Understanding and Supporting the Process of Learning about Augmented Reality and Virtual Reality Creation among New Creators

by

Narges Ashtari

M.Sc., Simon Fraser University, 2018
B.Sc., Art University of Isfahan, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Computing Science
Faculty of Applied Science

SIMON FRASER UNIVERSITY
Spring 2024

Declaration of Committee

Name: **Narges Ashtari**

Degree: **Doctor of Philosophy**

Thesis title: **Understanding and Supporting the Process of Learning about Augmented Reality and Virtual Reality Creation among New Creators**

Committee: **Chair:** Xing-Dong Yang
Associate Professor, Computing Science

Parmit K. Chilana
Supervisor
Associate Professor, Computing Science

Andrea Bunt
Committee Member
Professor, Computer Science
University of Manitoba

Wolfgang Stuerzlinger
Committee Member
Professor, Interactive Arts and Technology

Saba Alimadadi
Examiner
Assistant Professor, Computing Science

Stacey Scott
External Examiner
Professor, Computing Science
University of Guelph

Ethics Statement

The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

In recent years, the global technological landscape has witnessed a significant shift, with the widespread availability of consumer-level Augmented Reality (AR) and Virtual Reality (VR) devices. This democratization has opened doors for individuals from diverse backgrounds to start exploring the creative potential and tinkering with AR/VR applications.

The primary objective of this dissertation was to understand and support the needs and challenges encountered by new AR/VR creators, particularly those without formal training in designing and implementing immersive experiences. In pursuit of this goal, I conducted research involving a diverse group of AR/VR enthusiasts, including UI/UX designers, linguists, psychologists, and software engineers, each bringing their unique perspectives yet all being novices in AR/VR creation.

I first conducted interviews with 21 AR/VR new creators, which revealed that the barriers these creators faced in AR/VR, such as implementation, debugging, and testing, were more pronounced than in traditional web and mobile application development, mainly due to the rapid evolution of AR/VR hardware and software and the plethora of unknowns faced by creators. This rapid pace often left participants struggling with where to start and understanding the current state-of-the-art. To mitigate the challenges AR/VR new creators faced, I designed, built, and evaluated PONI platform allowing new AR/VR creators to locate relevant projects based on their programming and 3D modeling skills, development goals, and any constraints, such as time or budget.

Finally, recognizing a need for observational studies to complete our picture of new creators' process of learning about AR/VR development, I conducted an observational study with 12 software developers. This study assessed how developers new to AR approach the initial creation processes using a simplified development framework, the information resources they seek, and how their learning experience compares to the more mainstream 2D development. This stage shed light on the unique needs and challenges of emerging developers and implications for designing user-centered training and learning approaches for new developers in this field.

The central thesis of this dissertation is that *providing new AR/VR creators with personalized learning resources and assistance tailored and appropriated to their specific challenges* *can offer a useful and usable means to overcome their challenges in the initial stages of this multifaceted learning process.*

Keywords: AR/VR development; End-user development; Personalization; Software learnability; Information-seeking

Dedication

To Mohsen, my guiding star, who lights the darkest paths with his love.

Acknowledgements

Throughout my doctoral studies, I have had the extraordinary privilege of collaborating with numerous inspiring and supportive individuals, each of whom has played a significant role in my academic journey. For their invaluable help and guidance, I am profoundly grateful.

I extend my deepest appreciation to my senior supervisor, Dr. Parmit Chilana. Her dedicated mentorship, consistent encouragement, and insightful critiques have not only motivated me but also significantly shaped my approach to impactful, user-centric research. Parmit's dedication has been pivotal in my development as a researcher, continually opening doors and fostering my growth in countless ways.

I am also thankful to Dr. Andrea Bunt, Dr. Wolfgang Stuerzlinger, Dr. Joanna McGrenere, and Dr. Michael Nebeling for their critical feedback and deep insights at various stages of my research, which were crucial in refining both its focus and quality. My gratitude also extends to my examiners, Dr. Saba Alimadadi and Dr. Stacey Scott, whose probing questions and valuable suggestions enhanced the clarity and comprehensiveness of this dissertation.

My journey was also enriched by the support and love of my awesome friends Bita Azari, Faraz Shamshirdar, Nadia Ghobadipasha, Amir Parvardi, Yasaman Etesam, Ashkan Alinezhad, Kiana Mostaghasi, Mahsa Keramati, Payam Jome Yazdian, Pouya Ahmadvand, Nastaran Sedehi, Ladan Fathi, and many others. Their friendship and unwavering support have been a cornerstone of my PhD experience.

Special thanks go to the members of ixLab: Foroozan Daneshzand, Maryam Rezaie, Laton Vermette, David Wong, Rimika Chaudhury, Amir Jahanlou, Parsa Rajabi, Parnian Taghipour, Zezhong Wang, and others—who have shared this journey with me. I enjoyed every moment working in an environment filled with enthusiasm and creativity. The friendship and mutual support in ixLab have truly made it a special place to grow and learn. I am grateful to have been part of such a dynamic team where every day brought new learning opportunities and memorable experiences.

Finally, I owe everything to my family, my unwavering pillars of strength and support. To my mother, Badri, whose constant encouragement and belief in me have been my guiding lights; my father, Alireza, who has always been both my greatest advocate and my most treasured friend; and my sister, Maryam, whose loving support and wise advice have always kept me grounded. This degree is as much a reflection of their sacrifices and faith in me as it is of my own efforts. I share every part of this accomplishment with them.

Portions of this thesis were co-authored with other collaborators, all of whom have consented to the inclusion of our joint work in this dissertation.

Table of Contents

Declaration of Committee **ii**

Ethics Statement **iii**

Abstract **iv**

Dedication **vi**

Acknowledgements **vii**

Table of Contents **ix**

List of Tables **xiv**

List of Figures **xv**

1 Introduction **1**

1.1 Motivation 1
1.2 Research Overview 4
1.3 Contributions and dissertation structure 7

2 Background and related work **9**

2.1 State-of-the-art in authoring Augmented and/or Virtual Reality (AR/VR) . 9
2.1.1 Current developments and innovations in AR/VR authoring tools . 9
2.1.2 AR/VR creators as end-user developers 11
2.2 Supporting software learnability and information-seeking in HCI 12
2.2.1 Approaches to learning and seeking help 12
2.2.2 Innovations in help and tutorial systems 13
2.3 Core ideas and innovations in the field of personalization 13
2.3.1 Classifying personalizations along several axes 14
2.3.2 Methods of personalization 14
2.3.3 Methodologies for detecting software expertise 15
2.3.4 Personalization in formal learning and education 16
2.4 Summary 17

3 Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities **19**
3.1 Introduction . . . 19
3.2 Related work . . . 21
3.2.1 AR/VR authoring tools in research and practice . . . 21
3.2.2 Observations of AR/VR creators . . . 21
3.2.3 AR/VR creators as end-user developers . . . 22
3.3 Study design . . . 23
3.3.1 Participants and recruitment . . . 23
3.3.2 The interview protocol . . . 23
3.3.3 Data analysis . . . 24
3.3.4 Organization and presentation of result . . . 25
3.4 AR/VR creators and their motivations . . . 25
3.5 Barriers in understanding the AR/VR landscape . . . 27
3.5.1 Difficult to know where to start . . . 28
3.5.2 Difficult to make use of online learning resources . . . 29
3.5.3 Lack of concrete design guidelines and examples . . . 30
3.6 Barriers in designing and prototyping AR/VR experiences . . . 31
3.6.1 Difficult to design for the physical aspect of immersive experiences . 31
3.6.2 Difficult to plan and simulate motion . . . 32
3.6.3 Difficult to design story-driven immersive experiences . . . 33
3.7 Barriers in implementing and testing AR/VR applications . . . 34
3.7.1 Too many unknowns in development, testing, and debugging . . . 34
3.7.2 User testing and evaluation challenges . . . 36
3.8 Discussion . . . 39
3.8.1 Important to consider needs of AR/VR end-user developers . . . 39
3.8.2 Building learning opportunities into AR/VR tools . . . 39
3.8.3 Building AR/VR toolchains with integrated debugging and testing facilities . . . 41
3.8.4 Limitations . . . 42
3.9 Conclusions . . . 42

4 Using a Personalized Onboarding Approach for Getting Inspiration and Learning About AR/VR Creation **43**
4.1 Introduction . . . 43
4.2 Related work . . . 46
4.2.1 Challenges in creating AR/VR applications . . . 46
4.2.2 Innovations in software help and tutorial systems . . . 47
4.2.3 Personalization in formal learning . . . 48

4.3 Design considerations and goals 49
4.4 PONI: System design and implementation 51
4.4.1 Input Module: Defining user characteristics and desired outcomes . . 51
4.4.2 Suggestion module 52
4.4.3 Project description module 55
4.4.4 Implementation 55
4.5 Usability study 59
4.5.1 Baseline interface used for comparison 59
4.5.2 Participants 60
4.5.3 Study design and tasks 61
4.5.4 Study procedure and measures 61
4.5.5 Data analysis 62
4.5.6 Results 62
4.5.7 Areas of improvement 64
4.6 Discussion 64
4.6.1 Key Takeaways 64
4.6.2 Limitations 65
4.6.3 Expanding personalized approaches 66
4.6.4 Supporting long-term engagement 66
4.7 Conclusions 67

5 How New Developers Approach Augmented Reality Development Using Simplified Creation Tools **69**
5.1 Introduction 69
5.2 Related work 71
5.2.1 Tool innovations is AR application development 71
5.2.2 Domain-specific software development 72
5.2.3 Information Seeking in Software Development Tasks 73
5.3 Method 74
5.3.1 Participants and recruitment 74
5.3.2 In-Lab Observations and Task Design 74
5.3.3 Follow-up semi-structured interviews 77
5.3.4 Data analysis 78
5.4 Results 79
5.4.1 Overview of task completion 79
5.4.2 Overview of information-seeking activities 80
5.4.3 Challenges in information-seeking when developing AR 83
5.5 Discussion 88
5.5.1 Key takeaways 88

5.5.2 Enhancing AR development resources with problem solving strategies 89
5.5.3 Improving learning through in-context personalized approaches . . . 89
5.5.4 Leveraging help-seeking through Generative AI platforms 90
5.5.5 Enhancing user engagement and learning through adaptive feedback and milestone integration 90
5.5.6 Limitations 91
5.6 Conclusions 91

6 Reflections, implications, and future work 93
6.1 Key takeaways 93
6.2 Implications and promising directions for future work 95
6.2.1 Introducing the immersive dimension into the creative tools 95
6.2.2 Adapting to rapid technological evolution in AR/VR development . 95
6.2.3 Extending personalizations for power users in AR/VR development . 96
6.2.4 Extending the concept of PONI to community-driven AR/VR development 97
6.2.5 Personalizing AR/VR authoring for different domains 98
6.3 Limitations of the research methodology 98

7 Conclusion 100
7.1 Core research contributions 101
7.1.1 Empirical insights into how new AR/VR creators approach development process and their unique difficulties 101
7.1.2 Empirical insights into how new AR/VR creators approach help-seeking and their unique challenges 102
7.1.3 Design of PONI as a novel personalized learning platform for locating relevant AR/VR learning resources 102
7.1.4 Empirical insights into how software developers new to AR approach the development process and their use of AI-assisted tools 103
7.2 Secondary contributions 104
7.2.1 Empirical insights into challenges in designing physical aspects of AR/VR 104
7.3 Closing remarks 105

Bibliography 106

Appendix A Study materials 128
A.1 Appendix A.1.0 128
A.1.1 Pre-study Questionnaire 128
A.1.2 Interview Questions 129

A.2 Appendix A.2.0 130
A.2.1 Pre-study Questionnaire 130
A.2.2 Tasks 131
A.2.3 Follow-up Interview Questions 134
A.3 Appendix A.3.0 135
A.3.1 Pre-Study Questionnaire 135
A.3.2 Appendix A.3.2 135
A.3.3 Appendix A.3.3 136
A.3.4 Appendix A.3.4 136

List of Tables

Table 1.1 Diversity among new AR/VR creators across various phases of my research, highlighting their varied backgrounds and skillsets despite all being newcomers to the field. 3

Table 3.1 Summary of study participants. 24

Table 3.2 Summary and description of eight key barriers described by participants. 25

Table 3.3 Summary of different AR/VR creation approaches and key activities among different groups of creators 38

Table 4.1 To create augmented reality experiences, users are asked to identify the method used for triggering augmentation action. PONI adopts the clustering used in [214, 24, 256] to offer representative modes of interaction in AR and VR. 53

Table 4.2 Examples of the keywords used for finding AR/VR projects or tutorials on YouTube. We aimed for diversity in the technology used, level of complexity in 3D modeling and programming, hardware, software, and topics of the projects. 56

Table 4.3 The schema used for the curation of AR/VR projects. Authors curated each project by assigning them appropriate type, category, programming and 3D modelling level, output type, and development frameworks (Types, categories, and output type are based on categories introduced in [24, 214, 256]. 57

Table 5.1 Participants' demographic information, years of programming experience, role, and programming language proficiencies. All participants indicated they had completed introductory courses in 3D geometry and linear algebra, either at the high school or university level. 75

List of Figures

Figure 1.1 List of contributions in each chapter and main research questions addressed in this dissertation. 7

Figure 2.1 Classification of AR/VR Authoring Tools by Fidelity and Skill/Resources Requirements. Adapted from Nebeling et al. [184], this diagram categorizes AR/VR authoring tools based on (x-axis) potential fidelity in AR/VR experiences and (y-axis) required skills and resources (e.g., 3D models, 360 photos, hardware). An "ideal" class of tools would be positioned towards the far right, indicating high fidelity, yet requiring low skills and resources. Despite a wide array of digital tools for interactive prototypes, support for AR/VR content and interactive behavior is often limited and dispersed, necessitating diverse skills. (This Figure is reused from Michael Nebeling and Maximilian Speicher, "**The Trouble with Augmented Reality/Virtual Reality Authoring Tools**", 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Munich, Germany, 2018, pp. 333-337, doi: 10.1109/ISMAR-Adjunct.2018.00098. © 2018 IEEE.) 10

Figure 4.1 An example of a user journey in PONI that simulates the scenario described in Figure 4.8: a) The user first determines the type of immersive experience they want (AR or VR); b) Based on the type of experience, the user selects the category of experience (e.g., in VR, they can select simulations, 360° VR, story telling, or leave it open by indicating I don't know; c) The user then specifies their intended way for experiencing the output (e.g., for VR, the options include smartphones, tablets, Web/desktop, and HMDs) and any budget constraints or targets; d) The user specifies their background in terms of technical skills, such as programming and 3D modeling; e) Based on the answers, PONI generates a user profile and directs the user to the suggestion module which shows a ranked list of projects matching the user profile. 44

