGGRAPH Asia ’08: ACM SIGGRAPH ASIA 2008 courses*. Singapore: Association for Computing Machinery, 2008. ISBN: 9781450379243. DOI: 10.1145/1508044.

[Bob+19] Răzvan Boboc, Mihai Duguleană, Gheorghe-Daniel Voinea, Cristian-Cezar Postelnicu, Dorin-Mircea Popovici, and Marcello Carrozzino. “Mobile Augmented Reality for Cultural Heritage: Following the Footsteps of Ovid among Different Locations in Europe.” In: *Sustainability* 11.4 (2019), p. 1167. DOI: 10.3390/su11041167.

[BRA19] Vasileios Bachras, George E. Raptis, and Nikolaos M. Avouris. “On the Use of Persistent Spatial Points for Deploying Path Navigation in Augmented Reality: An Evaluation Study.” In: *Human-Computer Interaction – INTERACT 2019*. Ed. by David Lamas, Fernando Loizides, Lennart Nacke, Helen Petrie, Marco Winckler, and Panayiotis Zaphiris. Vol. 11749. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 309–318. ISBN: 978-3-030-29389-5. DOI: 10.1007/978-3-030-29390-1_17.

[Bro+19] Wolfgang Broll, Florian Weidner, Tobias Schwandt, Kai Weber, and Ralf Dörner. "Authoring von VR/AR-Anwendungen." In: *Virtual und Augmented Reality (VR/AR): Grundlagen und Methoden der Virtuellen und Augmentierten Realität*. Ed. by Ralf Dörner, Wolfgang Broll, Paul Grimm, and Bernhard Jung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 393–423. ISBN: 978-3-662-58861-1. DOI: 10.1007/978-3-662-58861-1_10.

[But+92] Jeff Butterworth, Andrew Davidson, Stephen Hench, and Marc. T. Olano. "3DM: a three dimensional modeler using a head-mounted display." In: *Proceedings of the 1992 symposium on Interactive 3D graphics - SI3D '92*. New York, New York, USA: ACM Press, 1992. DOI: 10.1145/147156.147182.

[Cam+14] Abraham G. Campbell, John W. Stafford, Thomas Holz, and G. M. P. O'Hare. "Why, when and how to use augmented reality agents (AuRAs)." In: *Virtual Reality* 18.2 (2014), pp. 139–159. ISSN: 1359-4338. DOI: 10.1007/s10055-013-0239-4.

[CCJ10] Chien Huan Chien, Chien Hsu Chen, and Tay Sheng Jeng. "An interactive augmented reality system for learning anatomy structure." In: Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010. 2010, pp. 370–375. ISBN: 9789881701282.

[CES22] Robbe Cools, Augusto Esteves, and Adalberto L. Simeone. "Blending Spaces: Cross-Reality Interaction Techniques for Object Transitions Between Distinct Virtual and Augmented Realities." In: *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022. DOI: 10.1109/ismar55827.2022.00069.

[Che+17] Xiaoou Chen, Fei Wang, Zhenwei You, Xiaochun Wang, Chunjing Tao, and Jian Liu. "Design of Interactive Tutorials on Mobile Applications for Chinese Middle-Aged and Older Adults." In: *Art and Design Review* 05.03 (2017), pp. 162–180. ISSN: 2332-1997. DOI: 10.4236/adr.2017.53013.

[Cop98] James O. Coplien. "Software Design Patterns: Common Questions and Answers." In: *The Patterns Handbooks: Techniques, Strategies, and Applications*. USA: Cambridge University Press, 1998, 311–319. ISBN: 0521648181.

[Den+21] Andreas Dengel, Josef Buchner, Miriam Mulders, and Johanna Pirker. "Beyond the Horizon: Integrating Immersive Learning Environments in the Everyday Classroom." In: *Proceedings of 2021 7th International Conference of the Immersive Learning Research Network (iLRN)*. Ed. by Daphne Economou. Piscataway, NJ: IEEE, 2021, pp. 1–5. ISBN: 978-1-7348995-2-8. DOI: 10.23919/iLRN52045.2021.9459368.

[Den+22] Andreas Dengel, Muhammad Zahid Iqbal, Silke Grafe, and Eleni Mangina. "A Review on Augmented Reality Authoring Toolkits for Education." In: *Frontiers in Virtual Reality* 3 (2022). DOI: 10.3389/frvir.2022.798032.

[DLB15] Dragoş Datcu, Stephan Lukosch, and Frances Brazier. "On the Usability and Effectiveness of Different Interaction Types in Augmented Reality." In: *International Journal of Human-Computer Interaction* 31.3 (2015), pp. 193–209. ISSN: 1044-7318. DOI: 10.1080/10447318.2014.994193.

[Dör+03] Ralf Dörner, Christian Geiger, Michael Haller, and Volker Paelke. "Authoring Mixed Reality - A Component and Framework Based Approach." In: *Entertainment Computing*. Ed. by Ryohei Nakatsu and Junichi Hoshino. Vol. 112. IFIP Advances in Information and Communication Technology. Boston, MA: Springer US, 2003, E1–E1. ISBN: 978-1-4757-5153-6. DOI: 10.1007/978-0-387-35660-0_65.

[dTC19] da Silva, Manoela M. O., Veronica Teichrieb, and Patricia Smith Cavalcante. "What are Teacher's Needs Concerning Augmented Reality Digital Authoring Tools?" In: *Congresso Brasileiro de Informática na Educação (CBIE 2019)*. Vol. VIII. 2019.

[Eas+08] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. "Selecting Empirical Methods for Software Engineering Research." In: *Guide to Advanced Empirical Software Engineering*. Ed. by Forrest Shull, Janice Singer, and Dag I. K. Sjøberg. London: Springer London, 2008, pp. 285–311. ISBN: 978-1-84800-043-8. DOI: 10.1007/978-1-84800-044-5_11.

[Efs+20] Efstratios Geronikolakis, Paul Zikas, Steve Kateros, Nick Lydatakis, Stelios Georgiou, Mike Kentros, and George Papagiannakis. "A True AR Authoring Tool for Interactive Virtual Museums." In: *Visual Computing for Cultural Heritage*. Springer, Cham, 2020, pp. 225–242. DOI: 10.1007/978-3-030-37191-3_12.

[EKH17] Felix Emmerich, Roland Klemke, and Thomas Hummes. "Design Patterns for Augmented Reality Learning Games." In: *Games and Learning Alliance*. Ed. by João Dias, Pedro A. Santos, and Remco C. Veltkamp. Vol. 10653. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2017, pp. 161–172. ISBN: 978-3-319-71939-9. DOI: 10.1007/978-3-319-71940-5_15.

[ESE06] Mike Eissele, Oliver Siemoneit, and Thomas Ertl. "Transition of Mixed, Virtual, and Augmented Reality in Smart Production Environments - An Interdisciplinary View." In: *2006 IEEE Conference on Robotics, Automation and Mechatronics*. IEEE, 2006, pp. 1–6. ISBN: 1-4244-0024-4. DOI: 10.1109/RAMECH.2006.252671.

[Fel+19] Nico Feld, Pauline Bimberg, Benjamin Weyers, and Daniel Zielasko. "Keep it simple? Evaluation of Transitions in Virtual Reality." In: *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. Ed. by Albrecht Schmidt, Kaisa Väänänen, Tesh Goyal, Per Ola Kristensson, and Anicia Peters. New York, NY, USA: ACM, 4192023, pp. 1–7. ISBN: 9781450394222. DOI: 10.1145/3544549.3585811.

[FGF11] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. "Sketch-sketch revolution." In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. Ed. by Jeff Pierce. ACM Conferences. New York, NY: ACM, 2011, pp. 373–382. ISBN: 9781450307161. DOI: 10.1145/2047196.2047245.

[Fia05] Mark Fiala. “ARTag, a Fiducial Marker System Using Digital Techniques.” In: *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Volume 02*. CVPR ’05. USA: IEEE Computer Society, 2005, 590–596. ISBN: 0769523722. DOI: `10.1109/CVPR.2005.74`.

[Fit+13] Elizabeth FitzGerald, Rebecca Ferguson, Anne Adams, Mark Gaved, Yishay Mor, and Rhodri Thomas. “Augmented Reality and Mobile Learning.” In: *International Journal of Mobile and Blended Learning* 5.4 (2013), pp. 43–58. ISSN: 1941-8647. DOI: `10.4018/ijmbl.2013100103`.

[Gal+14] Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. “FLARE: Fast layout for augmented reality applications.” In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2014)*. Ed. by Simon Julier. Piscataway, NJ: IEEE, 2014, pp. 207–212. ISBN: 978-1-4799-6184-9. DOI: `10.1109/ISMAR.2014.6948429`.

[Gam+95] Erich Gamma, Richard Helm, Ralph Johnson, and John M. Vlissides. *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley professional computing series. Reading, Massachusetts: Addison-Wesley, 1995. ISBN: 0201633612.

[Gas+02] Danilo Gasques, Janet G. Johnson, Tommy Sharkey, and Nadir Weibel. “What You Sketch Is What You Get: Quick and Easy Augmented Reality Prototyping with PintAR.” In: *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. Ed. by Stephen Brewster, Geraldine Fitzpatrick, Anna Cox, and Vassilis Kostakos. New York, NY, USA: ACM, 5022019, pp. 1–6. ISBN: 9781450359719. DOI: `10.1145/3290607.3312847`.

[Ger+20] Efstratios Geronikolakis, Paul Zikas, Steve Kateros, Nick Lydatakis, Stelios Georgiou, Mike Kentros, and George Papagiannakis. “A True AR Authoring Tool for Interactive Virtual Museums.” In: *Visual Computing for Cultural Heritage*. Ed. by Fotis Liarokapis, Athanasios Voulodimos, Nikolaos Doulamis, and Anastasios Doulamis. Cham: Springer International Publishing, 2020, pp. 225–242. ISBN: 978-3-030-37191-3. DOI: `10.1007/978-3-030-37191-3_12`. URL: `https://doi.org/10.1007/978-3-030-37191-3_12`.

[Ghi+09] Giuseppe Ghiani, Fabio Paternò, Carmen Santoro, and Lucio Davide Spano. “UbiCicero: A location-aware, multi-device museum guide.” In: *Interacting with Computers* 21.4 (2009), pp. 288–303. ISSN: 09535438. DOI: `10.1016/j.intcom.2009.06.001`.

[Gib77] James J. Gibson. “The theory of affordances.” In: *Perceiving, acting, and knowing: toward an ecological psychology*. Ed. by Robert E Shaw, John Bransford. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1977, p67–82.

[GK17] Yiannis Georgiou and Eleni A. Kyza. “The development and validation of the ARI questionnaire: An instrument for measuring immersion in location-based augmented reality settings.” In: *International Journal of Human-Computer Studies* 98 (2017), pp. 24–37. ISSN: 10715819. DOI: 10.1016/j.ijhcs.2016.09.014.

[GM14] Maribeth Gandy and Blair MacIntyre. “Designer’s augmented reality toolkit, ten years later.” In: *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST ’14*. Ed. by Hrvoje Benko, Mira Dontcheva, and Daniel Wigdor. New York, New York, USA: ACM Press, 2014, pp. 627–636. ISBN: 9781450330695. DOI: 10.1145/2642918.2647369.

[GM19] Renan Luigi Martins Guarese and Anderson Maciel. “Development and Usability Analysis of a Mixed Reality GPS Navigation Application for the Microsoft HoloLens.” In: *Advances in Computer Graphics*. Ed. by Marina Gavrilova, Jian Chang, and Nadia Magnenat Thalmann. Vol. 11542. Image Processing, Computer Vision, Pattern Recognition, and Graphics. Cham: Springer International Publishing, 2019, pp. 431–437. ISBN: 978-3-030-22514-8. DOI: 10.1007/978-3-030-22514-8_41.

[GMF10] Tovi Grossman, Justin Matejka, and George Fitzmaurice. “Chronicle: capture, exploration, and playback of document workflow histories.” In: *Proceedings of the 23nd annual ACM symposium on User interface software and technology*. Ed. by Ken Perlin, Mary Czerwinski, and Rob Miller. New York, NY, USA: ACM, 2010, pp. 143–152. ISBN: 9781450302715. DOI: 10.1145/1866029.1866054.

[Gra+09] Floraine Grabler, Maneesh Agrawala, Wilmot Li, Mira Dontcheva, and Takeo Igarashi. “Generating photo manipulation tutorials by demonstration.” In: *ACM Transactions on Graphics* 28.3 (2009), pp. 1–9. ISSN: 07300301. DOI: 10.1145/1531326.1531372.

[GW17] Jodie M. Gil and Vern Wiliams. “Byte-Sized Learning: A Review of Video Tutorial Engagement in a Digital Media Skills Course.” In: *Journalism and mass communication* (2017).

[Ha+10] Taejin Ha, Woontack Woo, Youngho Lee, Junhun Lee, Jeha Ryu, Hankyun Choi, and Kwanheng Lee. “ARtalet: Tangible User Interface Based Immersive Augmented Reality Authoring Tool for Digilog Book.” In: *International Symposium on Ubiquitous Virtual Reality (ISUVR), 2010*. Ed. by Choongsung Shin. Piscataway, NJ: IEEE, 2010, pp. 40–43. ISBN: 978-1-4244-7702-9. DOI: 10.1109/ISUVR.2010.20.

[Ham+06] Alastair Hampshire, Hartmut Seichter, Raphaël Grasset, and Mark Billinghurst. “Augmented Reality Authoring: Generic Context from Programmer to Designer.” In: *Proceedings of the 18th Australia conference on Computer-Human Interaction Design Activities, Artefacts and Environments*. Ed. by Toni Robertson. New York, NY: ACM, 2006, p. 409. ISBN: 1595935452. DOI: 10.1145/1228175.1228259.

[Ham+20] Ramy Hammady, Minhua Ma, Carl Strathern, and Mostafa Mohamad. “Design and development of a spatial mixed reality touring guide to the Egyptian museum.” In: *Multimedia Tools and Applications* 79.5-6 (2020), pp. 3465–3494. ISSN: 1380-7501. DOI: `10.1007/s11042-019-08026-w`.

[Har+16] Tilo Hartmann, Werner Wirth, Holger Schramm, Christoph Klimmt, Peter Vorderer, André Gysbers, Saskia Böcking, Niklas Ravaja, Jari Laarni, Timo Saari, Feliz Gouveia, and Ana Maria Sacau. “The Spatial Presence Experience Scale (SPES).” In: *Journal of Media Psychology* 28.1 (2016), pp. 1–15. ISSN: 1864-1105. DOI: `10.1027/1864-1105/a000137`.

[HBK03] Marc Hassenzahl, Michael Burmester, and Franz Koller. “AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität.” In: *Mensch & Computer 2003: Interaktion in Bewegung*. Ed. by Gerd Szwillus and Jürgen Ziegler. Wiesbaden: Vieweg+Teubner Verlag, 2003, pp. 187–196. ISBN: 978-3-322-80058-9. DOI: `10.1007/978-3-322-80058-9_19`.

[HD19a] Robin Horst and Ralf Dörner. “Integration of Bite-Sized Virtual Reality Applications into Pattern-Based Knowledge Demonstration.” In: *16. Workshop der GI-Fachgruppe VR/AR*. Ed. by P. Grimm, Y. Jung, R. Dörnber, and C. Geiger. Virtuelle und erweiterte Realität. 2019. ISBN: 3844068872.

[HD19b] Robin Horst and Ralf Dörner. “Mining Virtual Reality Nuggets: A Pattern-Based Approach for Creating Virtual Reality Content Based on Microlearning Methodology.” In: *IEEE International Conference on Teaching, Assessment and Learning for Engineering (TALE)* (2019).

[HD19c] Robin Horst and Ralf Dörner. “Virtual Reality Forge: Pattern-Oriented Authoring of Virtual Reality Nuggets.” In: *25th ACM Symposium on Virtual Reality Software and Technology on - VRST ’19*. Ed. by Tomas Trescak, Richard Skarbez, Anton Bogdanovych, Martin Masek, Simeon Simoff, Deborah Richards, Thierry Duval, Torsten Kuhlen, Huyen Nguyen, Shigeo Morishima, and Yuichi Itoh. New York, New York, USA: ACM Press, 2019, pp. 1–12. ISBN: 9781450370011. DOI: `10.1145/3359996.3364261`.

[HF08] Steven J. Henderson and Steven Feiner. “Opportunistic controls.” In: *Proceedings of the 2008 ACM symposium on Virtual reality software and technology - VRST ’08*. Ed. by Steven Feiner, Daniel Thalmann, Pascal Guitton, Bernd Fröhlich, Ernst Kruijff, and Martin Hachet. New York, New York, USA: ACM Press, 2008, p. 211. ISBN: 9781595939517. DOI: `10.1145/1450579.1450625`.

[HF10] Steven Henderson and Steven Feiner. “Opportunistic tangible user interfaces for augmented reality.” In: *IEEE transactions on visualization and computer graphics* 16.1 (2010), pp. 4–16. DOI: `10.1109/TVCG.2009.91`.

[HNR68] Peter Hart, Nils Nilsson, and Bertram Raphael. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths.” In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107. ISSN: 0536-1567. DOI: `10.1109/tssc.1968.300136`.

[Hor+20] Robin Horst, Ramtin Naraghi-Taghi-Off, Linda Rau, and Ralf Dorner. “Bite-Sized Virtual Reality Learning Applications: A Pattern-Based Immersive Authoring Environment.” In: *JUCS - Journal of Universal Computer Science* 26.8 (2020), pp. 947–971. ISSN: 0948-6968. DOI: `10.3897/jucs.2020.051`.

[Hor21] Robin Horst. “Virtual Reality Nuggets - Authoring and Usage of Concise and Pattern-Based Educational Virtual Reality Experiences.” PhD thesis. Hochschule RheinMain, 2021. DOI: `10.25716/PUR-9`.

[Hor+21a] Robin Horst, Dennis Fenchel, Reimond Retz, Linda Rau, Wilhelm Retz, and Ralf Dörner. *Integration of Game Engine Based Mobile Augmented Reality Into a Learning Management System for Online Continuing Medical Education*. INFORMATIK 2020. 2021. DOI: `10.18420/inf2020_88`.

[Hor+21b] Robin Horst, Ramtin Naraghi-Taghi-Off, Linda Rau, and Ralf Dörner. “Back to reality: transition techniques from short HMD-based virtual experiences to the physical world.” In: *Multimedia Tools and Applications* (2021). ISSN: 1380-7501. DOI: `10.1007/s11042-021-11317-w`.

[Hor+21c] Robin Horst, Ramtin Naraghi-Taghi-Off, Linda Rau, and Ralf Dörner. “Design Patterns and Author Roles for Pattern-Based Educational Virtual Reality Experiences.” In: *Proceedings of the 13th Workshop Virtual and Augmented Reality of the GI Group VR/AR (Shaker Verlag)*. 2021. DOI: `10.18420/vrar2021_3`.

[Hor+21d] Robin Horst, Linda Rau, Lars Dieter, Manuel Feller, Jonas Gaida, Andreas Leipe, Julian Eversheim, Julia Wirth, Jörn Bachmeier, Julius Müller, Maik Melcher, and Ralf Dörner. “Presenters in Virtual Reality in Slideshow Presentations.” In: *INFORMATIK 2020*. Ed. by Reussner, R. H., Koziolek, A. & Heinrich, R. Bonn, 2021, pp. 963–970. DOI: `10.18420/INF2020_89`.

[Hor+22a] Robin Horst, Simon Gerstmeier, Ramtin Naraghi-Taghi-Off, Julian Wagner, Linda Rau, and Ralf Dörner. “Virtual reality content creation based on self-contained components in the e-learning domain: Re-using pattern-based vr content in different authoring toolkits.” In: *Multimedia Tools and Applications* (2022). ISSN: 1380-7501. DOI: `10.1007/s11042-022-13362-5`.

[Hor+22b] Robin Horst, Ramtin Naraghi-Taghi-Off, Linda Rau, and Ralf Doerner. “Authoring With Virtual Reality Nuggets - Lessons Learned.” In: *Frontiers in Virtual Reality* 3 (2022). DOI: `10.3389/frvir.2022.840729`.

[Hug05] Theo Hug. “Micro Learning and Narration Exploring possibilities of utilization of narrations and storytelling for the designing of micro units and didactical micro-learning arrangements.” In: *Proceedings of Media in Transition*. 2005.

[HW16] Anuruddha Hettiarachchi and Daniel Wigdor. "Annexing Reality: Enabling Opportunistic Use of Everyday Objects as Tangible Proxies in Augmented Reality." In: CHI '16. San Jose, California, USA: Association for Computing Machinery, 2016, 1957–1967. ISBN: 9781450333627. DOI: 10.1145/2858036.2858134.

[IMT06] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. "Teddy." In: *ACM SIGGRAPH 2006 Courses on - SIGGRAPH '06*. New York, New York, USA: ACM Press, 2006. DOI: 10.1145/1185657.1185772.

[Iva+01] Ivan Poupyrev, Desney Tan, Mark Billinghurst, Hirokazu Kato, Holger Regenbrecht, and Nobuji Tetsutani. "Tiles: A Mixed Reality Authoring Interface." In: *INTERACT Conference on Human Computer Interaction* (2001).

[Jee+14] Hyung-Keun Jee, Sukhyun Lim, Jinyoung Youn, and Junsuk Lee. "An augmented reality-based authoring tool for E-learning applications." In: *Multimedia Tools and Applications* 68.2 (2014), pp. 225–235. ISSN: 1380-7501. DOI: 10.1007/s11042-011-0880-4.

[Jim+15] B. Jiménez Fernández-Palacios, F. Nex, A. Rizzi, and F. Remondino. "ARCube-The Augmented Reality Cube for Archaeology." In: *Archaeometry* 57 (2015), pp. 250–262. ISSN: 0003813X. DOI: 10.1111/arcm.12120.

[JK16] Bret Jackson and Daniel F. Keefe. "Lift-Off: Using Reference Imagery and Freehand Sketching to Create 3D Models in VR." In: *IEEE transactions on visualization and computer graphics* 22.4 (2016), pp. 1442–1451. DOI: 10.1109/tvcg.2016.2518099.

[Jun+16] Timothy Jung, M. Claudia tom Dieck, Hyunae Lee, and Namho Chung. "Effects of Virtual Reality and Augmented Reality on Visitor Experiences in Museum." In: *Information and Communication Technologies in Tourism 2016*. Ed. by Alessandro Inversini and Roland Schegg. Cham: Springer International Publishing, 2016, pp. 621–635. ISBN: 978-3-319-28230-5. DOI: 10.1007/978-3-319-28231-2_45.

[JWH18] Sungchul Jung, Pamela J. Wisniewski, and Charles E. Hughes. "In Limbo: The Effect of Gradual Visual Transition Between Real and Virtual on Virtual Body Ownership Illusion and Presence." In: *25th IEEE Conference on Virtual Reality and 3D User Interfaces*. Ed. by Kiyoshi Kiyokawa. Piscataway, NJ: IEEE, 2018, pp. 267–272. ISBN: 978-1-5386-3365-6. DOI: 10.1109/VR.2018.8447562.

[Kam+14] Carolien Kamphuis, Esther Barsom, Marlies Schijven, and Noor Christoph. "Augmented reality in medical education?" In: *Perspectives on medical education* 3.4 (2014), pp. 300–311. ISSN: 2212-2761. DOI: 10.1007/s40037-013-0107-7.

[Kat+03] H. Kato, K. Tachibana, M. Tanabe, T. Nakajima, and Y. Fukuda. "MagicCup: A Tangible Interface for Virtual Objects Manipulation in Table-top Augmented Reality." In: *IEEE International Augmented Reality Toolkit Workshop (ART)*. Piscataway, N.J: IEEE, 2003, pp. 75–76. DOI: 10.1109/ART.2003.1320434.

[KB97] Taeyong Kim and Frank Biocca. “Telepresence via Television: Two Dimensions of Telepresence May Have Different Connections to Memory and Persuasion.” In: *Journal of Computer-Mediated Communication* 3.2 (Sept. 1997). ISSN: 1083-6101. DOI: `10.1111/j.1083-6101.1997.tb00073.x`.

[Kİ20] Gökhan KURT and Gökhan İNCE. “ARgent: A Web Based Augmented Reality Framework for Dynamic Content Generation.” In: *European Journal of Science and Technology* (2020), pp. 244–257. DOI: `10.31590/ejosat.779946`.

[Kim+14] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller, and Krzysztof Z. Gajos. “Crowdsourcing step-by-step information extraction to enhance existing how-to videos.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Ed. by Matt Jones, Philippe Palanque, Albrecht Schmidt, and Tovi Grossman. New York, NY, USA: ACM, 2014, pp. 4017–4026. ISBN: 9781450324731. DOI: `10.1145/2556288.2556986`.

[KKL09] Eun Kwon, Gerard J. Kim, and Sangyoon Lee. “Effects of sizes and shapes of props in tangible augmented reality.” In: *8th IEEE International Symposium on Mixed and Augmented Reality*. 2009, pp. 201–202. DOI: `10.1109/ISMAR.2009.5336463`.

[Kli+02] G. Klinker, A.H. Dutoit, M. Bauer, J. Bayer, V. Novak, and D. Matzke. “Fata Morgana - a presentation system for product design.” In: *Proceedings of the International Symposium on Mixed and Augmented Reality*. 2002, pp. 76–85. DOI: `10.1109/ISMAR.2002.1115076`.

[Kot19] Konstantinos Kotis. “A reAl-time low-effoRt mulTi-entity Interaction System for creaTing reusable and optimized MR experiences.” In: *Research Ideas and Outcomes* 5 (2019). DOI: `10.3897/rio.5.e36128`.

[KP21] Thomas Klauer and Bastian Plaß. “Point Cloud Capturing and AI-based Classification for as-built BIM using Augmented Reality.” In: *Artificial Intellgence - Application in Life Sciences and Beyond. The Upper Rhine Artificial Intelligence Symposium UR-AI 2021*. 2021, pp. 158–166.

[Kra+21] Veronika Krauß, Alexander Boden, Leif Oppermann, and René Reiners. “Current Practices, Challenges, and Design Implications for Collaborative AR/VR Application Development.” In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Ed. by Yoshifumi Kitamura. ACM Digital Library. New York,NY,United States: Association for Computing Machinery, 2021, pp. 1–15. ISBN: 9781450380966. DOI: `10.1145/3411764.3445335`.

[KS17] Antonia Kampa and Ulrike Spierling. “Smart Authoring for Location-based Augmented Reality Storytelling Applications.” In: *INFORMATIK 2017*. SENSYBLE – Smart Systems for Better Living Environments (Sept. 25–29, 2017). Gesellschaft für Informatik. Bonn, 2017, pp. 915–922. ISBN: 978-3-88579-669-5. DOI: `10.18420/in2017_93`.

[Lav+21] Enricoandrea Laviola, Michele Gattullo, Vito Modesto Manghisi, Michele Fiorentino, and Antonio Emmanuele Uva. "Minimal AR: visual asset optimization for the authoring of augmented reality work instructions in manufacturing." In: *The International journal, advanced manufacturing technology* (2021), pp. 1–16. ISSN: 0268-3768. DOI: 10.1007/s00170-021-08449-6.

[LB19] Wallace S. Lages and Doug A. Bowman. "Walking with adaptive augmented reality workspaces." In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*. Ed. by Wai-Tat Fu. ACM Conferences. New York, NY: ACM, 2019, pp. 356–366. ISBN: 9781450362726. DOI: 10.1145/3301275.3302278.

[LCN19] Soralak Leela, Sasithorn Chookeaw, and Prachyanun Nilsook. "An Effective Microlearning Approach Using Living Book to Promote Vocational Students' Computational Thinking." In: *Proceedings of the 2019 The 3rd International Conference on Digital Technology in Education*. ACM Digital Library. New York, NY, United States: Association for Computing Machinery, 2019, pp. 25–29. ISBN: 9781450372206. DOI: 10.1145/3369199.3369200.

[LD97] Matthew Lombard and Theresa Ditton. "At the Heart of It All: The Concept of Presence." In: *Journal of Computer-Mediated Communication* 3.2 (Sept. 1997). JCMC321. ISSN: 1083-6101. DOI: 10.1111/j.1083-6101.1997.tb00072.x.

[Lee+04] G. A. Lee, C. Nelles, M. Billinghurst, and G. J. Kim. "Immersive Authoring of Tangible Augmented Reality Applications." In: *Third IEEE and ACM International Symposium on Mixed and Augmented Reality, 2004*. Los Alamitos, California: IEEE Computer Society, 2004, pp. 172–181. ISBN: 0-7695-2191-6. DOI: 10.1109/ISMAR.2004.34.

[LK09] Gun A. Lee and Gerard J. Kim. "Immersive authoring of Tangible Augmented Reality content: A user study." In: *Journal of Visual Languages & Computing* 20.2 (2009), pp. 61–79. ISSN: 1045-926X. DOI: 10.1016/j.jvlc.2008.07.001.

[LKB05] Gun A. Lee, Gerard J. Kim, and Mark Billinghurst. "Immersive authoring: What You eXperience Is What You Get (WYXIWYG)." In: *Communications of the ACM* 48.7 (2005), pp. 76–81. ISSN: 00010782. DOI: 10.1145/1070838.1070840.

[LKP02] Gun A. Lee, Gerard Jounghyun Kim, and Chan-Mo Park. "Modeling virtual object behavior within virtual environment." In: *Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA: ACM, 2002. DOI: 10.1145/585740.585748.

[LM97] F. Lu and E. Milios. "Globally Consistent Range Scan Alignment for Environment Mapping." In: *Autonomous Robots* 4.4 (1997), pp. 333–349. ISSN: 09295593. DOI: 10.1023/A:1008854305733.

[LRS10] Jae Yeol Lee, Gue Won Rhee, and Dong Woo Seo. "Hand gesture-based tangible interactions for manipulating virtual objects in a mixed reality environment." In:

The International journal, advanced manufacturing technology 51.9-12 (2010), pp. 1069–1082. ISSN: 0268-3768. DOI: 10.1007/s00170-010-2671-x.

[LTK18] Chris Lytridis, Avgoustos Tsinakos, and Ioannis Kazanidis. "ARTutor - An Augmented Reality Platform for Interactive Distance Learning." In: *Education Sciences* 8.1 (2018), p. 6. DOI: 10.3390/educsci8010006.

[Lüc+20] Thomas Lück, Hans Nopper, Olav Schendel, Dirk Weyhe, Daniela Salzmann, Verena Uslar, Anke V. Reinschluessel, Tanja Döring, Thomas Muender, Rainer Malaka, Andrea Schenk, and Christian Schumann. "Exploring realistic haptics for 3D-printed organ phantoms in surgery training in VR and AR." In: *Transactions on Additive Manufacturing Meets Medicine* Vol 2 No 1 (2020). DOI: 10.18416/AMMM.2020.2009026.

[Mac+03] B. MacIntyre, M. Gandy, J. Bolter, S. Dow, and B. Hannigan. "DART: the Designer's Augmented Reality Toolkit." In: *The second IEEE and ACM International Symposium on Mixed and Augmented Reality*. Los Alamitos, California: IEEE Computer Society, 2003, pp. 329–330. ISBN: 0-7695-2006-5. DOI: 10.1109/ISMAR.2003.1240744.

[Mac+04a] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. "DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences." In: *Proceedings of the 17th annual ACM symposium on User interface software and technology*. Ed. by Steven K. Feiner. New York, NY: ACM, 2004, p. 197. ISBN: 1581139578. DOI: 10.1145/1029632.1029669.

[Mac+04b] Asa MacWilliams, Thomas Reicher, Gudrun Klinker, and Bernd Bruegge. "Design Patterns for Augmented Reality Systems." In: *MIXER Exploring the Design and Engineering of Mixed Reality Systems*. CEUR Workshop Proceedings. 2004.

[MGF11] Justin Matejka, Tovi Grossman, and George Fitzmaurice. "Ambient help." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Ed. by Desney Tan. ACM Conferences. New York, NY: ACM, 2011, pp. 2751–2760. ISBN: 9781450302289. DOI: 10.1145/1978942.1979349.

[Mic] Microsoft. *Scene understanding SDK - Mixed Reality*. Online, accessed: 18.04.2024. URL: https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/scene-understanding-sdk.

[Min95] M. Mine. "ISAAC: A virtual environment tool for the interactive construction of virtual worlds." In: *UNC Chapel Hill Computer Science Technical Report TR95-020*. University of North Carolina at Chapel Hill, 1995.

[MJ02] Michael Lewis and Jeffrey Jacobson. "Game engines in scientific research." In: *Communications of the ACM, 2002*. Vol. 45 No. 1. 2002, pp. 27–31.

[MR17] Marc Beutner and Rasmus Pechuel. "Micro Units A New Approach to Making Learning Truly Mobile." In: *Proceedings of Society for Information Technology & Teacher Education International Conference*. AACE, 2017, pp. 744–751. ISBN: 978-1-939797-27-8.