Figure 4.2 An example simplified project page that provides minimal technical terms and specifications, allowing a novice user to quickly determine the suitability of the project in light of their own background, skills, and personal goals. 50
Figure 4.3 An instance of a project card that constitutes a single result retrieved by PONI. It has three main parts: (a) a header showing the project's title, (b) a preview image from the project, and (c) color-coded bars showing to what extent each project matches the user inputs (e.g., programming, 3D modeling, category, output, budget, time), and (d) a brief project description and the target audience. 54
Figure 4.4 The results page contains customization options for users to define the impact of each user input on showing the ordered list of matching projects. The initial importance (explained in 4.4.4) of each factor is highlighted in blue and can be updated by the user. When the user changes the importance of any factor, the order of the results update accordingly to revised user preferences. 54
Figure 4.5 The filtering tool presents the choice for users to limit recommendations to projects that strictly follow the defined criteria. The user's initial inputs are shown as highlighted to support recognition over recall. 55
Figure 4.6 An overview of participants' responses to post-test questionnaire options assessing usability (ease of use), utility (ease of locating relevant materials), and user engagement (feeling of control, confidence, system demand, and reuse value). 59
Figure 4.7 The BASELINE interface consists of a search bar (a) where users issue queries using keywords. The list of results shown only match the user's search query and there is no personalization. Similar to PONI, (b) the projects are listed in separate cards indicating metadata. But, unlike PONI, BASELINE does not provide any visual cues indicating to what extent each factor matches user inputs. 60
Figure 4.8 One example of the scenarios used in the user study. 61

Figure 5.1 Schematic of the multi-layered AR development task for participants in Unity. (a) The "Scene View" where participants interacted with the provided 3D assets. This platform allowed participants to navigate and manipulate 3D objects. (b) The "Inspector Panel" which provides properties of the selected 3D object(s) in the scene view. Through this panel, participants were able to modify objects' attributes like position, rotation, scale, and attach components or scripts. (c) The "Hierarchy Panel" which lists all the objects in the current scene and provides an easy way to select, organize, and manage game objects. (d) The "Project Panel" which is essentially the file browser within Unity and shows all assets, scripts, prefabs, scenes, etc. (e) The "Visual Studio" integrated IDE often used in conjunction with Unity for scripting and code editing. (f) The "Game View" where participants could preview what they built as it would appear when running. Using this participants were able to play, pause, and step through frames for testing. 76

Figure 5.2 Timeline overview of participants and their information resource and development environment navigation. Participants usually began with Google searches to locate learning resources, often consulting official documentation, forums, and code snippets, with an average of 13.9 queries per session. Despite having access to the basics of Unity prior to starting the study, most participants favored learning by trial-and-error, frequently querying Unity interface specifics. On average, they switched 11 times between coding and resources, spending 44.5 minutes outside Unity. While initially favoring written documentation for quick lookup, 7 out of 12 participants eventually consulted ChatGPT after other resources proved unhelpful. But, only P1 and P5 found ChatGPT useful for streamlining their learning and workflow. 80

Figure 5.3 Overview of participants' responses to post-task questionnaire. (a) Participants perceived AR development as notably challenging compared to 2D tasks, with most finding it much harder; (b) Most participants found their efforts in locating online resources for 3D/AR development ineffective; (c) Most participants found it difficult to transfer their existing programming skills to 3D/AR environments. The shift to 3D added complexities in visualizing object movements, affecting not only implementation but also problem-solving approaches; (d) Over half of the participants struggled to apply their old "test-and-develop" coding habits in the new 3D/AR environment. For example, they would choose only familiar parts of the code snippets suggested by ChatGPT, but found this strategy unhelpful for debugging intricate AR interactions involving physics forces. 82

Figure 5.4 Illustration of a part of P1's dialog with ChatGPT - one of the few examples in the study that shows successful use of AI assistance. P1 initiated the conversation by requesting C# code for a moon orbiting Earth. Faced with uncertainties in the script implementation, they followed up to clarify how to attach the script to the moon object. Subsequent questions involved adding axial rotation to Earth and the Moon, resolving issues related to object interactions and parenting. Additional troubleshooting involved resolving camera position resets and discrepancies between scene and game screens. This iterative, context-rich dialog resulted in more tailored guidance from ChatGPT, enabling P1 to successfully implement the desired orbital and rotational behaviors. 84

Figure 5.5 Illustration of a part of P4's unsuccessful interaction with ChatGPT: While P4 already had a script-managed movement system, incorporating animation (recommended by ChatGPT) proved challenging. P4's vague prompt led to generic advice, causing conflicting functions to override each other. 85

Chapter 1

Introduction

1.1 Motivation

Historically, Augmented Reality and Virtual Reality development was predominantly the domain of large tech corporations and gaming companies, focused primarily on creating high-end experiences. Access to specialized hardware and software was limited and costly. However, the landscape is rapidly changing as consumer AR glasses and VR headsets become more affordable and accessible. The increased availability of these devices is not only making AR/VR technologies more mainstream but also opening doors to new consumer-oriented opportunities [202].

This shift in availability is allowing a broader spectrum of users to explore a wide range of applications beyond traditional gaming and entertainment [202]. For example, includes artists are utilizing AR to display their artworks in interactive art installations [248], educators are incorporating VR into their classrooms to convey complex concepts such as geometry [24, 114], and architects are leveraging VR to create virtual visualizations of their designs [136, 24]. Individuals with diverse skillsets and backgrounds are now tinkering with AR/VR applications, bringing unique perspectives to the field and experimenting with the versatility and expansive capabilities of these immersive tools.

Despite the increased availability of AR/VR hardware and software, creators face technical challenges in AR/VR authoring environments [25] and difficulties in designing engaging user experiences [24, 202]. Although there is a growing focus in the human-computer interaction (HCI) community on novel interaction techniques and experiences in AR/VR environments, there is a lack of understanding about how creators utilize current state-of-the-art authoring tools and the challenges they encounter. Preliminary surveys, interviews, and workshops with AR/VR creators have primarily shown isolated aspects of these authoring tools [24, 159]. A significant gap still exists in comprehending the characteristics, motivations, needs, and barriers of AR/VR creators, particularly among those creators who have little to no formal training in the relevant technologies and programming frameworks.

Understanding the practices of new AR/VR creators who lack formal training in AR/VR development is crucial for several key reasons. First, relying solely on large companies for AR/VR development risks a concentration of power and limits innovation. By enabling individuals and smaller entities to design their own AR and VR applications, we democratize these technologies. This democratization is essential for fostering a diversity of content, ensuring that the development of AR/VR technologies is not monopolized by a few large players but is instead a collaborative and inclusive field where a multitude of voices and ideas can contribute. In addition, professionals from various sectors bring their domain-specific knowledge to the table, creating tailored, task-specific applications. This cross-pollination of expertise leads to more effective problem-solving and the emergence of innovative applications that might not occur to technical developers [48, 97].

Lastly, empowering a wider range of creators enriches the AR and VR landscape with diverse perspectives, making these technologies more relevant and impactful across different industries. This is especially more important than ever with the recent advent of tools like the Apple Vision Pro, which launched with over 600 apps and is expected to reach the masses [13]. A more varied creator base means AR and VR applications can be developed that cater to a broader audience, addressing a wide array of needs and challenges. This diversity in creation could lead to more robust, versatile, and innovative AR/VR environments, ultimately enhancing the overall impact and utility of these technologies in our lives.

Lastly, the integration of AI in software development, with tools like ChatGPT, is also transforming development practices across various tasks such as data wrangling [180], web development [128], and code summarization [229]. AI-assisted tools present unique opportunities for generating foundational code and accelerating the authoring process. Such tools have demonstrated potential in aiding end-user programmers to identify suitable starting points [155], an aspect particularly noteworthy for exploration in a complex domain like AR/VR.

The primary objective of this dissertation was to understand and support the needs and challenges encountered by new AR/VR creators, particularly those without formal training in designing and implementing immersive experiences. In pursuit of this goal, I conducted research involving a wide range of AR/VR enthusiasts. This diverse group included individuals from various fields, such as UI/UX designers, linguists, psychologists, and software engineers, each bringing their unique perspectives yet all being novices in AR/VR creation. 1.1).

In my first study, I initially explored the diverse motivations of a wide range of new AR/VR creators with varying technical skills and training in programming, user research, and UX design. This exploration led to identifying key barriers they faced, including understanding the landscape of authoring tools, and the complexities of designing, prototyping, implementing, debugging, and testing AR/VR experiences. Recognizing the unique needs,

	Prior experience in AR/VR creation	Roles
Chapter 3	All (21) had tried to create an AR/VR experience in the past but still self-identified as novices	UI/UX designer, Psychologist, AR/VR entrepreneur, Design instructor, Linguist, Biomedical engineer, Cognitive scientist, Audio designer
Chapter 4	12/16 completely new to AR/VR creation 4/16 had tried to create an AR/VR experience in the past but still self-identified as novices	Software engineer, UI/UX designer, Industrial designer, User researcher
Chapter 5	No prior experience in creating AR/VR	CS researcher, Software engineer, CS student

Table 1.1: Diversity among new AR/VR creators across various phases of my research, highlighting their varied backgrounds and skillsets despite all being newcomers to the field.

goals, skills, and resources of each creator, I developed and evaluated PONI (Personalized Onboarding Interface), designed to support new AR/VR creators in the early stages of the authoring process. PONI helps users find relevant projects aligned with their skills and constraints by providing a personalized list of curated AR/VR examples. The goal was to observe how new creators adapt to the AR/VR creation process when provided with tailored resources.

These studies highlighted that new creators often gravitate towards traditional 3D game engines like Unity or Unreal for their ease in integrating 3D experiences with various tools but still faced challenges in realizing their immersive projects [197]. To understand the complexity of AR development and the specific hurdles faced by developers, I did detailed observations of developers new to AR using a simplified development environment. This approach provided deeper insights into these developers' use of online resources, self-teaching methods, and how their development strategies compared with mainstream 2D development tasks. This phase highlighted the challenges they faced in applying their existing skills to the 3D context of AR, the generality of common information sources, and the limitations of AI-assistance.

At its core, this dissertation posits the following thesis:

> "*New creators in the fields of Augmented and Virtual Reality often struggle with grasping the intricacies of the authoring process and in finding suitable learning materials. Providing them with personalized learning resources and assistance tailored and appropriated to their specific challenges can offer a useful and usable means to overcome their challenges in the initial stages of this multifaceted learning process.*"

1.2 Research Overview

Overall, this dissertation takes inspiration from the ethnographic concepts of making work visible [238] and employs a design-based [266] research approach in exploring and supporting the early experiences of creating AR/VR experiences. It particularly emphasizes understanding the initial challenges faced, the tools utilized, and strategies for personalization in the AR/VR creation process. The overarching research questions guiding this dissertation were:

1. How do new creators experience the initial process of learning about and authoring AR/VR experiences?
2. How can we better support new AR/VR creators' authoring needs?

I began by conducting an in-depth interview study with 21 AR/VR newcomers based around these high-level questions. The study revealed that despite varying motivations and backgrounds, AR/VR creators, including hobbyists, domain experts, and professional designers, commonly encounter a range of obstacles in their design, implementation, and testing processes. Eight key barriers were identified, spanning from initial difficulties in understanding the AR/VR landscape and selecting appropriate tools, to challenges in designing for immersive experiences and debugging. The study also uncovered that many creators, particularly those not part of professional design teams, lacked formal UX design approaches, leading to challenges in creating user-centric designs.

Through this initial study, I found that one of the major challenges for newcomers was identifying a clear starting point in the vast and rapidly evolving AR/VR domain. I found that newcomers often need to first explore existing AR/VR projects and understand the possibilities for design and programming which I referred to as the onboarding stage of the AR/VR creation process as it consists of preparatory activities that newcomers do before actually tinkering with any of the authoring frameworks or development environments. The array of available tools and technologies was overwhelming, and the difficulties were compounded by the lack of effective online learning resources that newcomers could easily understand and apply, given the specialized terminology and complex integration of different software and hardware components.

These specific challenges led to deriving a set of design goals and the design of "Personalized ONboarding Interface" (*PONI*), that facilitates early stage AR/VR creation for newcomers to AR/VR creation. PONI is an interactive tool that uses examples and simplified descriptions to incrementally introduce the nomenclature and stages of the AR/VR creation process. It uses a rule-based approach [37, 36] to generate a user profile that captures the user's technical skills, development or design goals, and any constraints, such as time or budget. Based on the user profile, PONI retrieves a ranked list of projects tailored to user needs and characteristics from a database of curated AR/VR examples. To accommodate user navigation and support recognition over recall [187], PONI provides visual cues to show the extent to which each user input matches the retrieved project's characteristics. Users can further customize the suggestions by defining the importance of the factors impacting the ordering of the results or by applying filters on the suggested projects. In designing and building this system, several more specific research questions arose about better supporting AR/VR new creators' learning needs:

1. How effectively does a personalized onboarding approach facilitate new creators' understanding of AR/VR development?
2. What impact does personalization have on the usability and utility of onboarding tools for new AR/VR creators?
3. How do new creators perceive the personalized suggestions and resources provided by PONI?

To address these questions, I carried out an observational usability study involving 16 newcomers to AR/VR. This study involved a comparative analysis between PONI and a non-personalized, keyword-based BASELINE interface, both of which I developed. The aim was to assess the efficacy of PONI's personalized approach against the more traditional method represented by the BASELINE, thereby gauging the impact of personalization on the usability and learning experience of AR/VR newcomers. Findings of this study indicated that the majority of participants perceived PONI as more intuitive, beneficial, and engaging than the BASELINE. Specifically, they highlighted PONI's effectiveness as a centralized resource for acquiring knowledge about AR/VR terminologies and requirements, as well as for gaining inspiration for feasible projects aligned with their skill sets. A significant benefit of PONI, as noted by the participants, was its facilitation of systematic, self-directed exploratory learning, which offered a structured alternative to the often inefficient trial-and-error approach. This aspect of PONI was particularly appreciated for its role in streamlining the learning process and making it more accessible for newcomers to AR/VR development.

While the initial approach in understanding AR/VR creation and information-seeking revealed valuable insights, it was clear that attitudinal and interview-based studies were limited in capturing the behavioral nuances of how newcomers approach AR/VR development. To address this gap, I shifted my research focus to observing individuals who were new to AR/VR but had substantial programming experience in other contexts. This focus was particularly pertinent, echoing previous research [159, 182, 183] that underscored the significance of programming skills in AR/VR creation. The decision to concentrate on individuals already proficient in programming stemmed from findings in my initial study, which identified programming as a key skill in AR/VR development. A key insight from my previous studies was that access to various development tools requiring no coding was rarely useful; for creating comprehensive experiences, AR/VR creators often resorted to tools that necessitated coding such as Unity [242] and Unreal [82]. These tools simplified the integration of their creation with various tools, Head-Mounted Displays (HMDs), and SDKs (e.g., AR-kit, ARcore, Vuforia, etc.). By engaging with participants who already possessed the necessary technical programming skills, the study could delve into understanding how such foundational knowledge impacts their learning curve and adaptation strategies in AR/VR, providing valuable insights into the behavioral aspects of this transition. Lastly, this research phase coincided with the launch of Apple Vision Pro, featuring hundreds of in-house apps, showcasing significant enthusiasm among developers for this innovative platform. This underscored the necessity of studying software developers as another emerging group of new creators of AR applications.

My work revealed significant findings about how seasoned software developers, already adept with standard help resources in mainstream software development, initiate their journey into AR/VR development. Furthermore, this research line was pivotal in exploring how AI tools like ChatGPT could potentially alter the strategies new creators use in AR/VR development. In particular, this work provided answers to the following key questions:

1. How do software developers new to AR approach the creation process and how do their learning and information-seeking experiences compare with traditional 2D development?

2. What are the specific challenges faced by new AR developers in applying their 2D development experience to the 3D realm of AR?

3. How do developers use online resources and self-teaching methods in AR development, and what is the effectiveness of these strategies compared to mainstream 2D development tasks?

To address these questions, I carried out an observational study with 12 software developers new to AR creation in the context of using a simplified development environment. This study revealed that these developers often began by seeking code examples rather than breaking down complex problems, leading to challenges in visualizing the AR experience. They encountered issues with unfamiliar vocabulary and found trial-and-error methods ineffective due to a lack of familiarity with 3D environments' physics and motion. MY study showed that that conventional code reuse strategies in mainstream development may be less effective in AR development tasks. Additionally, this study revealed that developers often struggled to form a mental model of the underlying Large Language Models (LLMs) and to express their intentions accurately. While detailed, context-rich questions yielded more useful responses from ChatGPT, challenges arose when participants lacked sufficient context or the ability to validate responses.

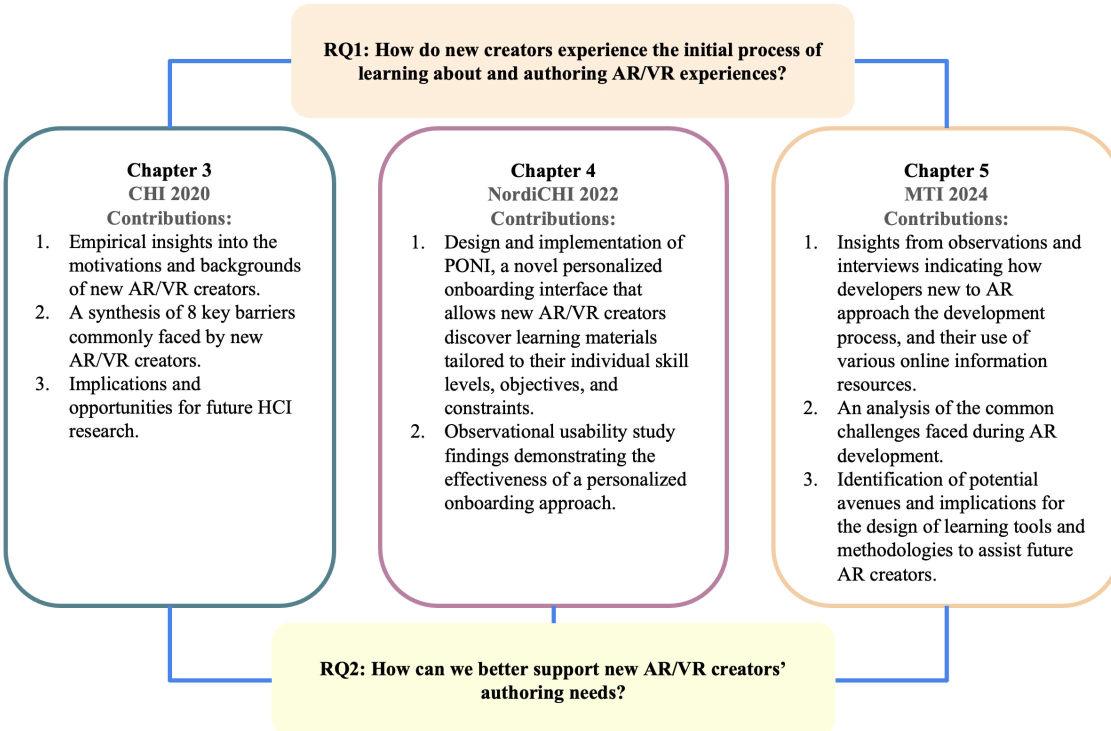


Figure 1.1: List of contributions in each chapter and main research questions addressed in this dissertation.

1.3 Contributions and dissertation structure

The main contributions presented in this dissertation are:

1. Empirical insights into the unique motivations and backgrounds of new AR/VR creators using today's authoring environments (**Chapter 3**).

2. A synthesis of eight key barriers commonly faced by new AR/VR creators, ranging from understanding the AR/VR development landscape to challenges in design exploration, prototyping, implementation, and testing (**Chapter 3**).

3. Several implications and opportunities for future HCI research, focusing on supporting end-user developers as an evolving group of AR/VR creators, integrating learning opportunities into AR/VR tools, and designing AR/VR authoring tool-chains that integrate debugging and testing (**Chapter 3**).

4. The design and implementation of PONI, a novel personalized onboarding interface that allows new AR/VR creators discover learning materials tailored to their individual skill levels, objectives, and constraints (**Chapter 4**).

5. Observational usability study findings demonstrating the effectiveness of a personalized onboarding approach in aiding newcomers in self-directed, exploratory learning within the AR/VR domain (**Chapter 4**).

6. Detailed insights from observations and interviews indicating how developers new to AR approach the development process, and their use of various online information resources and AI-assisted tools (**Chapter 5**).

7. An analysis of the common challenges faced during AR development, particularly in navigating the complexities of 3D environments, and identifying gaps in existing coping strategies to address these challenges (**Chapter 5**).

8. Identification of potential avenues and implications for the design of learning tools and methodologies to assist future AR creators, facilitating a smoother transition from mainstream software development to AR (**Chapter 5**).

The remainder of this dissertation is structured as follows:

Chapter 2 provides an outline of relevant research literature and other works forming the background for this dissertation. It integrates insights from previous studies on AR/VR authoring tools, observations of AR/VR creators, and the concept of end-user development in AR/VR. In addition it draws upon findings related to software learnability, innovations in help and tutorial systems, personalization in learning and software systems.

Chapter 3 describes my in-depth study that explores the experiences of AR/VR creators, examining their motivations, design and development practices, and the various challenges they encounter in creating AR/VR applications.

Chapter 4 introduces PONI and describes the user-centered design process that I followed to create it and evaluate its usability for new creators of AR/VR experiences.

Chapter 5 delves specifically into the journey of software developers transitioning to AR development for the first time using a simplified development environment, examining their unique strategies for acquiring AR-specific knowledge and overcoming the distinct challenges of this specialized field.

Chapter 6 and 7 reflect upon the broader takeaways for designing

Portions of this dissertation have been previously included in peer-reviewed publications (or are in submission for publication). Specifically, the material in Chapters 3 to 5 is largely derived from conference papers that I have authored. The original sources of these papers are duly cited at the beginning of each respective chapter. Additionally, select segments of Chapter 6 incorporate content from these same publications, as indicated in the relevant sections.

Chapter 2

Background and related work

In this chapter, I first position my work within the broader research literature surrounding the current state-of-the-art in authoring AR/VR experiences and draw upon insights from existing research on developing AR/VR authoring tools and observations of AR/VR creators in different contexts. Then, I discuss empirical studies on software learnability and user expertise in motivating research on the emerging domain of AR/VR authoring. Furthermore, I discuss the core ideas and innovations in software personalization and potential opportunities and challenges of using a personalized approach in the onboarding stage of AR/VR creation. Though individual chapters 3-5 each provide a more focused discussion of the literature most relevant to their respective studies, here I offer an overview of the major research areas that best contextualize my work as a whole.

2.1 State-of-the-art in authoring Augmented and/or Virtual Reality (AR/VR)

Within this section, I explore the significant developments and emerging trends in AR/VR authoring tools, highlighting the shift towards a more user-centric tool creation approach. I also examine how new AR/VR creators are increasingly contributing as end-user developers in this evolving field.

2.1.1 Current developments and innovations in AR/VR authoring tools

AR/VR experiences are designed to offer a deeply immersive experience to their intended audience. Achieving such fidelity demands that their creators possess expertise in a variety of skills, including spatial mapping, mathematics, physics, 3D modeling, interface design, and programming [184]. Unsurprisingly, a significant proportion of those responsible for crafting such experiences are seasoned professional developers with years of expertise in this field. As AR/VR devices and technologies become increasingly accessible, even amidst the inherent complexities of crafting immersive experiences, a growing number of non-technical creators,

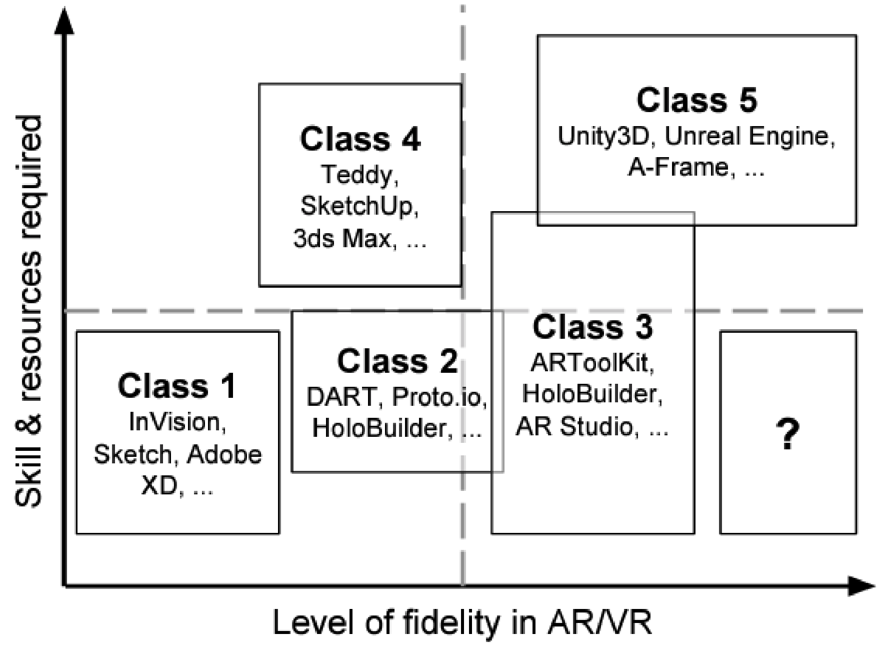


Figure 2.1: Classification of AR/VR Authoring Tools by Fidelity and Skill/Resources Requirements. Adapted from Nebeling et al. [184], this diagram categorizes AR/VR authoring tools based on (x-axis) potential fidelity in AR/VR experiences and (y-axis) required skills and resources (e.g., 3D models, 360 photos, hardware). An "ideal" class of tools would be positioned towards the far right, indicating high fidelity, yet requiring low skills and resources. Despite a wide array of digital tools for interactive prototypes, support for AR/VR content and interactive behavior is often limited and dispersed, necessitating diverse skills. (This Figure is reused from Michael Nebeling and Maximilian Speicher, "**The Trouble with Augmented Reality/Virtual Reality Authoring Tools**", 2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Munich, Germany, 2018, pp. 333-337, doi: 10.1109/ISMAR-Adjunct.2018.00098. © 2018 IEEE.)

such as artists, architects, and educators, are now exploring and experimenting with these applications [248, 114, 24, 136].

A multitude of research and commercial tools now cater to creators from diverse backgrounds and skill sets. For instance, established commercial AR/VR development platforms like Unity and Unreal offer robust support for high-fidelity prototyping, tailored interactions, and comprehensive implementation of AR and VR applications, complete with powerful physics engines to facilitate lifelike interactions. These platforms, while delivering high capability and flexibility, are feature-rich and primarily cater to experienced AR/VR creators. Packed with an abundance of features, they demand prior proficiency in programming, 3D modeling, and animation [184] (see Figure 2.1 for a detailed analysis of the current AR/VR development tools introduced in [184]).

In academia, a more user-centric approach towards AR/VR creation tools have emerged to simplify development for non-technical creators. For instance, authoring tools such as

DART [159] and HoloBuilder [233] enable creators to upload 360-degree photos for immersive scene creation and basic interactive behaviors through image maps with anchors. Another tool category focuses on AR camera interactions, including early research examples like ARToolKit [113], Tiles [205], Studierstube [114], and ComposAR [221]. Recent industry solutions like Meta's Spark AR Studio [173], Adobe Aero [5], and Snapchat's Lens Studio [109] offer libraries of trackers (face, hand, plane), 2D/3D objects, video masks, and animations, empowering users to create interactive, shareable camera effects responsive to their surroundings.

While these initiatives contribute to certain aspects of AR/VR authoring, they notably fall short in providing a foundational comprehension of the distinct categories of non-professional creators and their unique demands, and fall short in providing support for the whole process of designing, implementation, and testing AR/VR applications [184]. In Chapter 3 and 5, I will delve into a comprehensive exploration of this critical aspect, offering a more detailed analysis to bridge this knowledge gap.

2.1.2 AR/VR creators as end-user developers

In the evolving landscape of technology, the rise of user-generated content and development has marked a significant shift in how digital tools and applications are created and used. This trend, known as end-user development, has become a focal point in the HCI research. End-user development has involved a diverse group of individuals, often without formal training in computer science, who are increasingly contributing to the development of innovative solutions [130, 38]. These end-user programmers, coming from various backgrounds, initiate learning programming to enhance their professional tasks, personal projects, or leisure activities [154]. Their growing interest, the challenges they encounter in existing programming environments, and the utilization of various auxiliary learning resources have been extensively documented [60, 255, 45, 148, 162, 199].

Just like these end-user programmers, non-technical creators venturing into AR/VR share a similar profile. They are often non-formal learners of such skill with diverse backgrounds, not necessarily in computer science, who learn coding to augment tasks in their educational, work, or leisure contexts. This group mirrors the opportunistic and sometimes unplanned approach of many new AR/VR creators, a trend highlighted in several surveys [215, 27].

Therefore, insights from the research on end-user development are invaluable for supporting the emerging community of AR/VR creators. In chapter 3 and 5, I examine how new AR/VR creators as end-user programmers adapt to multidimensional authoring environments, and the strategies they employ to overcome their unique challenges. This exploration delves into the ways in which the principles and methodologies of end-user development can be tailored to better suit the specific needs of AR/VR creators.

2.2 Supporting software learnability and information-seeking in HCI

In HCI research, there is a significant focus on understanding how to improve the learnability of complex, feature-rich software applications, a challenge that also extends into the domain of AR/VR development. This research is crucial in identifying ways to "lower the barriers" of usage, making the steep learning curve associated with these advanced technologies more manageable for users.

2.2.1 Approaches to learning and seeking help

Insights from early HCI research, particularly from the 1980s and 1990s, shed light on software help-seeking behaviors. This was a time when written manuals were the norm for software assistance. Studies suggest that users often shied away from these manuals and formal documentation, favoring self-directed experimentation or trial-and-error methods instead [209, 208]. Carroll, in his work [40], coined this as the "Paradox of the Active User", based on observations from various user studies at the IBM User Interface Institute. He noted:

> "*Users seldom read manuals, choosing to directly engage with the software. Their focus is on completing immediate tasks, not on comprehensively understanding the system or spending time on initial setups or learning packages.*"

Further studies delved into why users tend to avoid consulting available help materials, and instead engage in free-form interface exploration [193, 192, 10, 194]. A key factor identified was the "*vocabulary problem*" – a disconnect between the user's mental model, the application's terminology, and the language used in help documentation [192, 78]. This issue made it challenging for users to find relevant information within help systems, which were often lengthy and complex, and used the same unfamiliar terminology as the application interface [41, 10, 192, 123].

Given these challenges, many users tend to prefer more interactive and social forms of learning. Instead of relying on official documentation, which might be dense and jargon-heavy, they often turn to online forums and community resources. These platforms can offer more direct and practical solutions to specific problems, reflecting a broader trend in adaptive learning behaviors in the face of intricate and novel technologies [141, 163, 227, 228]. This shift in help-seeking behavior is particularly pertinent in various software domains, including AR/VR, where the rapid evolution and deprecation of tools and resources make personalized, task-specific learning resources increasingly essential.