[MT+21] Ana Martí-Testón, Adolfo Muñoz, J. Ernesto Solanes, Luis Gracia, and Josep Tornero. “A Methodology to Produce Augmented-Reality Guided Tours in Museums for Mixed-Reality Headsets.” In: *Electronics* 10.23 (2021), p. 2956. DOI: `10.3390/electronics10232956`.

[Mue+19a] Thomas Muender, Anke Reinschluessel, Nima Zargham, Tanja Döring, Dirk Wenig, Rainer Malaka, Roland Fischer, Gabriel Zachmann, Christian Schumann, Valentin Kraft, Andrea Schenk, Verena Uslar, Dirk Weyhe, Hans Nopper, and Thomas Lück. *Application Scenarios for 3D-Printed Organ Models for Collaboration in VR & AR*. Mensch und Computer 2019 - Workshopband. 2019. DOI: `10.18420/muc2019-ws-625`.

[Mue+19b] Thomas Muender, Anke V. Reinschluessel, Sean Drewes, Dirk Wenig, Tanja Döring, and Rainer Malaka. “Does It Feel Real? Using Tangibles with Different Fidelities to Build and Explore Scenes in Virtual Reality.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Ed. by Stephen Brewster, Geraldine Fitzpatrick, Anna Cox, and Vassilis Kostakos. Glasgow, Scotland UK: ACM, 2019, pp. 1–12. ISBN: 9781450359702. DOI: `10.1145/3290605.3300903`.

[Mye95] Brad A. Myers. “User interface software tools.” In: *ACM Transactions on Computer-Human Interaction* 2.1 (1995), pp. 64–103. ISSN: 1073-0516. DOI: `10.1145/200968.200971`.

[Nat+20] Nattaon Techasarntikul, Photchara Ratsamee, Jason Orlosky, Tomohiro Mashita, Yuki Uranishi, Kiyoshi Kiyokawa, and Haruo Takemura. “The Effect of the Presence of an Embodied Agent in an AR Guiding System.” In: *Transactions of the Virtual Reality Society of Japan* 25.1 (2020), pp. 68–77. DOI: `10.18974/tvrsj.25.1_68`.

[NCP90] Jay F. Nunamaker, Minder Chen, and Titus D.M. Purdin. “Systems Development in Information Systems Research.” In: *Journal of Management Information Systems* 7.3 (1990), pp. 89–106. ISSN: 0742-1222. DOI: `10.1080/07421222.1990.11517898`.

[Neb+18] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. “ProtoAR: Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications.” In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Ed. by Regan Mandryk, Mark Hancock, Mark Perry, and Anna Cox. Montreal QC, Canada: The Association for Computing Machinery, 2018, pp. 1–12. ISBN: 9781450356206. DOI: `10.1145/3173574.3173927`.

[NM19] Michael Nebeling and Katy Madier. “360proto: Making Interactive Virtual Reality & Augmented Reality Prototypes from Paper.” In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. CHI ’19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, 1–13. ISBN: 9781450359702. DOI: `10.1145/3290605.3300826`.

[NS18] Michael Nebeling and Maximilian Speicher. “The Trouble with Augmented Reality/Virtual Reality Authoring Tools.” In: *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2018. DOI: `10.1109/ismar-adjunct.2018.00098`.

[Obj] Object Management Group. *About the Unified Modeling Language Specification Version 2.5.1*. Online, accessed: 18.04.2024. URL: `https://www.omg.org/spec/UML/1.4/PDF`.

[Pla+21] Bastian Plaß, Jan Emrich, Selina Götz, David Kernstock, and Thomas Klauer. “Evaluation of point cloud data acquisition techniques for Scan-to-BIM workflows in Healthcare.” In: *Smart surveyors for land and water management - challenges in a new reality*. Copenhagen: International Federation of Surveyors FIG, 2021, pp. 1–13. ISBN: 978-87-92853-65-3.

[PLP18] Nadine Pfeiffer-Leßmann and Thies Pfeiffer. “ExProtoVAR: A Lightweight Tool for Experience-Focused Prototyping of Augmented Reality Applications Using Virtual Reality.” In: *HCI International 2018 – Posters’ Extended Abstracts*. Ed. by Constantine Stephanidis. Vol. 851. Communications in Computer and Information Science. Cham: Springer International Publishing, 2018, pp. 311–318. ISBN: 978-3-319-92278-2. DOI: `10.1007/978-3-319-92279-9_42`.

[PM20] Roger S. Pressman and Bruce R. Maxim. *Software engineering: A practitioner’s approach*. Ninth edition. New York, NY: McGraw-Hill Education, 2020. ISBN: 9781259872976.

[Poi+22] Fabian Pointecker, Judith Friedl, Daniel Schwajda, Hans-Christian Jetter, and Christoph Anthes. “Bridging the Gap Across Realities: Visual Transitions Between Virtual and Augmented Reality.” In: *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2022. DOI: `10.1109/ismar55827.2022.00101`.

[Pon+11] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. “Pause-and-play.” In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. Ed. by Jeff Pierce. ACM Conferences. New York, NY: ACM, 2011, pp. 135–144. ISBN: 9781450307161. DOI: `10.1145/2047196.2047213`.

[PRD09] Qi Pan, Gerhard Reitmayr, and Tom W. Drummond. “Interactive model reconstruction with user guidance.” In: *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. 2009, pp. 209–210. DOI: `10.1109/ISMAR.2009.5336460`.

[PRS15] Jenny Preece, Yvonne Rogers, and Helen Sharp. “Interaction design: Beyond human-computer interaction.” In: *Universal Access in the Information Society*. 4th edition. Chichester: Wiley, 2015. ISBN: 9781119020752.

[Ram+13] Hector Ramirez, Eduardo Gonzalez Mendivil, Pablo Ramirez Flores, and Manuel Contero Gonzalez. “Authoring Software for Augmented Reality Applications for the Use of Maintenance and Training Process.” In: *Procedia Computer Science* 25 (2013), pp. 189–193. ISSN: 18770509. DOI: `10.1016/j.procs.2013.11.023`.

[Rau+21] Linda Rau, Robin Horst, Yu Liu, and Ralf Dorner. “A Nugget-Based Concept for Creating Augmented Reality.” In: *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2021, pp. 212–217. ISBN: 978-1-6654-1298-8. DOI: `10.1109/ISMAR-Adjunct54149.2021.00051`.

[Rau+22a] Linda Rau, Jessica L. Bitter, Yu Liu, Ulrike Spierling, and Ralf Dörner. “Supporting the creation of non-linear everyday AR experiences in exhibitions and museums: An authoring process based on self-contained building blocks.” In: *Frontiers in Virtual Reality* 3 (2022). DOI: `10.3389/frvir.2022.955437`.

[Rau+22b] Linda Rau, Dagny C. Döring, Robin Horst, and Ralf Dörner. “Pattern-Based Augmented Reality Authoring Using Different Degrees of Immersion: A Learning Nugget Approach.” In: *Frontiers in Virtual Reality* 3 (2022). DOI: `10.3389/frvir.2022.841066`.

[Rau+23] Linda Rau, Robin Horst, Manuel Feller, and Ralf Dörner. *Bridging Realities: Bidirectional Transitions from and to Augmented and Virtual Reality*. GI VR / AR Workshop. 2023. DOI: `10.18420/vrar2023_3371`.

[Rei+03] T. Reicher, A. Mac Williams, B. Brugge, and G. Klinker. “Results of a study on software architectures for augmented reality systems.” In: *Proceedings of the 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Comput. Soc, 2003, pp. 274–275. ISBN: 0-7695-2006-5. DOI: `10.1109/ISMAR.2003.1240717`.

[RG13] Colin Ray and Stephen B. Gilbert. “Bringing Authoring Tools for Intelligent Tutoring Systems and Serious Games Closer Together: Integrating GIFT with the Unity Game Engine.” In: *Proceedings of the Workshops at the 16th International Conference on Artificial Intelligence in Education*. Vol. 7. 2013, pp. 37–44.

[RM16] Colin Robson and Kieran McCartan. *Real world research: A resource for users of social research methods in applied settings*. Fourth edition. Chichester: Wiley, 2016. ISBN: 111874523X.

[Rob+16] Rafael Alves Roberto, João Paulo Lima, Roberta Cabral Mota, and Veronica Teichrieb. “Authoring Tools for Augmented Reality: An Analysis and Classification of Content Design Tools.” In: *Design, User Experience, and Usability: Technological Contexts*. Ed. by Aaron Marcus. Vol. 9748. Lecture Notes in Computer Science. Cham and s.l.: Springer International Publishing, 2016, pp. 237–248. ISBN: 978-3-319-40405-9. DOI: `10.1007/978-3-319-40406-6_22`.

[RP18] Patrick Renner and Thies Pfeiffer. "Attention Guiding Using Augmented Reality in Complex Environments." In: *25th IEEE Conference on Virtual Reality and 3D User Interfaces*. Ed. by Kiyoshi Kiyokawa. Piscataway, NJ: IEEE, 2018, pp. 771–772. ISBN: 978-1-5386-3365-6. DOI: 10.1109/VR.2018.8446396.

[RS02] Holger Regenbrecht and Thomas Schubert. "Measuring Presence in Augmented Reality Environments: Design and a First Test of a Questionnaire." In: *Proceedings of the Fifth Annual International Workshop Presence*. Porto, Portugal, 2002, pp. 138–144.

[SA14] Marcia Izabel Fugisawa Souza and Sérgio Ferreira do Amaral. "Educational Microcontent for Mobile Learning Virtual Environments." In: *Creative Education* 05.09 (2014), pp. 672–681. ISSN: 2151-4755. DOI: 10.4236/ce.2014.59079.

[Sal99] Sally Fincher. "Analysis of Design: An Exploration of Patterns and Pattern Languages for Pedagogy." In: *Journal of Computers in Mathematics and Science Teaching* 18.3 (1999), pp. 331–348. ISSN: 0731-9258.

[Sch+02] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L. Miguel Encarnação, Michael Gervautz, and Werner Purgathofer. "The Studierstube Augmented Reality Project." In: *Presence: Teleoperators and Virtual Environments* 11.1 (2002), pp. 33–54. ISSN: 1054-7460. DOI: 10.1162/105474602317343640.

[Sew15] Brenden Sewell. *Blueprints Visual Scripting for Unreal Engine*. Birmingham: Packt Publishing, 2015. ISBN: 978-1-78528-601-8.

[SFR01] Thomas Schubert, Frank Friedmann, and Holger Regenbrecht. "The Experience of Presence: Factor Analytic Insights." In: *Presence: Teleoperators and Virtual Environments* 10.3 (June 2001), pp. 266–281. DOI: 10.1162/105474601300343603.

[She+5] J. G. Sheridan, B. W. Short, K. van Laerhoven, N. Villar, and G. Kortuem. "Exploring Cube Affordance: Towards a Classification of Non-Verbal Dynamics of Physical Interfaces for Wearable Computing." In: *IEE Eurowearable '03*. IEE, 4-5 Sept. 2003, pp. 113–118. ISBN: 0 85296 282 7. DOI: 10.1049/ic:20030156.

[SHT17] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. "Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S)." In: *International Journal of Interactive Multimedia and Artificial Intelligence* 4.6 (2017), p. 103. DOI: 10.9781/ijimai.2017.09.001.

[Sin+21] Abbey Singh, Ramanpreet Kaur, Peter Haltner, Matthew Peachey, Mar Gonzalez-Franco, Joseph Malloch, and Derek Reilly. "Story CreatAR: a Toolkit for Spatially-Adaptive Augmented Reality Storytelling." In: *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2021, pp. 713–722. ISBN: 978-1-6654-1838-6. DOI: 10.1109/VR50410.2021.00098.

[Sla02] Mel Slater. "Presence and The Sixth Sense." In: *Presence* 11.4 (2002), pp. 435–439. DOI: 10.1162/105474602760204327.

[SLB08] Hartmut Seichter, Julian Looser, and Mark Billinghurst. "ComposAR: An intuitive tool for authoring AR applications." In: *Proceedings of the 7th IEEE International Symposium on Mixed and Augmented Reality 2008*. Ed. by Mark A. Livingston. Piscataway, NJ: IEEE Service Center, 2008, pp. 177–178. ISBN: 978-1-4244-2840-3. DOI: `10.1109/ISMAR.2008.4637354`.

[SP10] Ben Shneiderman and Catherine Plaisant. *Designing the user interface: Strategies for effective human-computer interaction*. 5th ed. Boston: Addison-Wesley, 2010. ISBN: 0-321-53735-1.

[Spe+15] Maximilian Speicher, Kristina Tenhaft, Simon Heinen, and Harry Handorf. "Enabling industry 4.0 with holobuilder." In: *INFORMATIK 2015*. Bonn: Gesellschaft für Informatik e.V., 2015, pp. 1561–1575. ISBN: 978-3-88579-640-4.

[Spe+18] Maximilian Speicher, Brian D. Hall, Ao Yu, Bowen Zhang, Haihua Zhang, Janet Nebeling, and Michael Nebeling. "XD-AR: Challenges and Opportunities in Cross-Device Augmented Reality Application Development." In: *Proceedings of the ACM on Human-Computer Interaction* Vol. 2 (2018), pp. 1–24. ISSN: 2573-0142. DOI: `10.1145/3229089`.

[SS96] A. Steed and M. Slater. "A dataflow representation for defining behaviours within virtual environments." In: *Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium*. Los Alamitos, California: IEEE Computer Society Press, 1996, pp. 163–167. ISBN: 0-8186-7296-X. DOI: `10.1109/VRAIS.1996.490524`.

[Ssi+19] Seung Youb Ssin, James A. Walsh, Ross T. Smith, Andrew Cunningham, and Bruce H. Thomas. "GeoGate: Correlating Geo-Temporal Datasets Using an Augmented Reality Space-Time Cube and Tangible Interactions." In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR 2019)*. Piscataway, NJ: IEEE, 2019, pp. 210–219. ISBN: 978-1-7281-1377-7. DOI: `10.1109/VR.2019.8797812`.

[SSL21] Maria Shehade and Theopisti Stylianou-Lambert. "Emerging Technologies and the Digital Transformation of Museums and Heritage Sites." In: *Proceedings of RISE IMET 2021*. Vol. 1432. Communications in Computer and Information Science. Nicosia, Cyprus: Springer Cham, 2021. ISBN: 978-3-030-83646-7. DOI: `10.1007/978-3-030-83647-4`.

[Ste+09] Frank Steinicke, Gerd Bruder, Klaus Hinrichs, Markus Lappe, Brian Ries, and Victoria Interrante. "Transitional environments enhance distance perception in immersive virtual reality systems." In: *Proceedings of the 6th Symposium on Applied Perception in Graphics and Visualization - APGV '09*. New York, New York, USA: ACM Press, 2009. DOI: `10.1145/1620993.1620998`.

[Ste+10] Frank Steinicke, Gerd Bruder, Klaus Hinrichs, and Anthony Steed. "Gradual transitions and their effects on presence and distance estimation." In: *Computers & Graphics* 34.1 (2010), pp. 26–33. ISSN: 0097-8493. DOI: `https://doi.org/10.1016/j.cag.2009.12.003`.

[SUS94] Mel Slater, Martin Usoh, and Anthony Steed. "Depth of Presence in Virtual Environments." In: *Presence: Teleoperators and Virtual Environments* 3.2 (1994), pp. 130–144. ISSN: 1054-7460. DOI: 10.1162/pres.1994.3.2.130.

[SVG15] Adalberto L. Simeone, Eduardo Velloso, and Hans Gellersen. "Substitutional Reality." In: *CHI 2015 crossings*. Ed. by Jinwoo Kim. New York, NY: ACM, 2015, pp. 3307–3316. ISBN: 9781450331456. DOI: 10.1145/2702123.2702389.

[SVN19] Priyank Sharma, Ruth Ann Vleugels, and Vinod E. Nambudiri. "Augmented reality in dermatology: Are we ready for AR?" In: *Journal of the American Academy of Dermatology* 81.5 (2019), pp. 1216–1222. DOI: 10.1016/j.jaad.2019.07.008.

[TLR22] Lingwei Tong, Robert W. Lindeman, and Holger Regenbrecht. "Adaptive Playback Control: A Framework for Cinematic VR Creators to Embrace Viewer Interaction." In: *Frontiers in Virtual Reality* 2 (2022). DOI: 10.3389/frvir.2021.798306.

[VF17] Dimitar Valkov and Steffen Flagge. "Smooth immersion: The benefits of Making the Transition to Virtual Environments a Continuous Process." In: *Proceedings of the 5th Symposium on Spatial User Interaction*. Ed. by Adalberto L. Simeone, Kyle Johnsen, Rob Teather, and Christian Sandor. New York, NY, USA: ACM, 2017, pp. 12–19. ISBN: 9781450354868. DOI: 10.1145/3131277.3132183.