2.2.2 Innovations in help and tutorial systems

In response to learners' difficulties in finding relevant learning materials, prior work focused on enhancing the retrieval of learning materials and supporting help-seeking activities, particularly in the context of complex software environments.

One prevalent method to aid tutorial usage involves embedding in-context help within the applications themselves. This is achieved through overlays that guide users to appropriate commands or provide Q&A support, as seen in systems like Stencile [116]. Additionally, leveraging video-based learning resources has become increasingly popular for understanding feature-rich applications. Common instances include Google's "*suggested video clips* and Bing's "*smart motion preview*". Researchers have further refined this approach by implementing techniques such as timeline markers [168, 96, 126, 125, 18], thumbnail images [125, 93, 18, 157, 44], transcript text [125], and clickable elements overlaid on the videos [185]. These tools help learners navigate video tutorials more effectively and control the progression of their learning experience [203]. Other innovative strategies aim to reduce the cognitive load of learning. These include adding gamification elements [152] and augmenting tutorials with user community input [46, 251, 169], making the learning process more engaging and interactive.

Despite the general applicability of these learning and help systems to AR/VR authoring processes, new creators of AR/VR still might encounter specific challenges even at the onset of their authoring journey. They often struggle with fundamental questions like: Which tools are most suitable for my needs? How do these tools differ in terms of required skills, and which align best with my specific project requirements? These inquiries are not just general queries, but are deeply intertwined with the individual tasks and specific activities unique to each creator's objectives. Addressing these questions requires a nuanced understanding of each user's context and goals. Within chapters 3-5, I will delve into specific user needs that emerge through the AR/VR creation process and assistive approaches that new AR/VR creators found to be usable and useful given specific contexts they required help with.

2.3 Core ideas and innovations in the field of personalization

The impact of individual differences and how different users learn and experience technology has been widely acknowledged in the HCI community [59, 66, 65, 32]. Understanding user expertise and awareness of where the user is along the progression could have important implications. For instance, systems can tailor the interface to meet a user's level of experience [108, 198], information can be delivered in a personalized manner, and interfaces could provide adaptability by providing different amounts of complexity given the user expertise [223, 70, 71, 80]. In this section, I provide a comprehensive survey of the literature on software personalization and related concepts for improving the current process of new AR/VR creators in locating relevant learning resources.

2.3.1 Classifying personalizations along several axes

While there is no universally agreed-upon terminology, the literature on personalization has identified a range of distinct types of personalized interactions. Opperman and Simm [196] categorize these interactions into two primary types: (1) *interface* adaptations, involving changes to the appearance and accessibility of interface features, and (2) *functionality* adaptations, concerning how the system operates. Interface adaptations might include modifications like altering menu visibility, adjusting screen layout, or changing the interface's visual theme or color scheme. Functionality adaptations, conversely, could involve adding plugins or add-ons, creating custom macros or shortcuts, and writing scripts to alter application behavior.

Bentley and Dourish [20] distinguish between *surface* personalizations, such as choosing from predefined options, and *deep* personalization, which involve more complex changes like adding new system behaviors. This distinction underscores the varying levels of user expertise required. Haraty and McGrenere [100] define advanced personalization as modifications that extend beyond aesthetic changes to include functional alterations. Marathe [164] introduces a similar classification, differentiating between *functional* (task-based) and *cosmetic* (presentation-based) personalizations.

Bunt [34] offers another perspective, dividing personalization into graphical user interface (GUI) customizations and content customizations. Content customization specifically refers to adaptations in the delivery, presentation, and timing of content. This form of customization is especially relevant in educational technology, where content is often tailored to individual students' needs, as seen in intelligent learning systems [9, 51, 225]. In Chapter 4, I draw inspiration from the concept of content adaptations, specifically in offering varying levels of complexity of learning materials tailored and visualized to the specific needs of each newcomer of AR/VR.

2.3.2 Methods of personalization

Adaptive and adaptable interfaces

In the realm of personalized learning systems, two key approaches have been introduced as adaptable and adaptive interfaces. Adaptable systems allow users to modify system characteristics themselves, placing the adaptation initiative in their hands [196]. In contrast, adaptive (or self-adapting) systems autonomously adjust their characteristics to suit user needs, often using artificial intelligence and user-tracking mechanisms like user models [115].

Both approaches have their merits and challenges. Adaptable interfaces offer full control to users but add complexity to interactions and require effort from the users themselves, which may not always enhance their success or productivity [160, 161]. Adaptive interfaces automate the personalization process but may lack user control, predictability, and transparency, potentially impacting user performance negatively [110].

The debate on the effectiveness of these approaches continues [224, 172], with empirical evidence suggesting varying preferences in different contexts. For instance, the GRAPPLE system in educational settings has shown success with adaptive learning environments [28], while some e-commerce site users prefer self-customization over automatic personalization [195]. Comparisons of adaptive interfaces with static alternatives have yielded mixed results, indicating the context-specific nature of their benefits [61, 28].

Information retrieval systems

Personalization extends to how information is presented to users, particularly relevant in scenarios like AR/VR creators searching for tutorials or learning materials. The challenge of information overload [156] can be addressed by recommender systems, which offer personalized content based on user profiles or interaction history [37].

These systems use various algorithms categorized into collaborative filtering (CF), content-based recommendations, and hybrid approaches [156]. CF bases recommendations on the preferences of similar users [31], while content-based methods rely on past interactions and user behavior patterns [201]. Both methods have limitations, including the cold start problem and over-specialization, which hybrid systems aim to overcome by combining techniques, thus enhancing recommendations [37]. In chapter 4 in the design of PONI [14], I utilized a static, rule-based approach [216, 89] to construct user profiles reflecting technical skills, development or design goals, and constraints like time or budget. While a hybrid approach combining static and dynamic methods like collaborative filtering offers great promise for delivering suitable learning materials, it requires historical data and user behavior traces. Due to the absence of such data, I opted for a static rule-based personalization as the primary method to create user profiles.

It is also worth to mention, I primarily focus on how users interact with adaptable systems, where users are responsible for making the adaptations themselves. This contrasts with adaptive recommendation systems in information retrieval research, which automatically personalize search results based on personal factors like interests and search history [201]. While adaptive systems provide valuable insights and inform some aspects of my designs, my core focus is on the explicitly user-controllable elements of the system. This approach prioritizes user control over the implicit, algorithm-driven, and often less transparent forms of adaptive personalization.

2.3.3 Methodologies for detecting software expertise

Declaring software expertise

Effectively delivering personalized content and interactions begins with accurately determining user characteristics and needs. In this context, Grossman and Fitzmaurice [94] have identified four key methods for assessing software expertise, each with its own strengths and challenges. The first method, self-assessment, involves users ranking their own expertise level using questionnaires or interactive dialogues [167]. This method, while easy and common in usability studies, might suffer from reliability issues [190] but can mitigate the cold start problem [216]. The second method, expert assessment, involves evaluations by one or more judges, potentially offering more reliable results. The third, laboratory tasks, involves users performing controlled tasks in a lab setting, with collected metrics analyzed for expertise assessment [86, 107]. The fourth method, in-situ measurements, captures data such as usage metrics [6, 139, 153] and lower-level input activities [79, 72] during actual software usage.

Measuring software expertise

Norman's "The Design of Everyday Things" [191] conceptualizes expertise by highlighting the differences between expert and novice behaviors, noting that well-practiced skills minimize the need for conscious control. Bhavnani [21] differentiates experts from novices based on strategic knowledge, where novices rely on familiar strategies rather than optimal methods. Lafrance [138] echoes this by defining experts as having more complex knowledge. Bhavnani & John's [21] application of GOMS analysis on CAD tasks further illustrates this, identifying key differences in command combinations used by novices and experts.

On the quantitative side, various methods have been explored to measure software expertise. These include assessing low-level operations like pauses [203], menu access time [107], mouse motions [86], frequency of command usage [91], and usage heatmaps [244]. Tutorials' difficulty has also been assessed using machine learning [212] and social voting mechanisms [251]. However, these methods often focus on single-system tasks, lacking a comprehensive understanding of user skills across multiple applications.

2.3.4 Personalization in formal learning and education

The concept of personalization in education and learning, deeply rooted in educational theory [117, 118], aims to foster student-centered practices and design tailored interventions to support instructors in individualizing and appropriating learning strategies. This approach enables the accommodation of diverse learning requirements, which might be challenging to meet in traditional educational settings. By applying personalized methods in curriculum delivery, instructors can adapt to the unique pace of each student, offer customized assignments for better assessment of mastery, create learning paths tailored to individual needs, and strategically group students for optimal learning outcomes [54].

The effectiveness of personalized support is evident in its ability to guide students clearly through their learning journey [260]. Non-tailored support, on the other hand, can lead to student disengagement, resulting from tasks that are either too challenging or insufficiently stimulating. An study by Van de Pol et al. [247] demonstrated that students achieve greater academic success with high levels of personalized support than with limited personalization.

While the advantages of personalized learning in formal educational environments are well-established, extending these methodologies beyond classroom settings presents an important research challenge. In Chapter 4, my focus shifts to understanding the specific needs of new AR/VR creators initiating their initial projects as they use personalized learning materials tailored to their specific needs and skills. I explore the design implications for learning systems that provide instructional materials suited to this complex and multifaceted process.

Impacts of personalization on learning outcomes

Research evaluating personalized learning systems has consistently shown that such systems can significantly enhance the learning process. Personalizations tailored to general or specific learner characteristics are effective and efficient, leading to a range of learning outcomes from modest to substantial improvements.

A study by Conati [50] exemplifies this, demonstrating larger learning gains in physics problem-solving through the use of a self-explanation coach. The impact of personalization extends beyond academic performance to influence the user's psychological state. Marathe and Sundar [164] found that spending time personalizing a web page can significantly enhance a user's sense of control and identity. This suggests that tailoring features to the user needs can play a pivotal role in shaping self-representation and promoting self-realization.

Personalized learning platforms have also been positively linked to learners' satisfaction. Studies indicate strong correlations between student satisfaction and factors such as motivation, retention, and recruitment [217, 63, 250]. To ensure the effectiveness of personalization, it is crucial to maintain the quality and relevance of the learning database. In Chapter 4, I elaborate on how PONI addresses this challenge by incorporating expert annotation of learning resources, providing a diverse array of AR/VR projects, ensuring the stability and usability of the information retrieval system, preserving user control and flexibility, and offering transparency in the decision-making process of the recommender system.

2.4 Summary

The field of AR/VR authoring has witnessed significant advancements in recent years, largely driven by the proliferation of cutting-edge tools and technologies. However, there remains a substantial gap in our understanding of how individuals new to creating such multifaceted experiences, who often lack formal training in the requisite skills, navigate the design, implementation, and testing processes. My research endeavors to address this gap by initially delving into a comprehensive exploration of the needs, motivations, and current practices of new AR/VR creators in relation to available devices, authoring tools, and learning resources.

Building upon this enhanced understanding, I have designed the PONI platform to serve as a resource for newcomers in the AR/VR domain. PONI aims to support new AR/VR creators during their initial stages by assisting them in locating pertinent learning resources and authoring tools. It achieves this by presenting personalized recommendations based on their individual skills, goals, and project-specific requirements. Furthermore, PONI offers the capability to explore these tailored suggestions through appropriated learning materials, visual representations, and the flexibility to customize recommendations and ranking systems.

Chapter 3

Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities

In my first study [1], I conducted a series of one-on-one interviews to investigate the practices employed by newcomers in the domain of Augmented Reality (AR) and Virtual Reality (VR) creation. Given the rapid advancements witnessed in AR/VR creation platforms and devices, the primary objective of this research was to offer an up-to-date overview of the utilization patterns of today's state-of-the-art authoring tools by new creators in this field, alongside an exploration of the challenges they encounter. This study employed a broad lens, aiming to understand how individuals new to AR/VR creation navigate the technological landscape in this field, the factors influencing their tool selection and creation practices, as well as the obstacles they confront in the process of learning about and crafting such immersive experiences.

3.1 Introduction

The increased availability of AR/VR-equipped devices is opening the door to exploring a wide range of consumer-oriented applications and opportunities beyond gaming and entertainment [202]. Although interest in creating AR/VR applications is rapidly growing, creators are often dealing with a number of technical hurdles with AR/VR authoring environments [25] and struggle in designing compelling user experiences [24, 202].

[1] Portions of this chapter were originally published in Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. 2020. 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). [15]

While research on novel AR/VR tools is growing within the human-computer interaction (HCI) community, we lack insights into how AR/VR creators use today's state-of-the-art authoring tools and the types of challenges that they face. Findings from preliminary surveys, interviews, and workshops with AR/VR creators mostly shed light on isolated aspects of the proposed AR/VR authoring tools [24, 159]. We especially lack an understanding of motivations, needs, and barriers of the growing population of AR/VR creators who have little to no technical training in the relevant technologies and programming frameworks.

In this chapter, we describe a semi-structured interview study with 21 AR/VR creators that investigates how they approach design and implementation in AR/VR. To better understand the diversity in AR/VR creation practices, we recruited participants with a wide range of motivations, backgrounds, skill levels, and experiences. For example, we included hobbyists creating their own games, researchers trying to apply AR/VR for domain-specific problems, and professional designers working on enterprise-level AR/VR products. These creators worked on a variety of applications, such as in-home design, sports and rehabilitation, medicine, cultural studies, and games, among many others.

Our overall findings indicate that, although AR/VR creators vastly differ in their motivations and skillsets, they experience similar challenges in designing and building AR/VR applications. We synthesized 8 key barriers described by AR/VR creators: from understanding the initial landscape of authoring tools, to designing and prototyping AR/VR experiences, to implementation, debugging, and testing.

While almost everyone mentioned the importance of creating a good user experience, most of our participants who were not in professional design teams were not aware of any formal UX design approaches and proceeded to development environments. In contrast, most participants who were designers tried using UX principles to prototype and test AR/VR user experiences, but constantly struggled with available design methods and tools, and felt that most required "too much coding." Compared to web and mobile application development, we found that the barriers that AR/VR creators faced during implementation, debugging, and testing were more acute as creators had to deal with "too many unknowns" and keep up with the rapidly evolving AR/VR hardware and software. Overall this work makes the following contributions:

1. Producing empirical insights that portray the unique motivations and backgrounds of AR/VR creators using today's authoring environments.
2. Providing a synthesis of 8 key barriers that were common across all AR/VR creators: from understanding the landscape of AR/VR development, to exploring designs and prototypes, to implementation and testing challenges.
3. Identifying opportunities for future HCI research to support end-user developers as a growing population of AR/VR creators, to build learning opportunities into AR/VR

tools, and to design AR/VR authoring toolchains that integrate debugging and testing.

3.2 Related work

To situate our study findings in the broader HCI literature, we draw upon insights from existing research on developing AR/VR authoring tools, observations of AR/VR creators in different contexts, and how we can look at emerging AR/VR creators as end-user developers.

3.2.1 AR/VR authoring tools in research and practice

Previous research on AR/VR authoring has addressed the specific development needs of creators across a wide spectrum. For example, early tools such as ARToolKit [113] and Studierstube [114] provide a low-level framework requiring the creator to have a strong programming background. Commercial AR/VR creation tools, such as Unity, Unreal, Visor, and A-Frame, are starting to provide new capabilities that make high-fidelity prototyping easier but still require some coding. Overall, the landscape of AR/VR authoring tools appears to be fragmented [184] and creators are required to learn multiple different tools.

Some research tools have explored how early stages of AR/VR prototyping can be facilitated without the need for programming (e.g., DART [159], ProtoAR [183], 360proto [182], ARtalet [98], iaTAR [146, 145], and Amire [263]). Although these approaches have been instrumental in opening a new space of AR/VR authoring, they are designed to work on specific predefined tasks and are usually not compatible with multiple platforms, frameworks, and hardware. Furthermore, these tools do not cover the full design cycle from prototyping to development and testing on AR/VR devices.

Despite the recent research progress in authoring support for AR/VR, we know little about how AR/VR creators (particularly those who are not professional developers) are using the tools available today. Our study provides such empirical insights, detailing how AR/VR creators approach the learning process, and where they face barriers during design and development activities.

3.2.2 Observations of AR/VR creators

A number of AR/VR authoring tools that have been developed in HCI research have often included formative studies or evaluations with AR/VR creators. However, as is common with system evaluations [144], most have a limited scope to demonstrate the benefits of the tool. The closest work that aligns with our project is "DART, ten years later" [84], a study on AR prototyping that investigated the work of 8 AR creators with design backgrounds using the DART timeline-based visual AR authoring tool that reduces the need for programming [159]. Similar to results reported in that study, we found that even creators with technical backgrounds had difficulty in debugging and wanted to minimize the extra effort required to integrate more tools into their existing workflows.

Another relevant study focused on extending AR applications for cross-device collaboration [231]. The authors reported findings from a survey with 30 designers, developers, and end-users. This study asked participants to consider two scenarios of an available AR application and to discuss challenges they would expect to face when implementing them. Findings of this study provided insights on technical challenges of AR development, such as crossdevice communication, calibration, environmental mapping, obtrusiveness of authoring platforms, gesture definition, and tracking in collaborative settings.

Our findings complement these prior works by adding insights from a more diverse pool of both AR and VR creators: hobbyists, domain experts, and professional designers. We also provide new empirical insights into the AR/VR authoring process by considering the full spectrum of user research, prototyping, development, and testing, as part of AR/VR creation. In addition, we illustrate differences between AR vs. VR during different stages of authoring.

3.2.3 AR/VR creators as end-user developers

Since our study includes participants with a wide range of technical expertise and motivations, we look toward the literature on end-user development to situate our findings. End-user development has been a core topic in HCI for decades [38, 130]. End-user programmers are non-professional developers with a variety of backgrounds (often other than computer science) who learn to write code as complementary to a task they are assigned in their educational or work settings or for leisure [154]. These types of programmers are opportunistic in terms of coding and do not always have a clear plan for their development needs. A dramatic rise in the number of such end-user programmers has been reported in several studies [215, 27], shedding light upon the learning areas in which these programmers show more interest, the bottlenecks they face while working with existing programming environments [45, 60, 255], and the use of their peripheral learning resources[162, 199, 148]. Various studies have demonstrated that many end-user programmers are reluctant to learn high-level concepts, showing more interest in informal learning and trial and error [64, 208, 209].

Our work adopts a similar perspective and focuses on the emerging community of AR/VR creators, such as hobbyists and experts in domains other than AR/VR, who informally learn development in a contextualized manner. While prior work on end-user programmers' learning process shows that novice end-user programmers' strategies differ from the ones who learn programming in a formal learning setting [41, 95, 264], we have limited insights into how end-user programmers participate in multidimensional AR/VR environments. Our study reveals that beyond challenges of prototyping and selecting the most suitable development framework, AR/VR creators also have to deal with a wide range of hardware challenges, such as selecting proper headsets targeted to their project and dealing with lengthy installation procedures specific to headsets and other peripherals.

3.3 Study design

To investigate current AR/VR design and development practices, we used a qualitative approach, conducting semi-structured interviews with 21 creators who had recently worked on an AR/VR project for work or for leisure. The goal of our interviews was to better understand AR/VR creators' design, implementation, and testing approaches and to learn about any challenges that they faced along the way.

3.3.1 Participants and recruitment

To obtain a broad overview of AR/VR creation practices, we focused on recruiting participants who were new to AR/VR and excluded experienced AR/VR developers. We recruited AR/VR creators by connecting with local AR/VR meet-up groups in person and through their mailing lists. We also recruited creators by advertising posters at local educational organizations, and through personal connections and snowball sampling. We aimed for a diverse participant pool in terms of the backgrounds of the participants and the types of projects they worked on.

We ended up with 21 participants (10F/11M) who had diverse backgrounds and roles, including user experience designers, gaming enthusiasts, instructors, and academic researchers (summarized in Table. 3.3.1). They ranged from having several years of programming experience to having no technical training, working on AR/VR projects ranging from enterprise products, to games, to biomedical studies. Our participant pool covered a range of age groups: 18-24 (23%), 25-34 (52%), 35-44 (5%) and 45-54 (20%). Although most of the participants were recruited from the greater Vancouver area, we also conducted phone interviews with creators from 6 different cities in Canada and USA.

3.3.2 The interview protocol

Before conducting the interviews, we collected demographic information from the participants via a questionnaire (e.g., age, gender, occupation, previous experience in programming, education, and AR/VR creation tools they use). We started the interview by asking participants to describe their current or recent AR/VR project and to describe if there was a team involved. Next, we asked about factors that influenced their initial encounter with AR/VR, their learning process, and the resources they used for getting started. Next, we asked participants to describe their creation and tool selection process, and methods used from the design step to the final product (prototyping, to coding, to evaluation). Throughout the interviews, participants with programming and design backgrounds were asked to

ID	Tools	Occupation	Example Projects
Professional Designers (PD)			
P4	Unity	AR/VR entrepreneur	AR work safety training app
P10	Unity, Unreal	Design instructor/ UX researcher	VR-based vection research
P14	Unity	AR/VR entrepreneur	AR Work training app
P16	Unity	UX designer	AR medical education app
P17	Unity	UX designer	VR social platform
P18	A-Frame	UX designer	AR doodling app
P19	Unity	AR/VR entrepreneur	VR Wilderness training app
P21	Unity	UX designer	AR storytelling app
Domain Experts (DX)			
P1	Unity	Biomedical engineer	Research in stereo deficiencies in AR/VR
P2	Unity	Biomedical engineer	VR-based sports & rehabilitation training
P8	Unity	Cognitive scientist	Research in human lucid dreaming in VR
P12	Unity	Linguist	Research in culture revitalization (AR)
P15	Unity	Psychologist	Research in attention (VR)
P20	Unity	Audio designer	Research in sound design (AR & VR)
Hobbyists (H)			
P3	Unity	CS student	Temple run game in VR
P5	Unity	Tangibles researcher	AR Christmas card/ games
P6	A-Frame, Unity	Web developer	AR 360° city showing app/ VR flight game
P7	Argon.js, A-Frame	French language student	AR home design app/ games
P9	Unity	Info Viz researcher	VR Maze game
P11	Unity	Software developer	VR game
P13	Unity	Gaming instructor	Various AR/VR projects

Table 3.1: Summary of study participants.

describe differences in their AR/VR practices from their other types of development (e.g., mobile/ web).

3.3.3 Data analysis

To investigate the workflows, tool preferences, and challenges of AR/VR creators, we coded all of the transcripts and analyzed them using the Atlas.ti software. We used an inductive analysis approach [53] and created affinity diagrams using the gathered data to explore the themes around our main research questions. Two members of the research team first began with an open coding pass to create a list of potential codes. Through discussion and use of affinity diagrams, we arrived at a single coding scheme. During the coding process, we focused on the motivations of different groups towards AR/VR creation, and the steps involved in the AR/VR authoring process, including prototyping, development, and testing.

3.3.4 Organization and presentation of result

We present our main findings by first describing the motivations of AR/VR creators in our study and how we categorized interviewees based on differences in their backgrounds, motivations, and the types of projects they pursued. This gives context to our findings and demonstrates the variety of AR/VR projects created by our participants. Despite many differences between groups, we found that most of our participants expressed similar challenges when pursuing AR/VR projects. We synthesized these challenges as 8 key barriers in AR/VR creation (Table. 3.3.4) and explain how each manifested in the experiences shared by our participants.

8 Key Barriers in Authoring AR/VR Applications
1. ***Difficult to know where to start:*** problems related to understanding the AR/VR landscape and selecting tools.
2. ***Difficult to make use of online learning resources:*** problems related to understanding the nomenclature, formulating search queries, and finding relevant and up-to-date information.
3. ***Lack of concrete design guidelines and examples:*** problems related to knowing what is good design in AR/VR relative to good design in other types of development and lack of example projects.
4. ***Difficult to design for the physical aspect of immersive experiences:*** problems related to the design of natural VR experiences and understanding of human motion, gesture, and audio design.
5. ***Difficult to plan and simulate motion in AR:*** Problems with planning targeted experiences in AR and forecasting users' actions/movements.
6. ***Difficult to design story-driven immersive experiences:*** problems related to providing a compelling, distraction-free AR experience.
7. ***Too many unknowns in development, testing, and debugging:*** problems related to constant changes in AR/VR technology and viable debugging strategies.
8. ***User testing and evaluation challenges:*** problems related to understanding of viable testing methods, users' knowledge of tool usage and accessibility to AR/VR devices.

Table 3.2: Summary and description of eight key barriers described by participants.

3.4 AR/VR creators and their motivations

Our 21 participants had a variety of different motivations for getting involved in AR/VR creation. The participants also varied in their technical skills and formal training in user research and design (UX design). Based on these differences, we saw three groups of participants emerge from our data: 1) *professional designers* who worked on creating consumer-facing commercial AR/VR products; 2) *hobbyists* who tried out AR/VR projects as a hobby or out of curiosity; and, 3) *domain experts* who used AR/VR as a new approach to tackle a domain-specific problem.

Professional Designers: Many of our participants (8/21) were professional designers, including user experience/interface designers and design consultants. These participants were typically asked to work on a range of commercial AR/VR products as part of their job and were motivated to keep up with the evolving landscape of AR/VR technologies and evolving design practice. For example, one designer explained what motivated him to learn AR/VR design guidelines:

> *"When you put on [a] VR headset, you immediately understand that this is going to be a part of the future...from a design perspective, I realized that there's a lot of new principles and guidelines I have to learn and follow if I want to get ahead of the market." (P18-PD)*

All of the professional designers in our study had formal training in UX design and half of them (4/8) also had training in CS. However, only 3 of them had worked on AR/VR design projects during their training—the rest were all learning about design methods in AR/VR on the job:

> *"I already had experience designing mobile apps and desktop... but I didn't know anything about [design in] VR or AR.... I was really interested in getting involved and exploring those fields just to see how it works." (P16-PD)*

Although all designers believed that UX design techniques were critical to designing compelling AR/VR products, compared to other types of design, these participants felt that the UX learning curve in AR/VR was steeper and they thought it was particularly important to understand the engineering effort required in AR/VR creation:

> *"Compared to the traditional apps and games that I've worked with... you just have to be really involved [with VR]... you actually have to try to understand how your developer works, how your software engineer works, and what it actually takes to implement a certain design... or a certain behavior." (P14-PD)*

Hobbyists: Another group of our participants was comprised of hobbyists (7/21) who were not working on commercial products and mostly described gaming-related personal projects. They often stumbled upon AR/VR creation out of curiosity or simply to try out something "new and trendy":

> *"...we started googling app ideas and things like that. And, we came up with a few things that used AR and we thought that would be cool, especially after Pokémon GO was so successful... " (P7-H)*

Most of the hobbyists (6/7) had formal training in CS or engineering and felt confident in tinkering with new technologies. However, none of them had any formal training in UX design and felt that they lacked the knowledge to create intuitive interfaces:

> *"I played a lot of games before and I was pretty confident that I knew how a game should look like. But, it turned out that I had limited knowledge. So, there was this one point where I was personally satisfied, but when I showed my project to [others], everybody commented on the experience being unintuitive." (P9-H)*

Domain experts: As shown in Table 1, participants categorized as domain experts (6/21) were mostly researchers and subject matter experts in areas such as sports and rehabilitation, cognitive science, biomedical engineering, and cultural heritage preservation. These domain experts commonly saw a new approach in using AR/VR to tackle a domain-specific research problem. For example, a cognitive scientist explained how she used VR to better understand human behavior around lucid dreaming:

> *"...lucid dreaming is something you can't experience in normal life. It's really hard to train for it, and learn it...you can't watch somebody have a dream. So, this [VR] is one way to experience that same thing. I don't see really any other medium that could really give you that same experience." (P8-Dx)*

Most of the domain experts (5/6) in our study did not have any formal training in CS and did not feel confident in starting AR/VR projects from scratch. They looked for existing examples of projects online and sought methods for showing a "proof of concept." Some domain experts informally talked to end-users during their research process, but similar to the hobbyists, none of the domain experts found it easy to translate their knowledge to AR/VR:

> *"My degree was in anthropology. I worked in a community where there's no electricity and no technology. I do interviews. I know how to hold focus groups and all those things. Which is a lot of what HCI does, but I was like, "I don't know what I'm doing." (P12-Dx)*

Overall, our participants got involved in AR/VR creation for a variety of reasons and came with a range of backgrounds and skillsets. However, when they started designing and building AR/VR applications, they expressed common difficulties (Table 2), as we discuss below.

3.5 Barriers in understanding the AR/VR landscape

Our participants reported a variety of formal and informal strategies that they used to understand the AR/VR authoring landscape, highlighting three major challenges that they faced along the way: difficulty in knowing where to start, making use of online learning resources, and dealing with the lack of design guidelines and examples.

3.5.1 Difficult to know where to start

Given the pace at which the landscape of AR/VR hardware and software is evolving, a major difficulty our participants experienced was in knowing how to even get started and understanding what is "state-of-the-art." Some barrier-to-entry issues were related to knowing about and having access to the current AR/VR hardware and software versions. For hobbyists and domain experts, not understanding the hardware was particularly problematic. For example, one hobbyist participant who was keen on AR development shared his frustration:

> *"I think we should have played around a little more with it [AR.js] and seeing if it really met our needs. We got somewhere really fast and then we found out it doesn't support what we wanted in the middle of the implementation." (P7-H)*

Another issue that participants explained was that there were few relevant experiences available in AR/VR to draw upon, unlike web and mobile application development which are well established today:

> *"It was hard to get started in terms of choosing what we were going to do. No one knew what was possible. We didn't have a model list. There's no AR app for language revitalization out there... It was like, I don't even know what I'm doing." (P12-Dx)*

Our participants also indicated that they usually failed on their own to find an AR/VR authoring tool that would meet their exact needs. We learned that AR/VR creators relied on their own personal and professional contacts to get recommendations and begin the creation process:

> *"I needed to develop something that can run on different platforms and there [are] a lot of graphical things going on there... I was talking to my buddy and then he said, you can use Unity...And, then since he was using Unity, he knows all the stuff. I said why don't you just walk me through it? And that's why I chose Unity." (P5-H)*

Most participants (19/21) reported using Unity as the first and main platform they used for AR/VR creation. For example, another participant explained why his team often chose Unity even at the prototyping stage:

"Usually, not everyone on the team knows how to use them [other prototyping apps]. Teams are fairly large and most people already know how to use Unity... they just don't want to add that extra effort to learn something if they already know something else. Unity might not be perfect, but it's enough for them." (P10-PD)

Although Unity is widely used among professional developers [235], our participants reported a number of issues in getting started with it. For example, participants reported difficulty wrapping their heads around the new programming structure in Unity:

"... problem with Unity is that you have to fight to get it to work... you kind of have to learn how to make it work instead of the opposite way around... if you're creating your own system, you need to work with their rules." (P10-PD)

While there are many other authoring tools available that are easier to access and use, including many of those created in research, most participants said that they were not aware of other options.