[VZL17] Eswara Rao Velamkayala, Manuel V. Zambrano, and Huiyang Li. "Effects of HoloLens in Collaboration: A Case in Navigation Tasks." In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 61.1 (2017), pp. 2110–2114. ISSN: 2169-5067. DOI: 10.1177/1541931213602009.

[WCL19] Yi-Chin Wu, Liwei Chan, and Wen-Chieh Lin. "Tangible and Visible 3D Object Reconstruction in Augmented Reality." In: *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2019)*. Piscataway, NJ: IEEE, 2019, pp. 26–36. ISBN: 978-1-7281-0987-9. DOI: 10.1109/ISMAR.2019.00-30.

[WLF07] Sean White, Levi Lister, and Steven Feiner. "Visual Hints for Tangible Gestures in Augmented Reality." In: *Proceedings / the Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality*. Los Alamitos, California: IEEE Computer Society, 2007, pp. 1–4. ISBN: 978-1-4244-1749-0. DOI: 10.1109/ISMAR.2007.4538824.

[WS01] Gerold Wesche and Hans-Peter Seidel. "FreeDrawer." In: *Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA: ACM, 2001. DOI: 10.1145/505008.505041.

[WS98] Bob G. Witmer and Michael J. Singer. "Measuring Presence in Virtual Environments: A Presence Questionnaire." In: *Presence: Teleoperators and Virtual Environments* 7.3 (1998), pp. 225–240. ISSN: 1054-7460. DOI: 10.1162/105474698565686.

[WTS10] Ming-Jen Wang, Chien-Hao Tseng, and Cherng-Yeu Shen. “An Easy to Use Augmented Reality Authoring Tool for Use in Examination Purpose.” In: *Human-Computer Interaction*. Ed. by Peter Forbrig. Vol. 332. IFIP Advances in Information and Communication Technology. Berlin, Heidelberg: IFIP International Federation for Information Processing, 2010, pp. 285–288. ISBN: 978-3-642-15230-6. DOI: 10.1007/978-3-642-15231-3_31.

[Zhe+21] Mengya Zheng, Xingyu Pan, Xuanhui Xu, and Abraham G. Campbell. “METAL: Explorations into Sharing 3D Educational Content across Augmented Reality Headsets and Light Field Displays.” In: *7th International Conference of the Immersive Learning Research Network (iLRN)*. 2021, pp. 1–6. DOI: 10.23919/iLRN52045.2021.9459419.

Appendix B

List of Own Publications

Peer-Reviewed Publications as First Author

Rau, Linda; Horst, Robin; Liu, Yu; Dörner, Ralf; Spierling, Ulrike: A Tangible Object for General Purposes in Mobile Augmented Reality Applications. In: Reussner, R. H., Koziolek, A. & Heinrich, R. (Hg.): INFORMATIK 2020. Bonn, S. 947–954.

Rau, Linda; Horst, Robin; Liu, Yu; Dorner, Ralf (2021): A Nugget-Based Concept for Creating Augmented Reality. In: 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). Bari, Italy, 10/4/2021 - 10/8/2021: IEEE, S. 212–217.

Rau, Linda; Döring, Dagny C.; Horst, Robin; Dörner, Ralf (2022): Pattern-Based Augmented Reality Authoring Using Different Degrees of Immersion: A Learning Nugget Approach. In: Front. virtual real. 3, Artikel 841066. DOI: 10.3389/frvir.2022.841066.

Bitter, Jessica L.; Dörner, Ralf; Liu, Yu; Rau, Linda; Spierling, Ulrike (2022): Follow the Blue Butterfly – An Immersive Augmented Reality Museum Guide. In: Constantine Stephanidis, Margherita Antona und Stavroula Ntoa (Hg.): HCI International 2022 Posters. 24th International Conference on Human-Computer Interaction, HCII 2022, Virtual Event, June 26 – July 1, 2022, Proceedings, Part III, Bd. 1582. 1st ed. 2022. Cham: Springer International Publishing; Imprint Springer (Springer eBook Collection, 1582), S. 171–178.

Rau, Linda; Bitter, Jessica L.; Liu, Yu; Spierling, Ulrike; Dörner, Ralf (2022): Supporting the creation of non-linear everyday AR experiences in exhibitions and museums: An authoring process based on self-contained building blocks. In: Front. virtual real. 3, Artikel 955437. DOI: 10.3389/frvir.2022.955437.

Rau, Linda; Horst, Robin; Feller, Manuel; Dörner, Ralf (2023): Bridging Realities: Bidirectional Transitions from and to Augmented and Virtual Reality. GI VR / AR Workshop. DOI: 10.18420/vrar2023_3371. Gesellschaft für Informatik e.V.. Köln. 19. - 20. September 2023

Rau, Linda; Döring, Dagny C.; Horst, Robin; Dörner, Ralf (2023): A Perspective on Interface Techniques in Tangible Augmented Reality for Mobile Devices. Journal of Virtual Reality and Broadcasting, 17 (2023), no. 1. DOI:10.48663/1860-2037/17.2023.1

Peer-Reviewed Publications as Co-Author

Döring, Dagny C.; Horst, Robin; Rau, Linda; Dörner, Ralf (2020): Interface Techniques for Tangible Augmented Reality in a Mobile Device Setup for Magic Lens Experiences.

Horst, Robin; Klonowski, Fabio; Rau, Linda; Dörner, Ralf (2020): The Shared View Paradigm in Asymmetric Virtual Reality Setups. In: i-com 19 (2), S. 87–101. DOI: 10.1515/icom-2020-0006.

Horst, Robin; Naraghi-Taghi-Off, Ramtin; Rau, Linda; Dörner, Ralf (2020): Bite-Sized Virtual Reality Learning Applications: A Pattern-Based Immersive Authoring Environment. In: J. Univers. Comput. Sci. 26 (8), S. 947–971. DOI: 10.3897/jucs.2020.051.

Horst, Robin; Fenchel, Dennis; Retz, Reimond; Rau, Linda; Retz, Wilhelm; Dörner, Ralf: Integration of Game Engine Based Mobile Augmented Reality Into a Learning Management System for Online Continuing Medical Education. In: Reussner, R. H., Koziolek, A. & Heinrich, R. (Hg.): INFORMATIK 2020. Bonn.

Horst, Robin; Rau, Linda; Dieter, Lars; Feller, Manuel; Gaida, Jonas; Leipe, Andreas et al.: Presenters in Virtual Reality in Slideshow Presentations. In: Reussner, R. H., Koziolek, A. & Heinrich, R. (Hg.): INFORMATIK 2020. Bonn, S. 963–970.

Bastian Plaß, Kira Zschiesche, Tamer Altinbas, Daniel Karla, Linda Rau, Martin Schlüter (2020): Künstliche Intelligenz als Strategie in der Ingenieurgeodäsie – erste Schritte im Bahnumfeld. In: zfv – Zeitschrift für Geodäsie, Geoinformation und Landmanagement (4/2020), S. 236–240. DOI: 10.12902/zfv-0307-2020.

Zschiesche, Kira; Schlüter, Martin; Rau, Linda (2020): BIM in der Praxis - Ansätze zur Integration von Structural Health Monitoring in ein Bestands-BIM. In: D. e. V. Gesellschaft für Geodäsie V.W. Geoinformation und Landmanagement und Runder Tisch GIS e. V. (Hg.): Leitfaden Geodäsie und BIM. Version 2.1 (2020). 1. Auflage. Augsburg: Wißner-Verlag, S. 166–167.

Zschiesche, Kira; Rau, Linda; Schlüter, Martin: Optische Schwingungsmessungen: Status, Integration, Pros und Contras. In: GeoMonitoring 2020.

Döring, Dagny; Horst, Robin; Rau, Linda; Dörner, Ralf (2021): Sensory Extension of a Tangible Object for Physical User Interactions in Augmented Reality. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - HUCAPP; ISBN 978-989-758-488-6; ISSN 2184-4321, SciTePress, pages 153-160. DOI: 10.5220/0010230301530160

Liu, Yu; Spierling, Ulrike; Rau, Linda; Dörner, Ralf (2021): Handheld vs. Head-Mounted AR Interaction Patterns for Museums or Guided Tours. In: Navid Shaghaghi, Fabrizio Lamberti, Brian Beams, Reza Shariatmadari und Ahmed Amer (Hg.): Intelligent Technologies for Interactive Entertainment, Bd. 377. Cham: Springer International Publishing (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 377), S. 229–242.

Horst, Robin; Naraghi-Taghi-Off, Ramtin; Rau, Linda; Dörner, Ralf (2021): Back to reality: transition techniques from short HMD-based virtual experiences to the physical world. In: Multimed Tools Appl. DOI: 10.1007/s11042-021-11317-w.

Robin Horst; Ramtin Naraghi-Taghi-Off; Linda Rau; Ralf Dörner (2021): Remote Emergency Teaching and Virtual Reality Education: A Case-Study Using VR Nuggets in non-VR Courses. In: Andreas Lingnau (Hg.): Proceedings of DELFI Workshops 2021: Hochschule Ruhr West (DELFI 2021 - 19. Fachtagung Bildungstechnologien der GI), S. 35–46.

Horst, Robin; Naraghi-Taghi-Off, Ramtin; Rau, Linda; Dörner, Ralf: Design Patterns and Author Roles for Pattern-Based Educational Virtual Reality Experiences. In: Proceedings of the 13th Workshop Virtual and Augmented Reality of the GI Group VR/AR (Shaker Verlag).

Horst, Robin; Gerstmeier, Simon; Naraghi-Taghi-Off, Ramtin; Wagner, Julian; Rau, Linda; Dörner, Ralf (2022): Virtual reality content creation based on self-contained components in the e-learning domain: Re-using pattern-based vr content in different authoring toolkits. In: Multimed Tools Appl. DOI: 10.1007/s11042-022-13362-5.

Horst, Robin; Naraghi-Taghi-Off, Ramtin; Rau, Linda; Doerner, Ralf (2022): Authoring With Virtual Reality Nuggets - Lessons Learned. In: Front. virtual real. 3, Artikel 840729. DOI: 10.3389/frvir.2022.840729.

Appendix C

Awards

Best Poster Award for:

Döring, D.; Horst, R.; Rau, L. and Dörner, R. (2021). Sensory Extension of a Tangible Object for Physical User Interactions in Augmented Reality. In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - HUCAPP; ISBN 978-989-758-488-6; ISSN 2184-4321, SciTePress, pages 153-160. DOI: 10.5220/0010230301530160

Appendix D

Authoring Workshop Manual

The manual for the authoring workshop was provided in english and german. The section "Task" was extended for group B. The other sections were identical for both groups.

Authoring Workshop Guide

17.03.2023 / 23.03.2023

Information

- You can find our AR nuggets, authoring tools, sample 3D models, etc. in the Unity project in the **Assets/AuthoringWorkshop** folder
- More 3D models can be found on the Internet. Most of the time, these can be easily imported into Unity, the best way to do this is .fbx format for it. Websites where you can download additional 3D models are, for example:
    - https://sketchfab.com/nebulousflynn/collections/cc0
    - https://www.turbosquid.com/de/Search/3D-Models/free
    - But also in our Unity project there are various examples of 3D models
- We are happy to help you and answer your questions. Talk to us:
    - Linda Rau
    - Jessica Bitter
    - Manuel Feller
    - Yu Liu (Joe)

Task [Group A]

1. Use a Transparency AR nugget to show how the skin of a whale or dinosaur lies over the bone. In our room here there are various physical mini-skeletons distributed, which you can enrich virtually with 3D models.
2. Use a Show & Tell AR nugget to explain the skeleton of an animal. You can orient yourself on the video shown at the beginning, in which the bones of a fin whale were labeled. In our room here there are various physical mini-skeletons distributed that you can label with virtual labels.
3. Try to realize your own ideas by customizing more AR nuggets. The standard AR nuggets with cubes, balls or other simple objects can serve as inspiration. In our research project, we applied the AR nuggets and tools in the context of a museum. They are allowed to stay in the museum area with their ideas, but may also adapt the AR nuggets and tools for ideas from other areas.

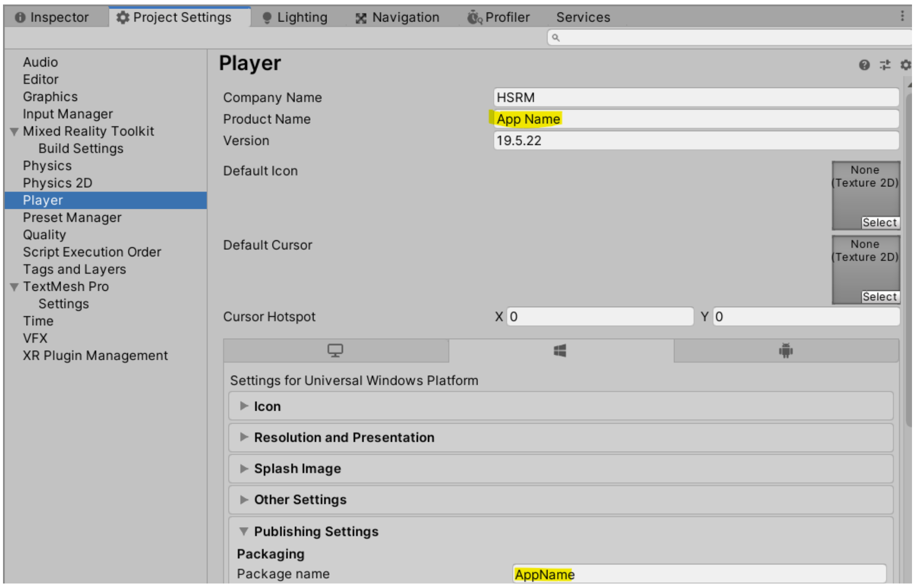
Task [Group B]

1. Use a Show & Tell AR Nugget to explain the skeleton of an animal. You can use the video shown at the beginning, in which the bones of a fin whale were labeled, as a guide.
    - Drag the Show & Tell AR Nugget from the Project window into the Hierachy.
    - Replace the cube with another 3D model of your choice.
    - Adjust the position and text of the label.
    - Duplicate (right-click Duplicate) the label or drag the prefab from the AuthoringWorkshop/Prefabs folder to the Hierachy to add more labels.
    - The Show & Tell AR Nugget already includes the "Grabbable" tool several times so you can move the labels individually on the HoloLens later. Consider which GameObjects you can move (because the Grabbable tool is attached in the Hierachy). Does this meet your own expectations?
    - Should the labels align to the user so that the text is readable from any position? If so, use the "RotateParentToGameobject" tool.

2. Create a second AR nugget, independent of the first. Use a Transparency AR Nugget to show how the skin of a whale, dinosaur, or other museum exhibit lies over the bones.
3. Decide when you want your AR Nuggets to start and stop. To do this, click the GameObject of the AR Nugget itself, e.g. "ShowAndTell" and define start and end conditions. Then select the NuggetManager and check the "FunctionsOn" box.
4. Test your AR application in Unity Play Mode. The NuggetManager takes about 3 seconds to control whether an AR nugget is turned on or off. Test both Visitor and Author modes by switching with the ModeSwitcher. In Author mode, you have the choice of seeing all AR nuggets at once or letting the Nugget Manager control it, as in Visitor mode (check Functions On in the Nugget Manager).
5. Optional: Try to implement your own ideas by customizing more AR Nuggets. The default AR Nuggets with cubes, spheres or other simple objects, can serve you as inspiration. In our research project, we applied the AR Nuggets and tools in the context of a museum. You may keep your ideas in the museum domain, but you may also adapt the AR Nuggets and Tools for ideas from other domains.

Build and authoring process on the HoloLens

- Give your app a name. Please be sure to give the visitor mode app a different name than the author mode app, such as "meineApp_Visitor" and "meineApp_Author". You can find the Project Settings either in the menu bar at the top under "Edit → Project Settings" or the window is already open next to the Inspector window in another tab.