3.5.2 Difficult to make use of online learning resources

Our participants reported using many different learning resources in their AR/VR creation process. The main resources used for learning included online search, video channels (e.g., YouTube), Unity forums, Stack Overflow, and Online MOOCs (e.g., Coursera or EDX).

Despite the growing availability of online learning resources on AR/VR creation, several participants said that these were either difficult to locate, not comprehensive enough, or became easily outdated:

"... if I look at some higher-level tutorials [on YouTube], I don't really understand it as I don't know what I should've learned before I learn this one... YouTube sometimes has some short videos that are for some specific small projects." (P2-Dx)

AR/VR creators also shared examples of their struggles in locating relevant tutorials, dealing with different platforms, and new versions of previously familiar technologies:

"A-Frame itself keeps updating. It keeps updating or upgrading its versions but the documentation is not there, and not enough for us for more advanced usage."(P18-PD)

"I used to develop for Oculus and Vive... but for the Windows Mixed Reality [framework], I had a lot of difficulty since the documentation is either not updated or without many examples." (P1-PD)

Even when a relevant resource was located, several participants (7/21) struggled with technical jargon and unfamiliar AR/VR terms. For example, if one creator wanted to know the position of a specific item [in Unity], the keyword transform would be needed to get relevant results. As an extreme case, one participant reported that he struggled with search keywords for two months due to his lack of knowledge in platform nomenclature:

> *"I always start by Google, hopefully there's some tutorial... My first big issue took me like a month or two and the problem was that I didn't understand Unity enough and the nomenclature of it to actually figure out what was going wrong [with my search]." (P4-PD)*

As a result, given the difficulties in finding useful learning resources, many of our participants relied on more informal trial-and-error methods for learning AR/VR development.

3.5.3 Lack of concrete design guidelines and examples

A key deficiency noted by participants was that compared to other mediums like mobile or web development, AR/VR development lacked concrete design guidelines and examples. This problem was particularly acute for hobbyists and domain experts with no background and little experience in UX design. For example, a hobbyist explained:

> *"We didn't have any guidelines...I mean, they say [in documentation] ... you have some assets in this Unity package, like standard buttons or standard windows. You can use them. But, they didn't say how to use them." (P11-H)*

Although professional designers' backgrounds in designing user interfaces gave them some intuition about good design, they reported that the available AR/VR guidelines were not only scattered all over the Web but also not suitable for supporting many complex scenarios:

> *"We were trying to apply some scattered guidelines from Medium, Apple or Coursera, but they were superficial... for example, when Google is designing for the Google Map, they pay attention to reminding users of "you don't want to stare at your phone all of the time." This is not something that's implemented in VR applications I've interacted with nor the design guidelines." (P17-PD)*

> *"There [has] been tons of research on how to approach designing for 2D experiences... But, for VR I think we're so early that we really require everyone's input into what makes a design good... So, depending on what your experience is in VR, you should know what makes it comfortable... what makes it not comfortable? What is good design to [users]? What is bad design to them?" (P19-PD)*

In fact, participants noted that the available guidelines (mostly through Apple and Google) still fell short when confronting the complexity and ambition expressed by many designers. At this stage, *Apple* and *Google* limit their focus to simple, single scene applications and make little allowance for complex mechanics or anything beyond simple object placement and sticker-like functionality. In particular, we found that guidelines were lacking for participants designing applications with interactive features such as object selection, conditional actions, scene flows or storyboards driven off of user behaviour, and movement between scenes using teleportation.

3.6 Barriers in designing and prototyping AR/VR experiences

The next set of barriers described by participants related to their struggles in trying to design interactive AR/VR experiences. Although some of these challenges manifested both in AR and VR, participants differentiated some struggles that were unique to either AR or VR.

3.6.1 Difficult to design for the physical aspect of immersive experiences

Some participants explained that while VR might look easier than AR in terms of maintaining the users' attention, VR experiences mostly fell short of providing natural and realistic experiences. This problem was reported due to two underlying reasons. First, as has been reported previously [232], designers described the difficulty they faced in simulating models and in providing realistic gestures:

> *"I was not satisfied with having the same idle movement for all bipedals...it wasn't realistic at all. I was like, I have to figure out a way to randomize idle states, while they're just standing around doing nothing. So, I built a randomizer [such] that it chose different actions." (P10-PD)*

According to the professional designers, while the initial user research step of design in AR/VR shared many similarities with designing 2D desktop/mobile applications, the physical aspects of designing immersive experiences were particularly difficult to address. For example, participants described how user research for AR/VR involved designing the posture of users, reducing fatigue, and eradicating simulators' sickness. In addition, our participants pointed out the importance of the cognitive aspect of how users navigate and how to maintain their attention via audio design.

> *"There's the challenge of landmarks [in VR]...for example, some experiments can't have any landmarks, so [we are] really stripping away anything that would help them [users] figure out where they're oriented." (P8-Dx)*

Some designers explained after extracting user journeys and different use case scenarios, they would follow up with a couple of brainstorming sessions to generate design ideas. As the first step of converting ideas to semi-tangible products, some designers (3/21) exploited methods such as 360° storyboarding, while others used role-playing (6/21) to illustrate immersive experiences.

One of the important differences between 360° storyboarding and flat storyboarding was having no control over the users' actions. In VR and AR experiences, users are not primed to perform specific interactions designed by the experience creator. This unlimited nature of the immersive experience is a significant departure from the 2D mobile/desktop experiences. While storyboarding was reported to be an effective method for conveying ideas, participants reported role-playing as a more effective, faster, and easier way to portray their thoughts. For example, one participant who had experience teaching VR prototyping to high school students reported the effectiveness of this method in easing the learning process for newcomers to VR:

> *"We do some storyboarding and sketching. Then we do some 'acting it out' because, paper prototyping kind of works, but it doesn't really get the feeling of what you want to do. So often we do kind of role-play of like, if you were in VR, what would you do?" (P10-PD)*

Our overall results revealed that prototyping for AR/VR was open-ended and non-representative of the real VR experience. Methods like role-playing or physical prototyping can simulate the real experience to some extent but were still not considered to be accurate in visual aspects (as has been shown in prior work [182]) and many other variables such as lighting and audio. For example, one UX designer reported the ineffectiveness of available methods in the representation of the real experience:

> *"In either AR or VR settings, the world is all around you. So the tilt, frame, or angles to show actually matter compared to 2D [prototyping]... it's going to involve multiple people... it's inevitable if we're making [mock up] videos from the objects that we create with paper, those objects are relatively small compared to our body [when showing the interactions], so the whole scene will look a little bit messy." (P17-PD)*

3.6.2 Difficult to plan and simulate motion

Another aspect of having limited control over users' actions was the difficulty designers faced when providing users with a targeted experience. Designing AR experiences can involve multiple users with different physical characteristics, different usage trends, and a variety of environments where the application may be used. Our participants reported having difficulty anticipating users' behavior and the way users hold their phones based on their different preferences in designing marker-based AR experiences:

"We have very practical usability issues...It's really awkward to hold a phone above a page... I actually programmed it to hold it perpendicular. But a lot of people go directly above." (P12-Dx)

Another participant shared difficulties in simulating multiple use case scenarios as a limitation of existing AR prototyping tools. This participant described a potential workaround, but felt that it involved a lot more coding effort than she was willing to expend during prototyping:

"... to demonstrate that kind of process [different user scenarios] we have to use a lot of animation tools to simulate that... I can make 2 to 3 simple codes to access turnarounds or the phone's orientations because I know how to code. But, that kind of thing would be more challenging for designers because if they don't know how to code, they have to simulate everything in animation tools." (P18-PD)

3.6.3 Difficult to design story-driven immersive experiences

Storytelling is a crucial aspect of creating immersive experiences [33]. In immersive experiences, end users are not just watching a story, but are actually a part of the story. While storytelling matters both in AR and VR, our participants explained some differences that they had experienced. In particular, participants who had worked on both AR and VR reported that they had an easier time authoring a compelling experience for VR applications. Compared to AR, VR lent itself more to storytelling due to the encompassing and limited nature of the experience:

"I see VR more as a storytelling medium than AR...That's not always true, but AR tends to lend itself towards shorter experiences. A lot of AR experiences are collection-based experiences. So, they're short. They don't involve much story unless there's a background story to why you should be collecting an object." (P20-Dx)

Since the story in VR is driven by the context and the environment around the user, a key challenge in VR is creating a virtual environment that tries to provide the sensations and engagement of the real world.

On the other hand, the restricted environment of VR actually reduces the distractions of the real world. In contrast, AR relies on an uncontrolled physical environment to drive the story. In fact, AR creators gave several examples of problems that they had in understanding where augmentation would affect the user experience and how to maintain users' attention while experiencing the real world around them:

"We have a lot of questions within AR; like, how do we want the user to look around and what do we want them to see while they're already experiencing the real world? How we are going to maintain their attention, and for how long before they're distracted by the real world." (P18-PD)

3.7 Barriers in implementing and testing AR/VR applications

Another set of barriers that emerged in our interviews was the nuances of implementing AR/VR experiences. In doing so, participants described various challenges in debugging and testing their applications.

3.7.1 Too many unknowns in development, testing, and debugging

Since the hardware and software needed for AR and VR development are constantly evolving, participants felt that they were always dealing with "*too many unknowns*" and had to plan ahead to anticipate and deal with problems:

"I think it really is the unknown unknowns... you just don't know until you start to program... when you start to create, these problems surface... [we have to] anticipate and plan for problems." (P20-Dx)

Compared to hardware available for 2D applications, rapid changes in hardware made things become obsolete more quickly in AR/VR industries. Persistent changes in AR/VR industries made it hard for creators to keep up-to-date and survive when the application might not be supported by the next generation of hardware to come:

"You're working in an environment where not everyone has figured out what's possible on that particular HMD. Or, you try your best to create an AR experience for the Samsung... and it doesn't work on any other Android phone. And, the client wants it on multiple phones. So, suddenly the team faces persistent changes." (P20-Dx)

Another aspect of having persistent changes in hardware was that AR/VR creators found it difficult to locate relevant technical support. In cases where most of the contributors to AR/VR technologies are start-up companies, tools can have a short lifespan and creators end up losing support:

"I own headsets that you can't get [an] SDK for any more... you spend $2,500, get on the early adopter program of something that seems to be viable and you use it for a year and then next thing you know they go bankrupt because their venture capital funding is pulled out... If they get bought up, their IP may go away and you don't have access to it anymore." (P13-H)

Participants noted several times how current development tools were not flexible in supporting diverse interactions. This sometimes forced creators to switch platforms in the middle of development as new requirements came up, which introduced even more unknowns in the creation process:

> *"When programming all these different interfaces... maybe I want to use an Apple watch that can change the visuals instead... I would have to go in and reprogram everything to include that. It would be great if there was something that's more flexible [such that] it recognizes the device and then you can just map it to whatever..." (P8-Dx)*

> *"One of the main problems is that AR change is very fast...the technology, the SDKs, the platform, the library that you use to create changes very often...I had to work with 3 different libraries, just because every time I worked in a library, it got canceled and I had to switch to a different one." (P10-PD)*

The issue of dealing with unknowns made it especially difficult to debug AR/VR applications. Participants identified many variables, including the dimension of motion and the complex structure of programming with Unity, as posing many difficulties in the debugging process. For example, one domain expert explained the difficulties she faced in systematically finding the location of errors:

> *"I don't like the debugging experience in Unity... sometimes the bug comes from Unity... like if I didn't attach some piece of code to objects in Unity. Sometimes the bug actually is in the code itself. So, the debugging becomes confusing. "(P2-Dx)*

In another example, one professional AR/VR creator described the physical aspects of the debugging process that remain neglected in online tutorials:

> *"It's good to see the person doing what they say they're doing physically. Maybe all the code is correct but what you're doing with your body in VR is incorrect. And usually people don't write about that aspect." (P10-PD)*

An important part of debugging AR/VR experiences involves checking the application behaviour by testing and inspecting the interactions visually. Our participants reported problems in referencing bugs that manifested visually but were hard to pinpoint in code, expressing a lack of efficient ways to control multiple, often concurrent, events without losing track:

> *"How am I going to make 400X number of targets? Also, every single target corresponds to a different audio clip: how am I going to keep a visual reference*

to what that audio clip is? What happens if my files get mixed up? Essentially, I had to create a way of keeping track of what was going on and then figure out a way that I could debug these targets." (P12-Dx)

In both AR and VR implementations, participants explained how locating the originating bug can be a difficult task. For example, the environment the application is being tested and the lighting can affect the object tracking process. Moreover, in marker-less AR with new tools like ARCore or ARKit decent knowledge in programming is usually required:

"It's just like it either works for me or it doesn't work and then there's no way to fix it my background does not involve any sort of computer vision (CV) and stuff. I believe there's this part in AR [that] is CV and tracking or recognition... I have no idea how those work. So those are like a black box [for me]." (P5-H)

3.7.2 User testing and evaluation challenges

As described in the barriers above, AR/VR technologies are "bleeding edge" at this stage and most of the effort is expended on getting things to work. The sheer number of barriers we identified implies that creators are busy dealing with many other issues, leaving little time for formal user testing or evaluation.

When there was interest in doing user testing, most AR/VR creators did not know how to do it properly. In particular, hobbyists and domain experts explained that they were not familiar with any usability evaluation methods, even if they wanted to improve the user experience of their applications:

"I pulled up old Xerox documents on user testing and pulled up their articles, and read about what they do. I picked up some books in the library and was like, "I need to learn how to do user testing. Let's read up on user testing, and how to do this." (P12-Dx)

Even for the professional designers who were invested in user-centered design and evaluation, there were major challenges in translating the UCD guidelines to AR/VR. They often attempted to test their applications with UX methods they had learned, but ultimately most participants in this group felt that their approaches fell short. Since most end users are still unfamiliar with AR/VR technologies, participants explained how there can be a long onboarding process for them. In addition, for many types of users, their lack of familiarity with the AR/VR technologies introduced unanticipated variables that affect the output of the experience:

"The moment it [VR headset] is placed on a user's head, it's one of the biggest challenges... especially if it's a new user, you're suddenly asking them to be blind and reach out and find their controllers... they see a virtual representation of it,

so they have problems to grasp that connection in their minds that what they touch is the equivalent of what they're seeing virtually." (P20-Dx)

As mentioned in the prototyping section, a key challenge for authoring AR applications was designing a compelling experience with minimum distractions. While the points of distraction are expected to be gleaned from user testing, a challenge resulting from low control over experiment variables was the lack of ability to pinpoint the specific sources of distraction.