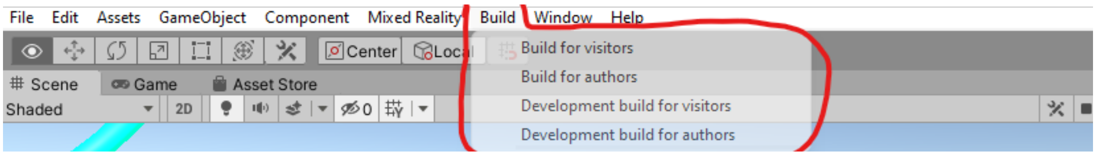
Player

Company Name: HSRM
Product Name: App Name
Version: 19.5.22
Default Icon: None (Texture 2D)
Default Cursor: None (Texture 2D)
Cursor Hotspot: X 0 Y 0

Settings for Universal Windows Platform

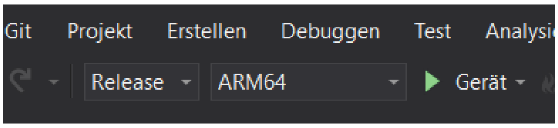
Publishing Settings
Packaging
Package name: AppName

- Build your app with Mode Switcher in visitor mode and author mode



- In the project folder, find your self-created app under \Builds\
Open the . sln file with Visual Studio, connect the HoloLens to your laptop with a cable, set

the following in Visual Studio and click "Debuggen" → “Starten ohne Debuggen”

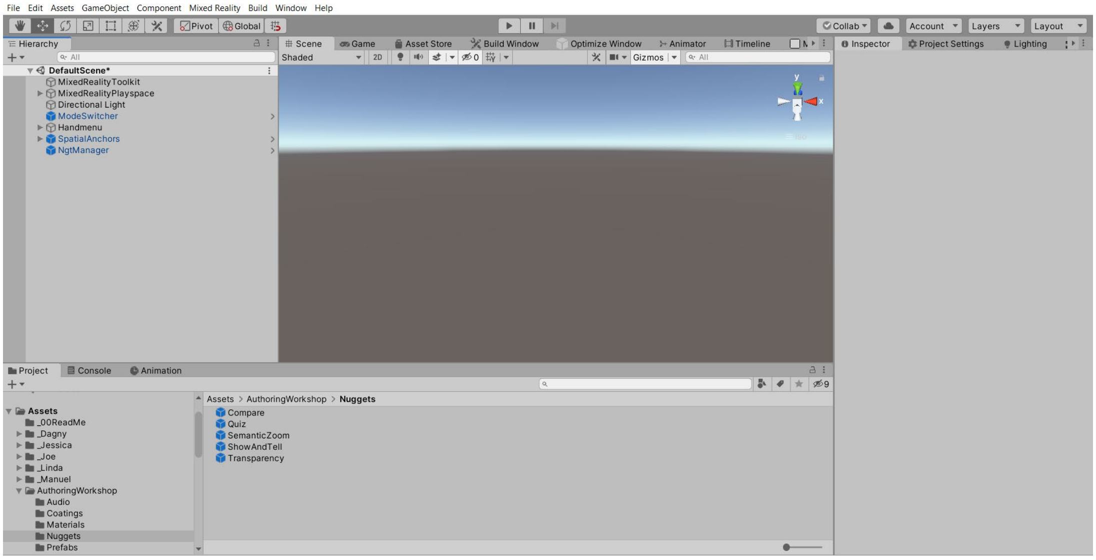


- Launch the authoring app on the HoloLens
    - Move to the place where you want to place an AR nugget
    - From the menu, you can select an AR nugget, which will then be placed in front of you
    - Alternatively and additionally, you can touch the virtual content or use the handbeam to move, rotate or scale it.
    - Finally, you should export the spatial anchors, i.e. your placements of the AR nuggets. To do this, use the menu again
    - you can also import spatial anchor in the same way
- Launch the visitor app on the HoloLens
    - The spatial anchors are imported automatically
    - Make sure that your content appears in the right place and that the application works as you imagined.

Descriptions and instructions for details (use if necessary)

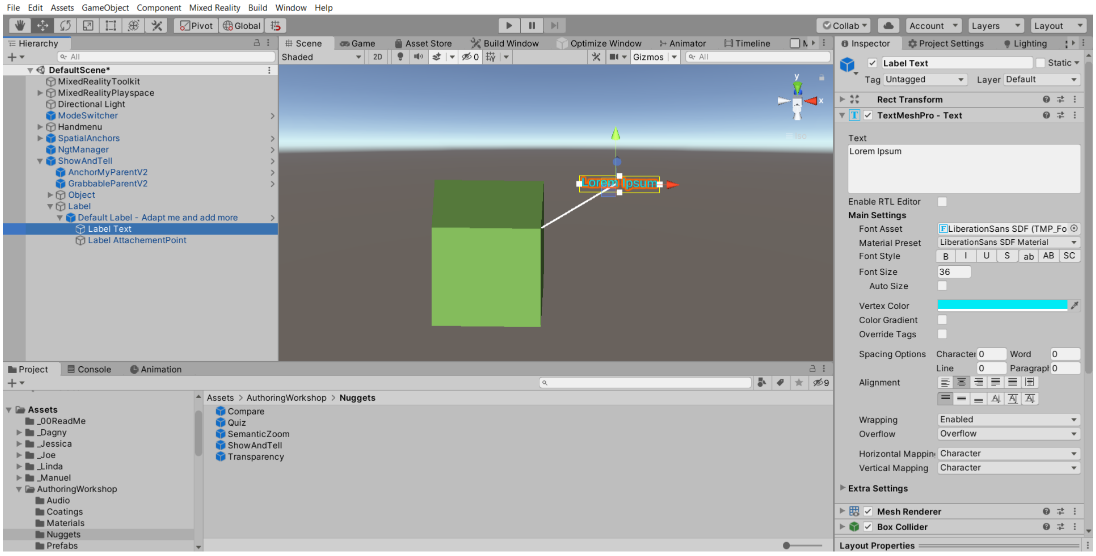
About Unity

- Drag & Drop from Project to Hierarchy



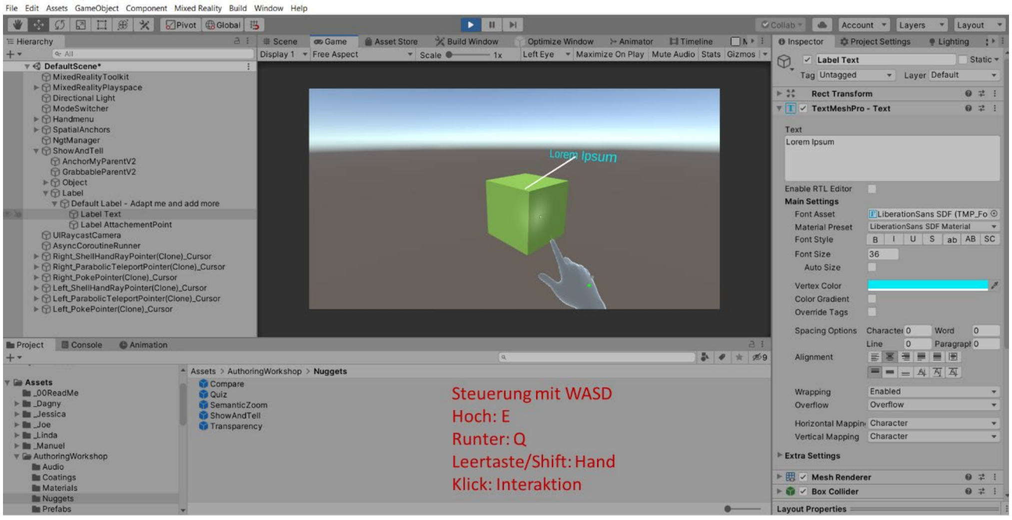
- Edit the position by dragging with the arrows

Edit properties in the Inspector window



Label Text
Tag Untagged
Layer Default
Text: Lorem Ipsum
Enable RTL Editor
Main Settings
Font Asset: LiberationSans SDF (TMP_Fo
Material Preset: LiberationSans SDF Material
Font Style: B I U S ab AB SC
Font Size: 36
Auto Size
Vertex Color
Color Gradient
Override Tags
Spacing Options: Character 0, Word 0, Line 0, Paragraph 0
Alignment
Wrapping: Enabled
Overflow: Overflow
Horizontal Mapping: Character
Vertical Mapping: Character
Extra Settings

- Play mode for testing with play button
  Attention: Changes made during Play mode will not be saved



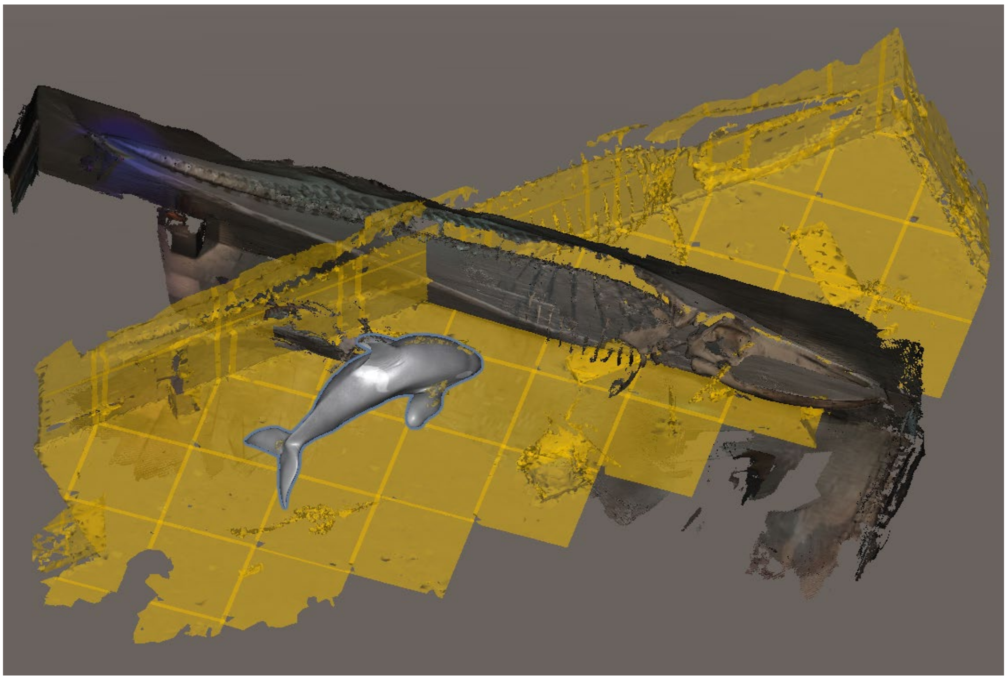
Steuerung mit WASD
Hoch: E
Runter: Q
Leertaste/Shift: Hand
Klick: Interaktion

Positioning of any content in real space

In Unity, virtual objects (and other objects as well) are called GameObjects. For each of these GameObjects, you define a position in a parent coordinate system in Unity, e.g. with the coordinates X = -1; Y = 0; Z = 3. This is also written as (-1, 0, 3).

X-axis (left/right): negative value to the left, positive value to the right
Y-axis (bottom/up): negative value downwards, positive value upwards
Z-axis (behind/forward): negative value to the back, positive value to the front

When using the HoloLens, the origin of the coordinate system is where the HoloLens is when the app starts. The GameObject from our example with position (-1, 0, 3) will therefore appear 1 meter to the left of and 3 meters before the HoloLens (the origin of the coordinate system) when the app starts. As in the picture below, it can happen that the desired (whale and yellow scan of the room) does not correspond to reality (shown here as a colored scan of the room). This means that in order to be able to place virtual objects correctly in real space, you would have to place the HoloLens in exactly the right place and start the app from there. This is difficult to implement in practice.



Therefore, we place the virtual objects once ig manually and store their position with spatial anchors. Manual placement is achieved with our "Grabbable" tool or via our placement menu.

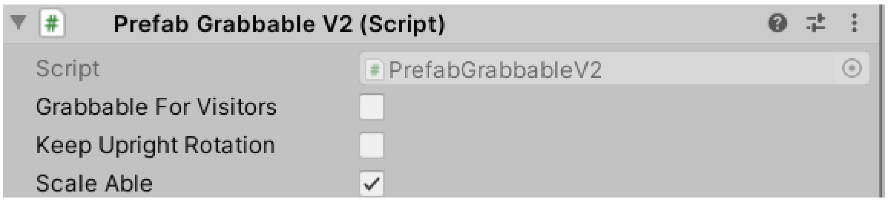
Authoring Tools in Unity

AnchorMyParent

- Adds a spatial anchor to the parent GameObject. This allows the GameObject to be anchored to a real location and to also appear there again when the HoloLens is switched off and on again.
- For example: With drag & drop, AnchorMyParent is dragged into the Hierachy window onto the 3D model of a whale. When the application runs on the HoloLens, the position of the virtual whale is stored and the virtual whale is placed there each time the application starts.

GrabbableParent

- Makes the parent object tangible so that it can be moved, rotated, or scaled with your hands.



Prefab Grabbable V2 (Script)

Script: PrefabGrabbableV2
Grabbable For Visitors ☐
Keep Upright Rotation ☐
Scale Able ☑

- For example, GrabbableParent can be dragged onto a 3D model of a whale. Then the virtual whale can be placed manually with the HoloLens in exactly the right place in the museum.
- Placing with the HoloLens can be challenging, sometimes rotating or scaling the object unintentionally. To avoid this, there are the options
  - "Keep Upright Position", which allows the GameObject to rotate only around the vertical axis
  - "scale-able", if this option is deactivated, the GameObject can no longer be scaled.

- Grabbable For Visitors: If this function is activated, visitors can also touch and move the GameObject in visitor mode. The visitor mode is used with the ModeSwitcher (s. ModeSwitcher and SelectParentMode).

ModeSwitcher and SelectParentMode

- The start scene already contains the GameObject "ModeSwitcher". Here you can switch between Visitor and Author mode with one click.

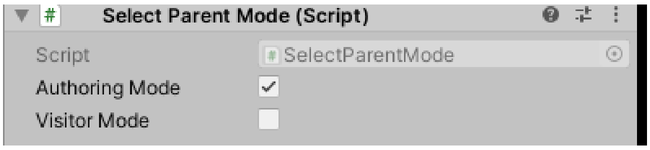
Mode Switcher (Script)

Current Mode: Author

Script: ModeSwitcher

Switch between modes

- GameObjects that use the Grabbable Tool are automatically deactivated (hidden) in Visitor mode, unless the Grabbable for Visitors option has been activated in the Grabbable Tool.
- In addition, GameObjects can be configured so that they are only active in authoring mode or exclusively in visitor mode. To do this, attach the SelectParentMode tool to the GameObject and select the desired mode.



Select Parent Mode (Script)

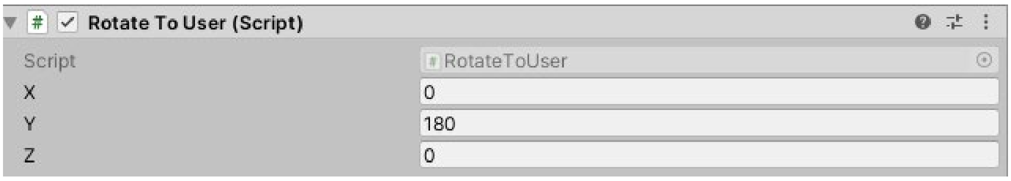
Script: SelectParentMode

Authoring Mode: ☑

Visitor Mode: ☐

RotateParentToUser

- Rotates the parent GameObject to the user.



Rotate To User (Script)

Script: RotateToUser

X: 0

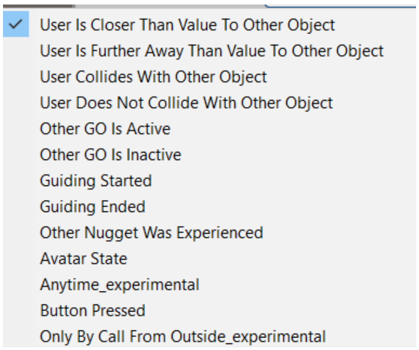
Y: 180

Z: 0

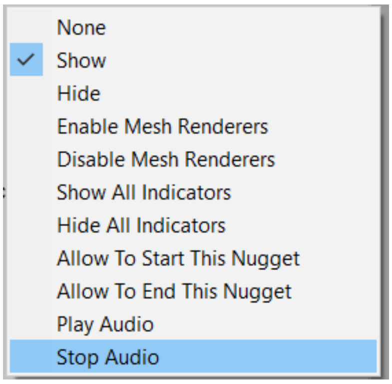
- If the parent GameObject is displayed exactly the wrong way around, the desired rotation can be set with the X, Y, Z values.

ARNugget and ARNuggetManager

- AR nuggets canbe started or stopped under certain conditions. These can be set via a drop-down menu and are automatically monitored by the ARNuggetManager.
- Conditions



    -
    - User is close than value to other object = If the user is closer than X meters to an object. If this option is selected, the fields "Distance" and "Other GameObject" that appear must show how close the user must be to the GameObject for the condition to be fulfilled and to which GameObject this refers.

    - User is closer than value to other object = If the user is further than X meters away from an object. If this option is selected, the fields "Distance" and "Other GameObject" that appear must show how close the user must be to the GameObject for the condition to be fulfilled and to which GameObject this refers.
- Reactions
    - After selecting a reaction to occur when the condition(s) are met, a window appears in which the associated GameObject can be defined, e.g., Select "Show" and a GameObject to activate.
    - Hide and Show enabled bzw. deactivates GameObjects, i.e. their functionalities are also activated or deactivated. If the GameObjects are only to be shown or hidden, Enable/Disable Mesh Renderers should be used.
    - The Show and Hide All Indicators options only work if the prefab "ParentIsIndicator" has been used to define what indicators are.
    - 

Condition Manager

- This tool is different from the other tools and is not needed to customize the available AR nuggets. However, it can be helpful if you have a creative idea and want to implement it that does not fit any of the available AR nuggets.
- With the pattern-based AR Nuggets Compare, Semantic Zoom, Show & Tell, Transparency and Quiz, the logic, e.g. that the slider can control the transparency of a virtual object in the Transparency AR Nugget, is already implemented.
- If your creative idea follows a different logic and therefore cannot be mapped with the currently available AR nuggets, you can implement your own logic with the ConditionManager. Such logic follows the pattern "If condition ABC is met, XYZ should be executed." For example
    - **If** the user is closer than 5 meters to the AR Nugget, **then** play a sound.
    - **If** this button is pressed, **then** display the virtual object.
- Several conditions can also be combined, e.g. "If the user stands within 3 meters of the dinosaur and presses a button, a sound should be played, and a virtual object should disappear."
- The conditions are explained in the ARNugget and ARNuggetManager sections.

Appendix E

Research Data

Material from expert user study (Section 6.1)

Participant#	**Q1.AR**	**Q1.VR**	**Q2.AR**	**Q2.VR**	**Q3.AR**	**Q3.VR**	**Q4.AR**	**Q4.VR**	**Q5**	**Q6**	**Q7**	**Q10**	**Q12.AR**	**Q12.VR**	**gender**	**age**
1	7	6	7	4	6	3	5	4	6	7	ja	ja	2	2	Männlich	61
2	7	7	7	7	7	7	6	6	6	6	ja	ja	2	2	Männlich	60
3	5	5	1	1	3	3	3	3	6	7	ja	ja	1	2	Weiblich	53
4	5	6	5	5	6	3	4	3	6	7	ja	ja	2	1	Männlich	52
5	6	4	5	4	5	2	4	4	5	7	ja	ja	2	1	Weiblich	60
6	7	4	7	5	7	6	7	5	7	7	ja	ja	2	1	Männlich	52
Mean	6,17	5,33	5,33	4,33	5,67	4,00	4,83	4,17	6,00	6,83			1,83	1,50		56,33
SD	0,90	1,11	2,13	1,80	1,37	1,83	1,34	1,07	0,58	0,37			0,37	0,50		4,03

Material From User Study About Tangible Interactions (Section 6.2)

Show & Tell AR Nugget		Generic Tangibles			Realistically Shaped Tangibles			-
Participant#	Group	Q1	Q2	Q4	Q1	Q2	Q4	Q8
1	A	7	7	3	7	7	2	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
2	A	6	4	3	6	6	5	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
3	A	5	6	4	6	6	3	universelles Tangible (Würfel auf Stativ)
4	A	5	5	3	5	4	4	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
5	A	7	6	2	7	7	7	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
6	A	4	6	3	3	5	2	universelles Tangible (Würfel auf Stativ)
7	B	7	2	5	7	3	6	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
8	B	7	7	7	6	7	7	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
9	B	6	6	6	5	6	6	universelles Tangible (Würfel auf Stativ)
10	B	5	4	7	7	7	5	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
11	B	4	2	6	7	6	5	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
Mean		5,73	5,00	4,45	6,00	5,82	4,73	
SD		1,14	1,71	1,72	1,21	1,27	1,71	

Progression AR Nugget		Generic Tangibles			Realistically Shaped Tangibles			-
Participant#	Group	Q1	Q2	Q4	Q1	Q2	Q4	Q8
1	A	6	6	2	7	7	2	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
2	A	6	5	6	6	4	2	universelles Tangible (Würfel auf Stativ)
3	A	7	7	2	7	7	6	universelles Tangible (Würfel auf Stativ)
4	A	7	2	5	7	3	5	universelles Tangible (Würfel auf Stativ)
5	A	6	7	5	7	7	7	universelles Tangible (Würfel auf Stativ)
6	A	2	6	6	6	6	6	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
7	B	5	2	5	5	3	5	universelles Tangible (Würfel auf Stativ)
8	B	7	7	7	7	7	7	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
9	B	6	6	6	6	6	6	universelles Tangible (Würfel auf Stativ)
10	B	6	6	7	7	2	7	anwendungsspezifisches Tangible (Haut- oder Wirbel-Model)
11	B	4	2	7	4	2	5	universelles Tangible (Würfel auf Stativ)
Mean		5,64	5,09	5,27	6,27	4,91	5,27	
SD		1,43	1,98	1,71	0,96	2,02	1,71	

General		Generic Tangibles				Realistically Shaped Tangibles				Combined Tangible							
Participant#	Group	Q9	Q10	Q11	Q13	Q9	Q10	Q11	Q13	Q9	Q10	Q11	Q13	Experience with AR	Experience with smartphones	age	gender
1	A	4	4	4	1	4	4	2	1	4	4	2	2	4	7	23	Männlich
2	A	4	4	4	4	4	4	4	4	4	4	4	4	2	4	58	Weiblich
3	A	4	3	5	5	4	4	3	2	4	4	5	5	1	6	26	Weiblich
4	A	4	4	1	1	5	6	1	1	4	4	1	1	5	7	27	Weiblich
5	A	4	4	4	1	4	4	1	1	4	4	1	1	3	6	26	Weiblich
6	A	4	1	4	4	4	4	3	5	4	3	3	3	4	6	29	Männlich
7	B	4	4	1	2	4	4	1	2	5	4	1	3	3	7	22	Weiblich
8	B	4	4	1	1	4	4	1	1	4	4	1	1	2	3	63	Männlich
9	B	4	4	2	2	3	4	3	2	4	4	2	2	3	5	30	Männlich
10	B	4	4	2	1	4	4	1	1	4	4	1	1	2	5	27	Weiblich
11	B	3	4	1	1	3	4	1	1	4	4	1	1	5	6	26	Männlich
Mean		3,91	3,64	2,64	2,09	3,91	4,18	1,91	1,91	4,09	3,91	2,00	2,18	3,09	5,64	32,45	
SD		0,29	0,88	1,49	1,44	0,51	0,57	1,08	1,31	0,29	0,29	1,35	1,34	1,24	1,23	13,44	

UEQ-S	Participant#	Group	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Pragmatic Quality	Hedonic Quality	Overall
Generic Tangible	1	A	3	2	3	2	3	3	2	2	2,50	2,50	2,50
	2	A	1	0	2	1	2	2	1	2	1,00	1,75	1,38
	3	A	2	2	3	2	2	2	3	3	2,25	2,50	2,38
	4	A	-2	3	-3	2	-1	-2	3	3	0,00	0,75	0,38
	5	A	3	2	1	2	2	2	3	3	2,00	2,50	2,25
	6	A	0	-1	-1	-1	0	0	0	2	-0,75	0,50	-0,13
	7	B	0	1	1	3	-1	-2	0	0	1,25	-0,75	0,25
	8	B	1	1	0	0	-1	1	3	3	0,50	1,50	1,00
	9	B	2	2	2	1	2	2	1	1	1,75	1,50	1,63
	10	B	2	2	2	0	-1	-1	0	0	1,50	-0,50	0,50
	11	B	1	3	2	-1	-3	-3	2	2	1,25	-0,50	0,38
Realistically Shaped Tangible	1	A	3	2	3	2	3	3	2	2	2,50	2,50	2,50
	2	A	2	-1	0	-1	2	0	1	1	0,00	1,00	0,50
	3	A	3	3	1	3	3	2	1	1	2,50	1,75	2,13
	4	A	3	3	3	3	3	3	3	3	3,00	3,00	3,00
	5	A	3	3	3	3	3	3	2	2	3,00	2,50	2,75
	6	A	2	0	1	1	1	1	-1	1	1,00	0,50	0,75
	7	B	1	2	0	1	1	-1	1	0	1,00	0,25	0,63
	8	B	3	3	3	3	2	3	3	3	3,00	2,75	2,88
	9	B	1	1	2	1	0	2	2	1	1,25	1,25	1,25
	10	B	3	3	3	3	2	2	1	1	3,00	1,50	2,25
	11	B	2	1	2	3	2	3	-2	-2	2,00	0,25	1,13
Combined Tangible	1	A	3	2	3	3	3	3	2	2	2,75	2,50	2,63
	2	A	0	0	2	2	2	2	1	1	1,00	1,50	1,25
	3	A	2	-1	3	1	3	3	3	3	1,25	3,00	2,13
	4	A	2	3	3	3	3	3	3	3	2,75	3,00	2,88
	5	A	2	2	2	1	2	3	3	3	1,75	2,75	2,25
	6	A	1	2	2	2	1	1	1	1	1,75	1,00	1,38
	7	B	1	1	-1	2	1	1	0	0	0,75	0,50	0,63
	8	B	3	-3	3	3	3	3	3	3	1,50	3,00	2,25
	9	B	2	2	2	1	2	2	1	1	1,75	1,50	1,63
	10	B	2	2	2	2	2	2	2	1	2,00	1,75	1,88
	11	B	2	3	2	1	2	2	-2	-2	2,00	0,00	1,00

Material From User Study About User Assistance (Section 6.3)

	overall				Distance					Movement					Touch					Video	
Participant#	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q5	Q6	Q7	Q8	Q9	Q5	Q6	Q7	Q8	Q9	Q5	Q7
1	2	1	1	2	1	1	2	3	3	2	2	4	4	3	4	3	3	3	3	1	1
2	1	1	1	2	1	1	1	1	4	1	1	1	5	4	1	1	1	4	4	1	1
3	1	1	1	2	1	1	1	5	4	1	2	1	5	3	1	1	1	5	4	1	1
4	1	1	1	1	1	1	1	4	4	3	2	1	5	3	2	3	1	5	3	1	1
5	3	1	3	1	1	3	1	4	5	1	4	2	4	3	1	1	1	4	3	1	1
6	2	2	1	3	1	2	2	4	1	1	4	1	4	1	3	4	4	5	3	3	2
7	3	2	1	4	1	3	1	5	2	3	3	1	5	2	1	3	1	5	2	1	1
8	1	1	1	1	1	1	1	5	5	1	1	1	5	5	1	1	1	5	3	1	1
9	1	1	1	1	1	1	1	3	3	1	1	1	1	3	1	1	1	3	3	1	1
10	3	1	1	3	2	3	1	5	5	1	1	1	5	1	1	1	1	5	3	1	1
Mean	1,80	1,20	1,20	2,00	1,10	1,70	1,20	3,90	3,60	1,50	2,10	1,40	4,30	2,80	1,60	1,90	1,50	4,40	3,10	1,20	1,10
SD	0,87	0,40	0,60	1,00	0,30	0,90	0,40	1,22	1,28	0,81	1,14	0,92	1,19	1,17	1,02	1,14	1,02	0,80	0,54	0,60	0,30

	Tracking Stage 1				Tracking Stage 2							
Participant#	Q5	Q7	Q8	Q9	Q5	Q6	Q7	Q8	Q9	Age	Gender	Experience with AR
1	1	1	4	3	1	2	1	3	5	23	Männlich	3
2	1	1	4	4	2	1	1	3	4	26	Männlich	3
3	1	1	5	3	1	1	1	4	3	36	Weiblich	4
4	1	1	4	3	1	1	1	5	3	23	Männlich	5
5	1	1	4	5	1	1	1	5	5	21	Männlich	5
6	2	1	5	1	2	1	1	5	1	23	Männlich	4
7	3	1	5	2	3	1	1	5	2	25	Weiblich	5
8	1	1	5	3	1	1	1	5	3	54	Weiblich	4
9	1	1	3	4	1	1	1	4	5	32	Männlich	5
10	1	1	5	3	1	1	1	5	5	30	Weiblich	3
Mean	1,30	1,00	4,40	3,10	1,40	1,10	1,00	4,40	3,60	29,30		4,10
SD	0,64	0,00	0,66	1,04	0,66	0,30	0,00	0,80	1,36	54,00		5,00

Material From User Study About the Combination of AR and VR Nuggets (Section 6.5)

Participant# / Transition	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	pragmatic quality	hedonic quality	attractiveness	self-location presence	possible actions presence	overall presence
Indicator																								
A11	2	4	3	3	2	4	2	3	5	2	2	3	2	2	2	2	2	2	3,0	2,8	3,5	2,3	2,0	2,1
A6	4	4	3	2	0	5	4	3	2	6	1	1	1	1	4	4	1	3	2,3	4,3	3,5	1,0	3,0	2,0
A2	0	5	0	0	0	4	2	0	6	2	1	2	2	3	4	4	5	2	1,5	2,0	2,5	2,0	3,8	2,9
A7	0	5	1	1	3	4	4	1	5	3	3	3	3	2	3	3	4	3	2,3	3,0	3,0	2,8	3,3	3,0
A3	0	3	4	2	2	4	6	5	2	5	1	1	1	1	1	1	1	1	2,0	4,3	4,0	1,0	1,0	1,0
A8	4	3	3	2	4	3	5	2	2	3	2	1	1	1	1	1	1	3	3,3	3,3	2,5	1,3	1,5	1,4
A4	2	5	1	0	2	4	3	2	0	3	1	1	1	1	1	5	4	4	1,3	2,5	3,5	1,0	3,5	2,3
A9	4	4	3	1	4	5	6	1	2	5	3	2	2	3	2	2	1	3	3,3	4,3	2,5	2,5	2,0	2,3
A5	1	4	0	2	2	4	3	0	5	0	1	5	1	1	1	2	1	4	2,0	2,3	2,0	2,0	2,0	2,0
A10	4	4	2	3	2	4	3	3	2	2	1	1	1	2	1	1	1	1	2,5	3,0	3,5	1,3	1,0	1,1
																		⌀	2,3	3,2	3,1	1,7	2,3	2,0
																		SD	0,6	0,8	0,6	0,6	0,9	0,6
Fade																								
A2	0	3	0	3	0	4	4	0	6	1	1	3	1	1	3	1	1	3	1,5	3,0	1,5	1,5	2,0	1,8
A7	0	4	3	3	3	4	5	2	3	3	3	4	2	3	3	2	2	3	2,3	3,8	3,0	3,0	2,5	2,8
A11	1	5	1	1	5	5	5	2	5	4	2	2	2	2	3	3	2	2	3,0	3,8	3,5	2,0	2,5	2,3
A6	5	6	3	0	3	6	6	0	0	4	1	1	1	1	2	3	1	2	2,8	4,0	3,0	1,0	2,0	1,5
A3	4	5	5	0	5	4	5	4	2	5	2	4	1	1	4	4	2	4	4,0	3,5	4,5	2,0	3,5	2,8
A8	2	5	2	1	2	4	5	3	4	4	1	1	1	1	3	3	3	3	2,5	3,5	4,0	1,0	3,0	2,0
A4	0	5	0	3	3	6	5	0	6	3	1	1	1	1	3	4	5	4	2,3	4,3	2,5	1,0	4,0	2,5
A9	0	5	1	2	3	5	4	0	6	3	1	2	1	1	2	1	3	2	2,5	3,5	2,5	1,3	2,0	1,6
A5	2	5	2	4	3	4	1	2	4	0	1	4	3	3	5	5	1	4	2,8	2,3	3,5	2,8	3,8	3,3
A10	2	3	5	3	4	2	4	3	4	1	3	4	1	3	5	3	4	2	3,8	2,5	3,0	2,8	3,5	3,1
																		⌀	2,7	3,4	3,1	1,8	2,9	2,4
																		SD	0,7	0,6	0,8	0,7	0,7	0,6
Arrow																								
A3	4	5	2	0	1	6	6	0	2	6	1	1	1	1	1	1	1	1	2,3	4,5	2,5	1,0	1,0	1,0
A8	4	4	1	2	5	3	5	3	2	3	4	2	1	4	3	1	2	3	3,0	3,3	3,5	2,8	2,3	2,5