Another point of difficulty in conducting user testing was the hardware used by both test participants and developers. The constant transition between the virtual world and the debugging console caused nausea and fatigue among AR/VR creators, often leading to either prematurely releasing an application or engaging in a long iterative testing process:

"... in almost every way it's more difficult [in VR]... you can't look at what you're experiencing in VR, and then also look at what's happening on the screen on the Unity window. And also, you have controllers, it's a two-handed experience and so you can't use your keyboard and mouse at the same time as well." (P18-PD)

From the perspective of users testing a VR application, the heaviness and warmness inside the HMDs posed additional difficulties. In some cases, VR controllers were not deemed to be representative of interactions in the real world and were confusing for users, as shown in other research [171]. With a longer onboarding process to help users pick up the new methods of interaction, the actual testing sessions tended to be time-consuming not always insightful for creators.

Table 3.3: Summary of different AR/VR creation approaches and key activities among different groups of creators

	Understanding the landscape	**Designing and prototyping**	**Implementing and testing**
Professional Designers (PD)	attended local meetups (5/8), asked technical colleagues (3/8), asked questions in internal Slack (2/8)	used their prior experience/ resources in 2D design (6/8)	used their prior experience in testing 2D apps (8/8), took formal courses in testing and implementation online and in person (4/8)
Domain Experts (Dx)	sought inspiration via online search (5/6), asked social contacts (4/6)	skipped this phase (3/6), mimicked similar online projects (3/6)	followed implementation-focused online tutorials and patched together code examples, but had trouble with debugging (6/6), skipped usability testing (5/6), failed to implement the project (1/6)
Hobbyists (H)	inspired by seeing interesting online videos/posts (6/7), heard from or asked social contacts (3/7)	skipped this phase (4/7), some ideation by sketching code on paper (3/7)	followed implementation-focused online tutorials and had functional apps (7/7), skipped any form of usability testing (5/7), performed QA testing (2/7)

3.8 Discussion

Our findings overall illustrate the current state of practice of AR/VR creation in our relatively diverse group of participants in terms of how they design, implement, and test AR/VR applications (summarized in Table. 3.7.2). In particular, we have highlighted 8 design and implementation barriers (Table. 3.3.1) that were common between our participant groups. We now reflect on the implications of our findings for future research in HCI. In particular, we discuss the importance of considering end user developers as a growing population of AR/VR creators, how we can build learning opportunities into AR/VR tools, and the need for building AR/VR toolchains that integrate debugging and testing.

3.8.1 Important to consider needs of AR/VR end-user developers

A lot of the current hype for AR/VR is among professional developers who can usually access cutting-edge tools on-the job. But, as illustrated in our findings, hobbyists, domain experts, and designers can have different needs for prototyping, programming, debugging, and testing AR/VR applications. Given that there is already a lot of momentum in HCI to better understand and support end-user developers [38, 130], we consider our study to be a starting point for looking at modern AR/VR development through this lens.

Most notably, we found that domain experts and hobbyists may not even know where to start and rely on ad-hoc social recommendations to select their authoring environments. This can result in choosing a tool that, while fitting their project need, may not fit their level of experience, and even if there are no major issues in the design phase, the issues tend to be aggravated during implementation and testing.

The frequent AR/VR hardware and software updates can make end-user developers feel especially left behind and struggle to keep up. One of our participants put it as:

> *"The industry [is] trying to solve the problem to get as many headsets in consumer's hands as possible... but at the same time, they're leaving the developers behind." (P13-H).*

Several of our participants expressed a similar level of frustration and considered giving up because of the dramatic hardware or software changes they experienced and the lack of relevant expertise that they had in getting back on track. This is an important finding for future tool developers, where it would be worthwhile to consider techniques such as progressive enhancement from web development (also suggested in [231]), to help users manage these transitions.

3.8.2 Building learning opportunities into AR/VR tools

Our results show that AR/VR creators used two main classes of authoring tools. The most prominent category consisted of professional, feature-rich frameworks, such as Unity, which was originally designed as a game engine and only recently grew into a popular platform for AR/VR. Since these tools are more established, there is often a larger community of AR/VR creators to provide support and examples for learning [154, 177]. However, the large feature set poses issues with tool explorability and has a steep learning curve. The second class of tools was more targeted at AR/VR development, but consisted of tools created in start-ups (e.g., Torch), or tools developed in research (e.g., Argon.js). Our participants found these tools were often less refined and had a relatively smaller user community, with fewer accompanying examples and more limited support.

In light of the authoring-related issues described by AR/VR creators, we discuss potential avenues for HCI research.

Supporting early-to-middle-stage AR/VR prototyping

Some current work is already exploring methods for lowering the barrier to entry in AR/VR development. For example, Torch tries to provide a code-free experience for designers such that they can quickly prototype their ideas. However, such tools may, in fact, be too high-level and abstract away all the design and development challenges. This can lock creators into the tool and make it hard to transition to more powerful platforms such as Unity, which they will ultimately need when going beyond the prototyping stage. One approach could be to adapt the principles from emerging prototyping tools, such as ProtoAR [183] that use Play-Doh props as 3D model stand-ins or 360proto [182] for new paper prototyping templates, and integrate them with advanced tools like Unity as a way of supporting early-to middle-stage prototyping even in developer tools.

Personalizing AR/VR authoring tools based on expertise

Our hope for future authoring tools is that they can find a better match between expressivity and learnability—end user developers in AR/VR can benefit from starting with a simple development environment but with the opportunity to learn the more advanced concepts directly inside the tool. One way to do this could be to draw upon the adaptive interfaces literature to tailor feature-rich interfaces of complex authoring environments according to users' expertise level [34, 71]. Another direction could be to explore ways of making AR/VR authoring tools more collaborative such that novice creators could express ideas and explore interactions while more experienced developers could take the ideas through to implementation [92, 166]. This could also be extended to use online and on-demand developer communities [43, 88].

Integrating access to learning resources within implementation workflows

We identified several learning barriers experienced by AR/VR creators: lack of understanding and background knowledge in nomenclature, problems finding relevant tutorials, and figuring out what basic knowledge is important before jumpstarting an AR/VR creation task. Just like with the problem of constantly evolving tools, the updating rate for the tutorials and contents does not map with the update rate of the technology. This means that tutorials quickly become outdated and put the creation process at stake. Future work can draw upon learnability research for feature-rich software [95, 123] to better understand and support the learnability of AR/VR authoring tools. An interesting challenge here would be the interplay between hardware and software and design of help for immersive experiences.

3.8.3 Building AR/VR toolchains with integrated debugging and testing facilities

A recent review of the AR/VR tool landscape [184] shows that there is a rapidly growing number of authoring tools, but only a few transition points between them. Our interviews confirmed this, highlighting many difficulties when designing for the physical aspect of immersive experiences and the need to plan for and react to users' motions. The need to constantly transition between a VR headset and the console made it especially difficult to debug and properly test applications.

This opens up the design space for new AR/VR tools where debugging and testing facilities could be an integral part of the authoring experience. Although it would be difficult and not even desirable to build a tool that fits all needs, it is worth exploring how to design transition points into authoring tools. For example, this could mean that AR/VR creators could move from a transition point focused on prototyping, to different ones focused on implementation and debugging, to again different ones focused on testing. Future work could also explore more interactive debugging tools like the *WhyLine* [132] and investigate how they can be extended in these virtual environments to help people locate bugs and discern why their applications are not behaving as intended.

A lot of promising work in HCI is already considering testing and evaluation issues for AR/VR. For example, Dey et. al's comprehensive review of ten years of AR usability [57] reported 369 AR user studies. However, we found that most AR/VR creators, even in professional design teams, are not using these "more research-style" approaches. It may be worth thinking about what could be the parallel "discount usability" [188] methods for testing AR/VR applications that can help practitioners. A starting point would be to revisit and reconcile heuristics [68, 176, 240] proposed in prior research for evaluating specific AR and VR applications.

Lastly, even when creators had user tests set up, they often struggled to get experienced AR/VR test participants. Although some participants had experience with a certain AR/VR headset, that experience did not always transfer to a different device. It would be worth exploring emulator designs that can help with parallel testing and level the playing field in AR/VR creation.

3.8.4 Limitations

One limitation of our study is that it presents perspectives of AR/VR creators from North America only. Given the qualitative characteristic of our study, there should be some caution used in generalizing the findings. Future research can complement the insights from this study with large-scale surveys or other approaches that include more geographically diverse groups of AR/VR creators.

3.9 Conclusions

In our study, we gathered insights from 21 AR/VR creators with diverse backgrounds using current authoring environments. Among the eight barriers we identified, a critical issue was helping newcomers find a clear starting point in the expansive and rapidly evolving AR/VR domain. New creators often need to first familiarize themselves with existing AR/VR projects and the possibilities for design and programming as part of their preparatory activities, prior to engaging with authoring environments. The overwhelming variety of tools and technologies, coupled with the scarcity of individualized learning resources that can be easily understood and applied, exacerbates the challenges faced by these creators.

In the subsequent chapter, I introduce PONI, a platform designed to mitigate these issues by helping users locate relevant and personalized learning resources and project examples for inspiration.

Our long-term vision is to broaden participation in AR/VR authoring so that end-user developers can solve domain-specific problems and create more compelling and meaningful user experiences.

Chapter 4

Using a Personalized Onboarding Approach for Getting Inspiration and Learning About AR/VR Creation

To support the initial learning needs of new AR/VR creators from different backgrounds, I designed and implemented a novel Personalized Onboarding Interface (PONI) [1] that allows users to locate relevant projects based on their programming and 3D modeling skills, development goals, and any constraints, such as time or budget. In this study, I adopted an experimental lens and examined how AR/VR newcomers make use of the personalized onboarding approach in their initial stage of learning about AR/VR creation. In addition, I discussed ways in which the personalization could be further enhanced and how the potential of PONI could be explored to improve onboarding in contexts beyond AR/VR development.

4.1 Introduction

As consumer-level augmented reality (AR) and virtual reality (VR) devices are getting cheaper and easier to access around the world, there has been growing interest in creating new types of AR/VR applications. This growth has led to a proliferation of different AR/VR authoring tools and development environments. For example, commercial frameworks such as *Unity*, *Unreal*, and *A-Frame* allow developers to create industry-level games and other types of creative AR/VR experiences. There is also another class of emerging tools with simpler user interfaces that aim to lower the barrier-to-entry for AR/VR development (e.g., Vizor.io [3], CenarioVR [1], and Cospaces.io [2]). Research in HCI is also pushing the bound-

[1] Narges Ashtari, Parsa Alamzadeh, Gayatri Ganapathy, and Parmit Chilana. 2022. PONI: A Personalized Onboarding Interface for Getting Inspiration and Learning About AR/VR Creation. In Nordic Human-Computer Interaction Conference (NordiCHI '22). [14]

ary of AR/VR prototyping by exploring tools that eliminate the need for programming or 3D modeling to make AR/VR creation easier to access (e.g., [181, 159, 183, 182, 232, 146]).

Figure 4.1: An example of a user journey in PONI that simulates the scenario described in Figure 4.8: a) The user first determines the type of immersive experience they want (AR or VR); b) Based on the type of experience, the user selects the category of experience (e.g., in VR, they can select simulations, 360° VR, story telling, or leave it open by indicating I don't know; c) The user then specifies their intended way for experiencing the output (e.g., for VR, the options include smartphones, tablets, Web/desktop, and HMDs) and any budget constraints or targets; d) The user specifies their background in terms of technical skills, such as programming and 3D modeling; e) Based on the answers, PONI generates a user profile and directs the user to the suggestion module which shows a ranked list of projects matching the user profile.

In chapter 3 we demonstrated that, despite the availability of several options for authoring AR/VR applications, getting started with AR/VR development still presents a steep learning curve for newcomers to the field [15]. Newcomers often need to first explore existing AR/VR projects and understand the possibilities for design and programming. We refer to this as the *onboarding stage* of the AR/VR creation process as it consists of preparatory activities that newcomers do before actually tinkering with any of the authoring frameworks or development environments. This onboarding stage can be particularly problematic for the growing community of AR/VR creators who come from a range of different domains and may have limited or no professional training in software development, design or engineering [15]. These creators can include artists who are exploring AR to showcase their creations in art installations [248], teachers who are tinkering with VR in their classes to convey complex ideas like geometry [24, 114], and architects who are trying to create virtual physicalizations of their designs [136, 24]. These diverse non-professional AR/VR creators are often not familiar with relevant terminology and concepts and find it difficult to understand the full landscape of AR/VR techniques [15].

To locate examples and learning materials, most new AR/VR creators currently start their informal onboarding process by initiating a web search and peruse through resources such as MOOCs, YouTube videos, and online forums [15]. However, since these creators are not familiar with the AR/VR nomenclature and do not fully understand the interplay between different hardware and software components, they have difficulty in formulating their queries and expressing their desired goals (e.g, should they be choosing *marker-based* or *marker-less* AR?). Another key problem for these creators is assessing the suitability and reliability of the retrieved materials relative to their own skills in programming, 3D modeling, or other technologies as demonstrated in my initial study [15]. For example, a newcomer may not realize that the tutorial that they are looking at requires advanced knowledge of 3D geometry or skills in adapting a particular API. Furthermore, creators may be working within the constraints of a timeline or a specific budget and the examples or tutorials that they find online may not be possible to recreate within these constraints. As a result, in chapter 3 we saw that creators can get entangled in inefficient trial-and-error processes as they look for relevant examples and guidance.

Given the difficulties that new AR/VR creators face in understanding the landscape of different design possibilities and determining the suitability of tutorials for their own needs, we wondered how we could use a *personalized* approach to support these creators' onboarding process. In particular, our research question is: *How can we design a **personalized onboarding** tool for helping new AR/VR creators retrieve learning materials that are appropriate for the creators' level of technical skills, desired goals, and a given set of constraints?*

In this paper, we present the design and evaluation of a "**P**ersonalized **ON**boarding **I**nterface" (*PONI*), that facilitates early stage AR/VR creation for newcomers. PONI is an interactive tool that uses examples and simplified descriptions to incrementally introduce the nomenclature and stages of the AR/VR creation process. It uses a rule-based approach [216, 89] to generate a user profile that captures the user's technical skills, development or design goals, and any constraints, such as time or budget. Based on the user profile, PONI retrieves a ranked list of projects tailored to user needs and characteristics from a database of curated AR/VR examples. To accommodate user navigation and support recognition over recall [187], PONI provides visual cues to show the extent to which each user input matches the retrieved project's characteristics. Users can further customize the suggestions by defining the importance of the factors impacting the ordering of the results or by applying filters on the suggested projects.

To evaluate PONI, we ran an observational usability study with 16 AR/VR newcomers and compared PONI with another non-personalized keyword-based BASELINE interface. We found that almost all of the participants found PONI to be more intuitive, useful, and engaging compared to the BASELINE. In particular, participants indicated that PONI served as a useful centralized hub for learning about AR/VR terminologies and requirements and to get inspiration for potential projects that were actually feasible given one's skill sets. A key advantage of PONI for participants was that they could engage more in systematic self-directed exploratory learning instead of relying on trial-and-error.

The main contributions of this paper are: (1) the design and implementation of PONI, a novel personalized onboarding interface that allows new AR/VR creators to discover learning materials tailored to their skill levels, goals, and constraints; and, (2) insights from an observational usability study that demonstrate the utility of the personalized onboarding approach for newcomers and how it could be used for self-directed exploratory learning. Our findings confirm that one-size-fits-all approaches do not work well for the differing needs of AR/VR creators and that personalization techniques could provide a fruitful starting point for supporting nuances in onboarding. Overall, our work highlights the importance of adopting a user-centred interaction design perspective for designing personalized systems in the context of supporting informal learning.