Participant# / Transition	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	pragmatic quality	hedonic quality	attractiveness	self-location presence	possible actions presence	overall presence
A11	2	4	2	2	4	4	3	2	5	4	2	2	2	2	3	2	2	3	3,3	3,3	3,0	2,0	2,5	2,3
A6	0	5	1	1	3	6	3	0	6	3	3	3	1	2	1	4	1	2	2,5	3,3	2,5	2,3	2,0	2,1
A2	0	2	0	3	1	4	1	1	6	1	1	2	1	1	4	3	1	1	1,8	2,3	1,5	1,3	2,3	1,8
A7	1	3	2	2	2	4	3	2	5	3	2	2	2	2	3	2	2	2	2,5	3,0	2,5	2,0	2,3	2,1
A4	4	4	5	2	4	2	4	4	1	3	1	1	1	1	5	5	5	4	3,5	2,8	4,0	1,0	4,8	2,9
A9	0	5	1	1	2	3	5	1	6	4	1	1	1	3	3	2	2	3	2,3	3,3	3,0	1,5	2,5	2,0
A5	1	4	0	1	3	4	4	0	4	2	2	4	2	3	5	5	4	5	2,0	2,8	2,0	2,8	4,8	3,8
A10	4	4	2	5	4	5	5	1	3	4	4	4	1	1	2	1	1	2	3,3	4,8	2,5	2,5	1,5	2,0
																		⌀	2,6	3,3	2,7	1,9	2,6	2,2
																		SD	0,5	0,7	0,6	0,6	1,1	0,7
Portal																								
A4	0	5	0	1	3	6	4	0	6	3	1	1	1	1	1	2	1	1	2,3	3,5	2,5	1,0	1,3	1,1
A9	2	3	1	2	1	6	3	3	5	4	3	2	2	3	2	2	1	1	2,3	3,8	3,0	2,5	1,5	2,0
A11	1	5	1	1	5	5	5	1	6	4	2	2	2	2	2	2	2	2	3,3	3,8	3,0	2,0	2,0	2,0
A6	0	6	0	1	2	6	6	0	2	3	1	1	1	1	4	4	2	3	1,0	4,0	3,0	1,0	3,3	2,1
A2	0	3	1	2	0	3	1	1	6	2	2	3	2	3	4	5	3	5	1,8	2,0	2,0	2,5	4,3	3,4
A7	2	5	3	3	3	3	3	3	3	2	1	2	2	2	3	3	3	2	2,8	2,8	4,0	1,8	2,8	2,3
A3	0	5	0	2	0	6	5	0	6	5	1	1	1	1	1	1	1	1	1,5	4,5	2,5	1,0	1,0	1,0
A8	1	4	2	2	2	4	5	2	5	4	2	2	1	1	1	2	1	2	2,5	3,8	3,0	1,5	1,5	1,5
A5	5	4	4	1	5	4	5	2	0	4	2	4	3	3	5	5	5	5	3,5	3,5	3,0	3,0	5,0	4,0
A10	5	6	0	0	5	5	5	1	5	5	1	1	1	1	1	1	1	1	3,8	3,8	3,5	1,0	1,0	1,0
																		⌀	2,5	3,5	3,0	1,73	2,4	2,0
																		SD	0,8	0,6	0,5	0,68	1,3	0,9
Hand																								
A5	0	4	0	3	2	4	2	0	6	1	1	4	3	3	5	5	1	4	2,0	2,5	2,0	2,8	3,8	3,3
A10	1	2	3	5	5	2	2	5	2	1	2	2	2	1	2	2	1	1	2,8	2,5	3,5	1,8	1,5	1,6
A11	0	5	0	1	5	5	5	1	6	5	1	1	1	1	2	2	2	2	2,8	4,0	3,0	1,0	2,0	1,5
A6	0	6	0	0	2	6	6	0	4	5	3	3	1	1	2	2	3	3	1,5	4,3	3,0	2,0	2,5	2,3
A2	1	6	5	0	4	5	6	2	2	6	2	4	3	3	5	4	5	4	3,0	4,3	4,0	3,0	4,5	3,8

Participant# / Transition	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	pragmatic quality	hedonic quality	attractiveness	self-location presence	possible actions presence	overall presence
A7	2	2	3	2	3	2	4	3	2	3	2	3	2	3	3	2	3	3	2,5	2,8	2,5	2,5	2,8	2,6
A3	1	6	1	0	1	4	6	0	6	4	1	1	1	1	2	2	1	1	2,3	3,5	3,0	1,0	1,5	1,3
A8	4	3	2	2	4	4	4	2	3	3	2	2	1	1	1	2	1	3	3,3	3,3	2,5	1,5	1,8	1,6
A4	6	2	6	1	5	1	5	4	0	5	1	1	1	1	1	2	1	2	4,3	3,0	3,0	1,0	1,5	1,3
A9	4	2	4	3	3	4	2	3	3	2	3	4	3	2	4	3	2	3	3,5	2,8	2,5	3,0	3,0	3,0
																		∅	2,8	3,3	2,9	2,0	2,5	2,2
																		SD	0,7	0,6	0,5	0,7	0,9	0,8
None																								
B1	3	3	4	3	3	3	3	3	3	3	4	4	3	3	3	3	3	2	3,3	3,0	3,0	3,5	2,8	3,1
B6	0	6	0	0	5	4	6	0	5	5	4	1	1	1	1	1	1	1	2,5	3,8	3,0	1,8	1,0	1,4
B2	0	6	0	0	0	6	6	0	6	6	1	1	1	1	1	1	1	1	1,5	4,5	3,0	1,0	1,0	1,0
B7	0	6	0	3	0	3	3	1	6	3	1	5	3	1	1	1	1	4	1,5	3,0	3,5	2,5	1,8	2,1
B3	0	5	0	0	1	5	5	1	6	5	2	1	1	1	2	1	1	2	1,8	3,8	3,0	1,3	1,5	1,4
B8	0	1	2	1	1	3	2	4	2	1	4	4	2	2	4	4	2	4	1,3	1,8	2,5	3,0	3,5	3,3
B4	0	4	0	0	2	6	6	0	6	4	1	1	1	1	3	2	4	1	2,0	4,0	2,0	1,0	2,5	1,8
B9	0	5	1	3	3	6	5	2	3	3	2	2	2	1	4	2	1	3	1,8	4,3	3,5	1,8	2,5	2,1
B6	1	3	5	3	1	3	3	2	4	1	2	3	2	2	4	2	2	4	2,8	2,5	2,5	2,3	3,0	2,6
B10	2	5	4	3	1	5	4	3	3	3	2	2	1	1	4	5	2	3	2,5	3,8	4,0	1,5	3,5	2,5
AR2																								
B1	3	3	3	3	3	3	3	3	3	3	2	3	3	3	3	3	3	2	3,0	3,0	3,0	2,8	2,8	2,8
B6	0	0	5	5	1	1	2	3	6	2	4	5	5	3	4	4	5	4	3,0	2,5	1,5	4,3	4,3	4,3
B2	0	6	0	0	1	6	6	0	6	5	1	1	1	2	1	1	1	1	1,8	4,3	3,0	1,3	1,0	1,1
B7	0	4	1	1	0	3	2	1	6	2	1	2	3	1	1	1	1	5	1,8	2,0	2,5	1,8	2,0	1,9
B3	0	3	0	4	0	3	3	1	6	3	1	2	1	2	2	2	1	2	1,5	3,3	2,0	1,5	1,8	1,6
B8	0	6	0	1	1	5	5	1	3	5	2	4	1	2	4	2	4	4	1,0	4,0	3,5	2,3	3,5	2,9
B4	0	6	0	0	2	3	4	1	6	4	2	2	5	1	4	3	2	1	2,0	2,8	3,5	2,5	2,5	2,5
B9	0	3	2	3	2	6	4	2	3	3	1	2	3	3	3	2	1	3	1,8	4,0	2,5	2,3	2,3	2,3
B6	1	3	3	5	3	2	3	2	3	2	2	2	2	2	4	2	2	3	2,5	3,0	2,5	2,0	2,8	2,4
B10	0	6	1	2	3	6	4	1	6	6	1	1	1	1	1	1	2	1	2,5	4,5	3,5	1,0	1,3	1,1

Participant# / Transition	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	pragmatic quality	hedonic quality	attractiveness	self-location presence	possible actions presence	overall presence
VR2																								
B1	1	3	3	3	3	3	3	3	3	3	2	2	3	2	2	1	2	1	2,5	3,0	3,0	2,3	1,5	1,9
B6	3	6	0	0	6	6	6	0	2	6	1	1	1	1	4	1	1	1	2,8	4,5	3,0	1,0	1,8	1,4
B2	0	6	0	0	4	6	6	0	5	6	1	1	1	1	1	1	1	1	2,3	4,5	3,0	1,0	1,0	1,0
B7	3	4	3	2	0	3	1	3	6	6	3	5	2	2	1	1	1	5	3,0	3,0	3,5	3,0	2,0	2,5
B3	1	5	0	2	1	4	5	1	4	3	2	2	2	2	2	2	2	3	1,5	3,5	3,0	2,0	2,3	2,1
B8	2	5	4	3	2	5	3	1	2	5	4	4	4	3	4	2	2	4	2,5	4,0	3,0	3,8	3,0	3,4
B4	3	5	5	2	2	6	6	0	2	6	3	3	1	2	4	2	1	5	3,0	5,0	2,5	2,3	3,0	2,6
B9	1	5	0	2	3	6	5	1	3	5	3	2	2	2	4	1	1	3	1,8	4,5	3,0	2,3	2,3	2,3
B6	2	4	4	3	5	2	3	2	2	3	3	2	4	4	5	5	5	4	3,3	2,8	3,0	3,3	4,8	4,0
B10	0	4	3	1	0	5	4	1	4	6	1	1	1	1	1	1	1	2	1,8	4,0	2,5	1,0	1,3	1,1
AR3																								
B1	3	3	3	3	3	3	3	3	3	3	2	3	2	2	3	3	3	2	3,0	3,0	3,0	2,3	2,8	2,5
B6	0	1	0	5	0	5	5	1	6	1	1	1	1	1	1	1	1	1	1,5	4,0	1,0	1,0	1,0	1,0
B2	0	6	0	0	1	6	6	0	6	6	1	1	1	1	1	1	1	1	1,8	4,5	3,0	1,0	1,0	1,0
B7	0	4	2	2	3	4	3	2	3	2	2	5	2	2	1	1	1	4	2,0	2,8	3,0	2,8	1,8	2,3
B3	0	4	1	3	0	5	3	1	3	2	2	3	1	2	3	2	3	3	1,0	3,3	2,5	2,0	2,8	2,4
B8	1	5	4	1	5	1	5	4	1	5	2	2	1	3	4	4	1	4	2,8	3,0	4,5	2,0	3,3	2,6
B4	1	6	0	1	6	6	6	0	2	6	1	1	1	1	2	2	1	2	2,3	4,8	3,0	1,0	1,8	1,4
B9	0	6	0	1	4	5	5	1	3	6	3	2	2	2	4	2	1	3	1,8	4,3	3,5	2,3	2,5	2,4
B6	4	5	2	1	6	5	6	0	3	6	2	3	4	2	5	4	4	4	3,8	4,5	2,5	2,8	4,3	3,5
B10	0	6	1	2	1	6	5	0	6	6	2	1	1	2	1	1	1	1	2,0	4,8	3,0	1,5	1,0	1,3
VR3																								
B1	3	3	3	3	3	3	3	3	3	3	3	3	2	2	4	2	2	2	3,0	3,0	3,0	2,5	2,5	2,5
B6	6	6	0	0	6	6	6	0	6	6	1	1	1	1	1	1	1	1	4,5	4,5	3,0	1,0	1,0	1,0
B2	0	6	0	0	1	6	6	0	6	6	1	1	1	1	1	1	1	1	1,8	4,5	3,0	1,0	1,0	1,0
B7	2	6	1	0	1	6	6	0	3	6	2	4	2	2	1	2	2	4	1,8	4,5	3,0	2,5	2,3	2,4
B3	0	4	1	2	1	4	4	1	5	3	1	1	1	1	3	3	3	4	1,8	3,3	2,5	1,0	3,3	2,1
B8	5	4	4	0	5	2	5	3	1	4	2	1	1	2	5	5	4	5	3,8	2,8	3,5	1,5	4,8	3,1

Participant# / Transition	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	pragmatic quality	hedonic quality	attractiveness	self-location presence	possible actions presence	overall presence
B4	4	6	0	0	6	6	6	0	0	6	1	1	1	1	1	1	1	2	2,5	4,5	3,0	1,0	1,3	1,1
B9	0	6	0	0	5	6	6	0	3	6	2	2	2	2	3	1	1	2	2,0	4,5	3,0	2,0	1,8	1,9
B6	1	5	5	1	6	4	6	0	3	6	2	3	1	2	4	4	3	4	3,8	4,3	2,5	2,0	3,8	2,9
B10	0	4	2	2	3	2	3	3	4	2	4	5	4	3	5	4	5	5	2,3	2,3	3,5	4,0	4,8	4,4
⌀ A	1,1	4,4	1,8	1,8	2,4	4,3	4,3	1,5	3,9	4,0	2,0	2,3	1,9	1,8	2,6	2,1	1,9	2,6	**2,58**	**3,33**	2,94	1,82	2,52	**2,17**
SD ⌀ A	1,5	1,6	1,8	1,4	1,8	1,5	1,4	1,2	1,7	1,6	1,0	1,3	1,1	0,8	1,4	1,1	1,2	1,3	0,67	0,66	0,60	0,67	0,99	0,71
⌀ B	2,1	5,0	1,6	0,8	3,7	4,5	5,1	1,0	3,4	4,8	1,9	2,2	1,6	1,7	2,8	2,4	2,3	3,0	**2,29**	**3,66**	2,92	2,00	2,36	**2,18**
SD ⌀ B	2,2	1,1	1,7	1,1	2,1	1,6	1,2	1,3	1,9	1,5	0,9	1,4	0,9	0,6	1,6	1,4	1,3	1,5	0,76	0,82	0,57	0,85	1,08	0,98
⌀ A&B	1,3	4,5	1,8	1,6	2,6	4,3	4,4	1,4	3,8	4,2	2,0	2,3	1,9	1,8	2,7	2,1	2,0	2,7	**2,44**	**3,50**	2,93	1,91	2,44	**2,17**
SD ⌀ A&B	1,7	1,5	1,8	1,4	1,9	1,6	1,4	1,3	1,7	1,6	1,0	1,3	1,1	0,8	1,4	1,2	1,2	1,3	0,80	0,87	0,63	0,85	1,06	0,95

Material From User Study About ARNAUDDI (Section 6.6)

Participant#	complicated - simple	ugly - attractive	impractical - practical	tacky - stylish	table - predictable	cheap - premium	unimaginative - creative	bad - good	complicated - simple	dull - captivating
1	0	1	2	2	0	1	6	2	0	3
2	0	0	0	0	0	1	0	0	3	0
3	1	1	1	1	2	3	5	2	0	4
4	2	1	4	3	2	3	5	4	2	3
5	2	1	0	3	4	3	3	1	0	1
6	4	2	2	2	3	3	5	3	4	3
7	3	1	3	2	1	2	4	2	2	0
8	3	2	3	2	1	2	4	3	1	1
9	3	2	3	2	1	2	4	3	1	0
10	3	2	3	2	1	2	4	3	1	0
11	2	2	4	3	1	2	5	3	3	4
12	2	1	3	2	1	2	5	2	1	2
13	2	2	2	1	2	5	3	2	1	2
14	4	3	5	2	4	4	5	4	4	2
15	2	2	1	2	1	1	3	1	2	0
16	4	2	5	4	5	4	4	5	2	3
17	2	3	5	3	2	2	6	5	2	3
18	3	4	6	4	3	3	4	5	2	3
19	2	2	2	3	2	4	5	5	4	3
20	1	2	3	1	2	2	5	3	2	4
21	0	1	5	4	1	3	6	2	2	4
22	1	2	2	5	2	3	5	2	2	4
23	3	4	6	4	3	4	4	4	3	4
24	5	2	0	4	3	3	5	3	1	3
25	2	3	3	4	2	2	6	2	3	6
26	2	1	2	4	1	3	6	4	1	4
27	2	5	5	4	3	5	6	5	2	3
28	1	1	3	2	3	2	5	1	1	2
29	4	3	5	3	3	4	5	5	5	3
30	3	1	2	3	0	4	5	3	2	4
31	2	4	4	3	2	4	4	4	2	3
32	3	1	5	5	5	2	6	5	1	3
33	5	2	5	4	5	4	4	4	5	5
34	5	5	5	5	3	4	6	5	5	4
35	1	2	1	0	3	2	5	3	2	2
36	1	1	4	1	2	2	5	3	3	4
37	2	1	2	2	0	2	5	2	1	2
38	3	1	4	1	1	4	6	3	6	4
39	0	3	3	3	0	3	5	2	1	3
40	3	3	5	5	4	3	4	5	4	4
41	1	2	4	2	3	3	3	3	1	2
42	3	4	3	4	2	4	5	3	2	4
43	1	1	3	3	1	3	4	2	1	4
44	2	4	2	3	6	5	5	6	0	3
45	2	1	4	3	1	1	6	2	1	6
46	2	3	4	3	2	1	5	2	5	4
47	5	4	4	3	2	4	5	4	2	4
48	4	1	0	2	0	3	5	2	6	2
ø	2,35	2,13	3,17	2,77	2,10	2,88	4,71	3,10	2,23	2,94
SD	1,33	1,20	1,60	1,25	1,45	1,09	1,10	1,34	1,56	1,43

PQ	2,5
HQ	3,3
ATT	2,6

Participant#	Q1a	Q1b	Q1c	Q1d	Q1e	Q1f	Q1g	Q2	Q3
1	Computer	Computer	Computer	Computer	Computer	Computer	Computer	7	7
2	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
3	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	3	3
4	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	7	7
5	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
6	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
7	Computer	Computer	Computer	Computer	Computer	Computer	Computer	7	7
8	Computer	Computer	Computer	Computer	Computer	Computer	Computer	7	7
9	Computer	Computer	Computer	Computer	Computer	Computer	Computer	7	7
10	Computer	Computer	Computer	Computer	Computer	Computer	Computer	7	7
11	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	1	4
12	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	Computer	6	6
13	AR -Gerät	AR -Gerät	AR -Gerät	Computer	AR -Gerät	Computer	AR -Gerät	1	3
14	Computer	Computer	Computer	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	5	5
15	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	1	5
16	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	4	4
17	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	6	5
18	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	3	4
19	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	4	4
20	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	5	5
21	Computer	Computer	Computer	AR -Gerät	Computer	Computer	AR -Gerät	7	7
22	Computer	Computer	Computer	AR -Gerät	Computer	Computer	AR -Gerät	7	7
23	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
24	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	6	6
25	Computer	Computer	Computer	AR -Gerät	Computer	Computer	AR -Gerät	6	6
26	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	2	2
27	Computer	Computer	Computer	Computer	Computer	Computer	Computer	2	2
28	Computer	Computer	Computer	Computer	Computer	Computer	Computer	7	7
29	Computer	Computer	Computer	Computer	Computer	Computer	Computer	5	5
30	Computer	Computer	Computer	Computer	Computer	Computer	Computer	1	4
31	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
32	Computer	Computer	Computer	Computer	Computer	Computer	Computer	4	4
33	Computer	Computer	Computer	Computer	Computer	Computer	Computer	2	2
34	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
35	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	5	7
36	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	6	6
37	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
38	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	6	6
39	Computer	Computer	AR -Gerät	Computer	Computer	AR -Gerät	AR -Gerät	6	2
40	Computer	Computer	Computer	Computer	Computer	Computer	Computer	5	4
41	Computer	Computer	Computer	AR -Gerät	AR -Gerät	AR -Gerät	AR -Gerät	6	6
42	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	3	5
43	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
44	Computer	Computer	Computer	Computer	Computer	Computer	Computer	1	1
45	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
46	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	2	6
47	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	5	6
48	Computer	Computer	Computer	Computer	Computer	Computer	AR -Gerät	7	7
ø								**5,10**	**5,44**
SD								**2,11**	**1,74**

Participant#	Q4	Q7a	Q7b	Q7c	Q7d	How often do you use AR applications?	experience with authoring AR applications	age	gender
1	4	7	7	7	7	1	1	21	Männlich
2	4	7	7	7	7	1	1	26	Männlich
3	7	5	7	3	2	2	1	21	Männlich
4	4	7	7	7	2	1	1	24	Männlich
5	4	not used	4	not used	not used	1	2	24	Männlich
6	4	3	4	not used	not used	1	1	34	Männlich
7	4	4	1	5	5	3	3	22	Männlich
8	4	5	6	not used	not used	1	1	25	Weiblich
9	4	5	6	not used	not used	1	1	23	Männlich
10	4	5	6	not used	not used	1	1	24	Männlich
11	3	5	not used	not used	not used	1	1	24	Männlich
12	2	1	not used	not used	1	1	1	23	Männlich
13	6	6	6	6	5	2	3	32	Männlich
14	5	4	6	not used	not used	2	3	24	Männlich
15	4	4	4	not used	not used	2	2	22	Männlich
16	1	5	5	5	5	2	1	30	Männlich
17	2	5	3	7	7	1	1	24	Männlich
18	4	4	2	5	not used	1	1	24	Männlich
19	4	6	3	7	7	2	1	25	Männlich
20	5	3	5	not used	not used	4	2	23	Männlich
21	6	7	7	7	3	2	2	23	Männlich
22	5	5	not used	not used	not used	7	1	28	Männlich
23	6	2	2	2	1	1	1	22	Weiblich
24	2	4	5	6	not used	2	5	23	Männlich
25	6	4	4	5	5	1	1	23	Männlich
26	4	6	not used	7	not used	2	1	29	Männlich
27	2	7	7	not used	not used	2	1	29	Männlich
28	4	6	6	not used	not used	1	3	22	Männlich
29	4	2	5	not used	6	1	1	29	Männlich
30	3	3	6	not used	not used	2	1	21	Männlich
31	6	2	not used	4	not used	2	1	23	Weiblich
32	4	1	not used	not used	3	1	1	26	Männlich
33	7	7	7	7	7	1	1		
34	4	5	5	5	1	4	5	24	Männlich
35	5	6	7	not used	not used	1	1	21	Männlich
36	4	5	7	7	not used	2	1	20	Männlich
37	4	6	6	6	6	1	2	22	Männlich
38	3	5	5	5	5	1	1	22	Männlich
39	7	6	7	7	not used	2	1	22	Männlich
40	3	3	3	4	4	1	1	26	Männlich
41	5	not used	6	6	2	2	2		
42	4	4	4	7	6	2	2	25	1x Männlich, 1x Weiblich
43	4	7	7	7	not used	1	1	22	Männlich
44	2	7	7	7	7	1	1		
45	6	3	7	not used	4	2	1	22	Männlich
46	4	7	6	not used	not used	2	1	20	Männlich
47	2	1	not used	not used	not used	1	1	23	Männlich
48	3	6	7	not used	not used	3	4	27	Weiblich
⌀	**4,13**	**4,74**	**5,41**	**5,85**	**4,50**	**1,71**	**1,54**	**24,20**	
SD	**1,39**	**1,76**	**1,64**	**1,38**	**2,10**	**1,08**	**1,02**	**3,07**	

Participant#	Q8. sat	Q9. sat	Q10. sat	Q13. sat	Q16. sat	Q8. q	Q9. q	Q10. q	Q13. q	Q16. q	Q8.s z	Q9.s z	Q10. sz	Q13. sz	Q16. sz
1	4	3	3	4	7	5	5	5	5	7	5	4	4	7	7
2	4	4	4	4	4	4	4	4	4	4	4	4	4	5	4
3	3	5	4	7	7	2	2	2	7	7	3	7	6	7	7
4	2	2	3	4	7	3	3	3	6	5	6	3	3	4	4
5	1	4	3	4	1	5	4	5	6	1	2	1	2	3	1
6	4	3	3	5	5	3	5	5	6	2	4	6	5	6	6
7	4	6	6	5	5	4	6	6	5	4	4	7	6	6	4
8	6	7	6	7	1	7	7	7	7	1	4	1	3	2	6
9	6	7	6	7	1	7	7	7	7	1	4	1	3	2	6
10	6	7	5	7	1	7	7	7	7	1	4	1	3	2	6
11	6	2	4	3	3	6	6	5	3	2	4	5	4	3	3
12	2	3	2	6	4	3	6	6	4	4	1	2	2	5	5
13	4	5	3	5	7	6	4	7	3	5	1	2	3	4	2
14	3	2	2	5	6	6	5	6	6	5	2	3	3	5	6
15	3	1	2	2	2	5	2	6	6	3	1	1	3	1	1
16	5	5	6	6	7	6	6	6	7	7	6	6	6	7	6
17	6	6	6	6	3	3	2	6	6	3	4	2	3	2	3
18	5	4	4	5	3	3	4	7	5	3	4	4	4	4	4
19	7	6	6	5	5	2	5	3	4	5	5	3	2	3	3
20	2	5	1	4	5	5	3	5	6	5	3	2	2	2	5
21	5	3	4	6	7	2	7	4	6	3	2	5	3	7	7
22	5	3	3	6	6	3	1	4	6	3	2	3	2	4	6
23	4	3	4	6	7	4	5	5	6	7	4	6	6	4	7
24	4	2	3	6	7	3	2	4	6	7	4	2	4	7	7
25	5	5	4	4	7	5	4	6	5	6	4	5	5	5	6
26	5	2	3	7	5	7	6	5	7	5	4	2	3	2	5
27	6	6	6	6	4	5	6	7	6	6	5	6	6	4	5
28	3	4	5	2	3	2	2	5	2	2	1	3	2	3	5
29	3	5	3	6	6	2	6	4	6	6	5	3	4	6	4
30	2	4	3	3	4	6	1	3	6	4	3	3	3	3	4
31	5	1	3	5	6	6	4	6	5	2	4	3	4	3	4
32	7	5	5	5	4	5	4	4	4	4	6	2	2	3	6
33	2	3	3	6	2	1	1	4	5	1	2	2	2	5	2
34	6	7	7	6	5	6	5	7	7	3	5	3	5	7	6
35	3	4	2	2	5	2	2	3	1	6	6	6	3	1	5
36	6	5	3	5	6	3	2	5	5	4	6	5	5	6	2
37	4	4	6	3	3	4	2	3	4	3	4	3	2	3	3
38	4	5	2	6	7	7	6	6	6	3	3	2	1	1	7
39	1	2	2	6	7	2	2	4	7	6	1	1	2	5	7
40	5	5	5	7	7	6	6	6	7	7	4	5	5	7	7
41	2	3	2	5	3	2	5	4	6	4	5	5	4	3	4
42	6	6	4	7	5	3	3	3	4	4	3	3	3	4	4
43	6	6	6	6	6	5	4	4	4	4	1	1	1	1	1
44	3	2	2	5	3	5	6	5	4	4	3	2	2	1	4
45	6	6	4	7	7	6	2	3	6	7	7	2	7	2	7
46	3	6	6	3	6	4	3	6	3	3	5	2	6	6	5
47	2	2	2	2	6	7	6	6	6	6	2	6	6	6	6
48	6	4	6	6	7	6	3	4	7	7	1	1	1	4	7
ø	4,21	4,17	3,90	5,10	4,90	4,40	4,15	4,96	5,35	4,21	3,60	3,27	3,54	4,02	4,83
SD	1,62	1,67	1,54	1,48	1,93	1,74	1,81	1,35	1,42	1,88	1,59	1,79	1,55	1,91	1,77

Material From the Authoring Workshop with Integrated Authoring Tools (Section 6.7)

Participant#	Group	Q1	Q2a	Q2b	Q2c	Q2d	Q2e	Q2f	Q2g	Q2h	Q7	Q8	Q10	Q11	Q12	Q7	Q8	Q10	Q11	Q12
											Show & Tell					Compare				
1	A	5	1	1	2	2	4	1	2	4	1	6	5	7	7	1	6	5	6	7
2	A	5	1	3	2	3	3	1	2	3	1	4	1	6	7					
3	A	6	2	1	2	2	3	2	2	2	2	3	4	3	5	2	3	2	2	7
4	A	3	1	1	4	4	4	1	2	4						2	3	5	2	4
5	A	4	1			3		2			1	2	2	6	7	1	1	1	1	3
6	A	6	2	1	4	2	2	1	2	2	1	2	3	6	7					
7	A	5	1	1	4	2	4	6	6	4	2	4	5	5	7					
8	A	1	5	5	7	7	7	2	7	7	4	7	1	7	1					
9	B	2	1	1	4	3	1	1	3	1	3	5	7	7	7					
10	B	6	1	1	1	1	4	1	1	1	2	4	5	6	2					
11	B	2	1								2	4	1	7	7					
12	B	2	3	4	4	5	4	2	2	4	4	4	5	5	3					
13	B	4	3	1	5	4	1	1	4	5	4	4	4	3	4					
14	B	5	3	7	6	4	4	3	3	5										
Mean		4,00	1,86	2,25	3,75	3,23	3,42	1,85	3,00	3,50	2,25	4,08	3,58	5,67	5,33	1,50	3,25	3,25	2,75	5,25
SD		1,65	1,19	1,96	1,69	1,53	1,55	1,35	1,73	1,71	1,16	1,38	1,89	1,37	2,17	0,50	1,79	1,79	1,92	1,79

Participant#	Group	Q7	Q8	Q10	Q11	Q12	Q7	Q8	Q10	Q11	Q12	Q7	Q8	Q10	Q11	Q12	Experience with AR	Experience Game Engines	Experience Authoring Tools	Age	Gender
		Quiz					Semantic Zoom					Transparency									
1	A	2	7	7	7	7	2	7	7	7	7	3	5	6	6	6	6	7	1	28	male
2	A	1	1	1	7	3						1	4	6	7	7	4	1	7	49	male
3	A	4	3	5	5	7	3	4	7	4	5	3	4	5	5	6	5	3	1	30	male
4	A						2	2	2	6	5	3	7	3	6	5	4	4	3		
5	A	1	6	6	7	7	4	7	6	7	7	4	7	6	7	7	5	7	5	27	female
6	A	2	5	6	7	6	6	6	6	7	6	2	2	3	6	5	5	6	2	26	male
7	A						2	6	7	7	3	1	4	5	7	7	4	6	3	28	male
8	A											4	7	1	4	1	7	1	7	41	female
9	B											3	4	3	4	4	6	4	2	22	female
10	B											1	2	7	7	5	6	4	1	24	male
11	B											4	4	1	7	4	7	3	5		
12	B											5	4	5	5	4	5	4	1	32	female
13	B											5	5	3	3	4	4	7	1	34	male
14	B																3	2	2	16	female
Mean		2,00	4,40	5,00	6,60	6,00	3,17	5,33	5,83	6,33	5,50	3,00	4,54	4,15	5,69	5,00	5,07	4,21	2,93	29,75	
SD		1,10	2,15	2,10	0,80	1,55	1,46	1,80	1,77	1,11	1,38	1,36	1,60	1,87	1,32	1,62	1,16	2,04	2,12	8,32	

Material From User Study About Constraint-Based Authoring (Section 6.8)

Participant#	**Virtual Object**	**Q1.1**	**Q1.2**	**Q1.3**	**Q2.3**
1	Christmas tree	Floor	1	5	1
	Yellow present box	Background	3	5	2
	Blue present box	Wall	1	3	2
	Garland	Ceiling	4	5	1
	Candy	Background	1	2	4
	Latern	Wall	5	4	5
2	Christmas tree	Floor	2	5	n.a.
	Yellow present box	Platform	4	5	1
	Blue present box	Platform	4	5	1
	Garland	Wall	3	3	5
	Candy	Background	3	2	n.a.
	Latern	Wall	2	5	5
3	Christmas tree	Floor	2	5	1
	Yellow present box	Platform	4	5	2
	Blue present box	Platform	4	5	2
	Garland	Wall	1	5	3
	Candy	Background	2	1	1
	Latern	Ceiling	3	5	n.a.
4	Christmas tree	Platform	2	3	5
	Yellow present box	Background	5	4	3
	Blue present box	Wall	2	5	2
	Garland	Wall	1	5	5
	Candy	Platform	1	3	n.a.
	Latern	Ceiling	5	4	n.a.
5	Christmas tree	Floor	2	5	2
	Yellow present box	Background	2	2	4
	Blue present box	Platform	3	4	1
	Garland	Wall	3	4	n.a.
	Candy	Background	1	2	1
	Latern	Wall	1	5	3
6	Christmas tree	Floor	3	5	1
	Yellow present box	Background	2	2	1
	Blue present box	Background	3	2	1
	Garland	Wall	1	4	4
	Candy	Background	2	1	4
	Latern	Ceiling	3	5	1
7	Christmas tree	Platform	4	5	n.a.
	Yellow present box	Floor	3	5	3
	Blue present box	Platform	2	2	n.a.
	Garland	Wall	5	4	1
	Candy	Ceiling	4	5	1
	Latern	Platform	1	1	n.a.

Participant#		**Experience**	**Age**	**Gender**
1		2	25	Männlich
2		1	28	Weiblich
3		1	33	Weiblich
4		3	23	Männlich
5		1	26	Weiblich
6		3	21	Männlich
7		1	23	Männlich
Mean		1,71	25,57	
SD		0,88	3,70	