4.2 Related work

This work builds upon prior research on challenges in getting started with AR/VR creation, innovations in software learnability, and personalization approaches used in formal learning.

4.2.1 Challenges in creating AR/VR applications

New AR/VR creators can face a number of different challenges in getting started as they need to understand the capabilities of various platforms, tools, and devices and determine how they work together to create a cohesive AR/VR experience. Currently available creation frameworks can vary widely in terms of system structure and hardware constraints to support intended use-cases and relevant features [231, 184]. In most cases, experiences created with one framework only run on one kind of device, and repurposing it to another framework or adjusting it to support more devices is either complicated or expensive. For example, marker-based applications are created entirely differently than ones that work with spatial mapping. In addition, in chapter 3, we saw that while a newcomer may start with a quick web search, identifying an appropriate tool-chain requires experience and domain knowledge [15]. This even makes it harder for newcomers to understand the strengths and weaknesses of different frameworks, understand different creation processes, and figure out ways to best combine available resources to satisfy application requirements [184].

One approach for lowering the barriers to entry is automating some of the technical aspects of AR/VR creation, such as generating the initial prototypes [214, 182, 119, 22, 145, 143]. However, in chapter 3 we saw that such approaches may not work for AR/VR creators who have little to no software development experience and lack a conceptual model of the overall creation process. We also learned that new AR/VR creators, such as hobbyists and domain experts, face difficulty in knowing where to even start, lack access to concrete design guidelines and examples, and struggle in making use of online learning resources. Other research on AR/VR creation [23, 231] has also identified the struggles that creators face with the fragmented landscape of AR/VR authoring tools and how newcomers often fail to select appropriate programming languages, authoring tools, or testing hardware that meets their project-specific needs. Lastly, in chapter 3 we revealed that most newcomers try to draw inspiration from existing example AR/VR projects and use them to jump-start their design, but struggle in finding learning resources that contain an appropriate amount of high-level (e.g., general rules and strategies of the AR/VR creation process) and low-level details (e.g., software, hardware, and devices used for a particular AR/VR experience) [15, 184]. This makes it difficult to determine the feasibility of a given project that matches an individual's needs and constraints.

Considering the challenges that new AR/VR creators face during onboarding, in this research we attempt to lower the barriers of entry through the design of an interface that 1) personalizes newcomers' initial learning experiences considering their background and constraints, 2) helps them gain domain knowledge through exploration of example projects, and 3) helps them assess the suitability of learning materials.

4.2.2 Innovations in software help and tutorial systems

Although consumer-level AR/VR creation has only recently started receiving attention in HCI, there is a long history of research on software learnability and supporting help-seeking activities. Since beginners are known to struggle in locating relevant learning materials [123], some research advocates embedding the relevant help in the form of tutorials and Q&A within the target application through overlays and other in-context techniques [74, 46, 251, 168, 169]. Researchers have also explored techniques for improving interaction with video-based tutorials (e.g., [116, 134, 75, 125, 203, 200, 18]), which tend to be more popular way of learning about using a feature-rich application [123]. Other approaches have tried to lower the learning cognitive load by adding gamification elements [152] or augmenting tutorials with input from the user community [204, 140, 35]. Although the general concepts in this software learning and help systems can be applied to specific AR/VR authoring tools, prior work shows that newcomers face challenges in even knowing what tools to select in the first place and their first onboarding need is understanding the overall landscape of AR/VR development.

Individual differences in training and technical expertise also play a significant role when a user starts learning a new programming language or works with a new feature-rich application [65, 59]. Some works have explored ways of detecting software expertise [94] to support users coming from different backgrounds and differing in skill levels, often using low-level operations such as pauses or dwells [203], time of access to the menu [107], mouse motions [86], and usage heatmaps [244]. While these approaches can be effective after a considerable amount of user interaction with the system, they suffer from cold start problem and cannot provide much advantages to the user without exposure to actual user profiles or activities. Another class of tools has explored ways of assessing a tutorial's difficulty by using machine learning techniques to automatically assess a tutorial's difficulty (e.g., [212]) or by using social voting mechanisms to classify difficulty level of instructions [251]. But, these approaches only consider one type of application-specific expertise and, in practice, most software activities span multiple applications. To accomplish this type of activity learners need to equip themselves with a "tool-belt" [239] often differing in characteristics, commands, and output, but we are only starting to see some work in HCI exploring application-independent learning support (e.g, *RePlay* [76]).

In summary, most existing works only provide targeted help for specific tasks that are performed within a single system without acquiring a deep understanding of the user needs, characteristics, and target project. In contrast to the existing approaches in learning and help-seeking, PONI presents a novel design that focuses on personalized onboarding. PONI applies a rule-based [101] method to provide an opportunity for newcomers to declare their own backgrounds and intents. Users can see an overview of the chain of tools used in the creation process of various AR/VR projects activities that is personalized for their needs and level of experience in programming and 3D modeling. Furthermore, PONI personalizes tool suggestions based on user's access to devices and budget constraints.

4.2.3 Personalization in formal learning

Personalization in education and learning has a long history [117, 118, 26]. The goal of personalization in formal learning is to adopt student-centered practices and design intervening mechanisms to help instructors better individualize learning strategies. They take into account differences in students' skill levels, needs, and interests, and assist learners to *succeed* at a task [170, 206]. By drawing on this method during the delivery of the curriculum, instructors can allow students to move at more individualized paces, assign customized assignments to assess each student's mastery, devise a path that is customized to address each student's needs at the moment, and cluster students strategically [54]. When personalized support is provided [260], the student knows which steps to take and how to proceed independently. When support is non-tailored to students' understanding, students often withdraw from the task as it is beyond or beneath their reach causing frustration or boredom. One large-scale study [247] of a personalized learning classroom intervention program showed that students' achievement (measured with a multiple-choice test and a knowledge assignment) increased with high levels of personalized support compared to limited levels of personalizations in student advising. But, how these interventions could be designed beyond a classroom or formal learning setting is an open research question.

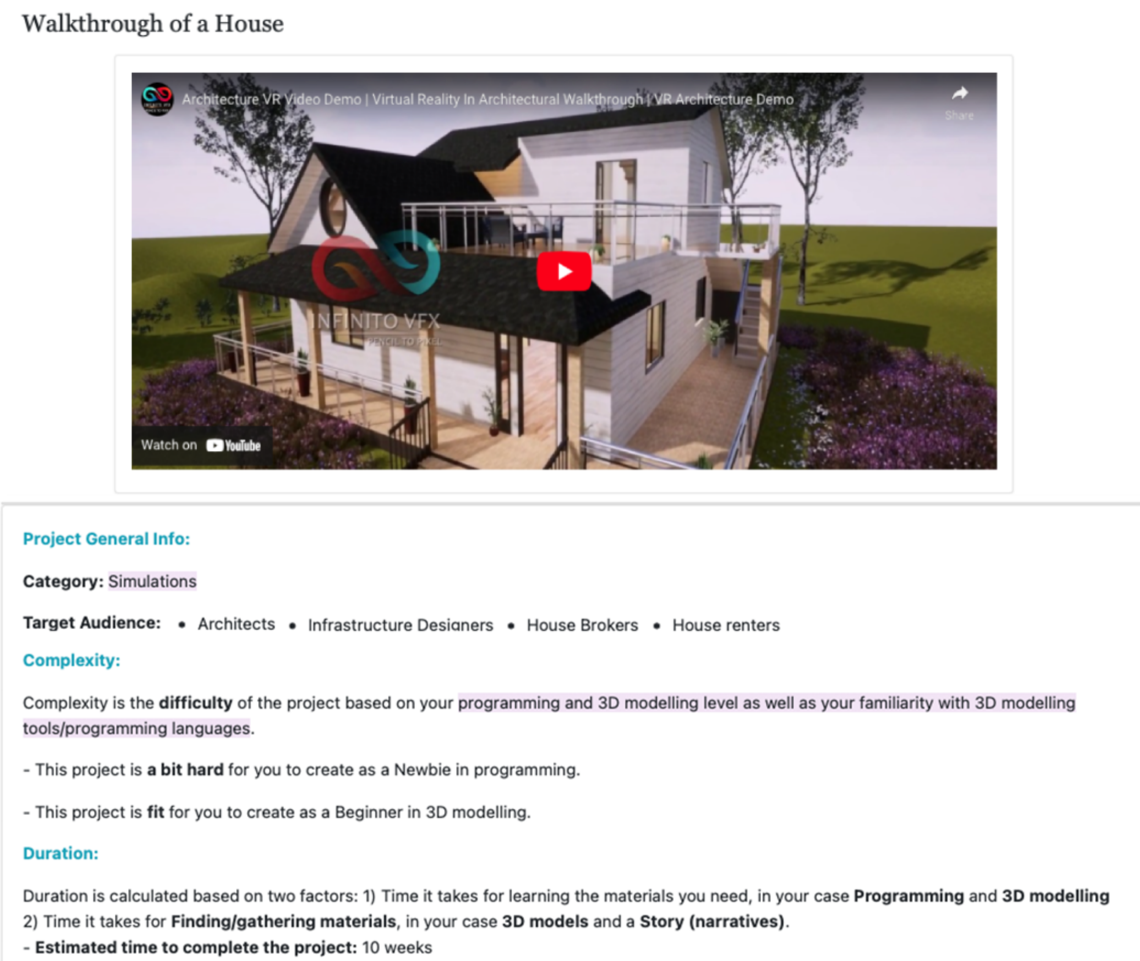
Our work takes inspiration from personalized learning approaches in classroom settings and explores personalization for informal learning practices, such as looking up technical tutorials and examples. Personalization requires information about the user, whether the data are explicitly gathered by asking people to fill out forms (e.g., rule-based personalization [101]) or implicitly through analysis of behavioral data (e.g., data-driven personalization [36, 85] or collaborative filtering (e.g., [219]). The latter techniques require historical data and digital traces of user behavior to tailor the learning materials to user needs. In designing PONI, since we did not have prior access to user profiles or digital traces related to onboarding, we could not use data-driven and collaborative filtering approaches for recommending relevant content. We instead used a rule-based personalization technique to build user profiles that take into account a user's background, skills, and constraints to offer them tailored onboarding content.

4.3 Design considerations and goals

In this paper, we explore the design of a personalized onboarding interface that helps newcomers get familiar with the landscape of AR/VR technologies and terminologies, and allows them to retrieve learning materials relevant to their interests, skills, and constraints. Majorly based on my initial work in chapter 3 as well as the rest of the related work discussed above, we considered different aspects of designing and structuring the personalized onboarding process for new AR/VR creators and derived five design goals:

DG1: Locate targeted learning materials given the creator's background and skills. Prior work in learning research shows that people build new knowledge based on what they already know and believe [124, 69, 254] and personal variables (motivations, goals, and self-efficacy) may be predictors of engagement in a development activity. Onboarding approaches for AR/VR creation should take into account learners' prior knowledge (e.g., programming and 3D modelling skills) and should be flexible enough to adapt to learner differences [15, 184].

DG2: Locate targeted learning materials given the desired creation outcomes and constraints. Creating AR/VR experiences requires working knowledge across a chain of software and hardware tools. For example, designing a VR 360° experience for YouTube could consist of an initial 3D prototype in Blender with mock ups and animations added in Unity, and use of the YouTube video player for testing, or optimization for a more



Walkthrough of a House

Project General Info:

Category: Simulations

Target Audience: • Architects • Infrastructure Designers • House Brokers • House renters

Complexity:

Complexity is the **difficulty** of the project based on your programming and 3D modelling level as well as your familiarity with 3D modelling tools/programming languages.

- This project is **a bit hard** for you to create as a Newbie in programming.

- This project is **fit** for you to create as a Beginner in 3D modelling.

Duration:

Duration is calculated based on two factors: 1) Time it takes for learning the materials you need, in your case **Programming** and **3D modelling** 2) Time it takes for **Finding/gathering materials**, in your case **3D models** and a **Story (narratives)**.
- **Estimated time to complete the project:** 10 weeks

Figure 4.2: An example simplified project page that provides minimal technical terms and specifications, allowing a novice user to quickly determine the suitability of the project in light of their own background, skills, and personal goals.

immersive experience in an Oculus headset. However, newcomers often fail to find examples and tutorials that cover this entire process [15, 184]. Moreover, newcomers usually have different constraints based on their allocated budget and time commitment [15]. Onboarding for AR/VR creators should allow creators to locate a personalized set of learning materials that are appropriate and feasible for their desired creation outcomes and constraints.

DG3: Get inspiration and browse relevant example projects. Although a plethora of learning resources are available online (MOOCs, YouTube videos, tutorials, forums), they are laden with device-specific or application-specific instructions [15, 184]. However, newcomers often do not even know where to begin as they not understand the landscape of possibilities and are less familiar with the vocabulary used in tutorials [15, 78]. Onboarding for AR/VR creation should facilitate the early stages of exploration by offering examples that creators can use to see design possibilities at a high-level and simplify the descriptions of needed components.

DG4: Assess relevance of learning materials in relation to the creator's background and desired goals. A key challenge for newcomers often is recognizing relevant projects and tutorials from a list of search results as they lack a mental model of the underlying application [123]. Onboarding techniques should use visualization to provide an intuitive "at a glance" explanation for suggested learning materials. For example, highlighting matching metadata when presenting recommended content has been shown to be a useful technique [102]. Moreover, features like adding contextual cues to search results can provide *information scent* [74, 67] that can help users more quickly and easily navigate the results.

DG5: Offer freedom and flexibility in personalizing and exploring relevant learning materials. A known drawback of personalization and profile-based recommender systems is that the underlying algorithms can be opaque to the end user, especially if the system is not open to user inspection or modification [103]. Instead, onboarding should be both *adaptable* (e.g., allow for manual configuration by the user), as well as *adaptive* (e.g., provide proactive personalizations to satisfy the needs of the user) [189]. If the adaptation logic is defined in the form of rules, users can gain control over the system by being able to inspect, understand, and modify the underlying adaptation model. This is particularly important for supporting the wide range of AR/VR creation possibilities and diversity among new creators.

4.4 PONI: System design and implementation

Based on the above design considerations, we followed an iterative design approach consisting of rounds of sketching and wireframing, and elicitation of user feedback [266]. We designed PONI, a novel *personalized onboarding interface* to help AR/VR newcomers with diverse backgrounds, skill levels, and range of development goals to locate relevant learning materials. The design splits across three main modes of interaction: (1) *the input module* where the users specify their background, skills, desired outcomes, and constraints; (2) *the suggestion module*, which suggests relevant learning materials based on user input and allows users to assess the suitability of each resource; and, (3) the *project description module* allows users to see details of each retrieved result and assess their relevance using metadata such as the required hardware, authoring tools, and programming languages, among others.

4.4.1 Input Module: Defining user characteristics and desired outcomes

Since PONI is a new design concept, we did not have access to users' information in advance. For providing personalized onboarding experience, we decided to use a rule-based approach

[216, 89] that focused on incrementally asking for various user characteristics, preferences, and constraints related to AR/VR development.

Determining the type and category of immersive experience

Fulfilling DG2, the input module enables users to identify their target project by learning about the general concept and vocabulary used in creating immersive experiences. PONI prompts users to choose the type of immersive experience (Figure 4.1.a) among two available options (*Augmented Reality* or *Virtual Reality*). We arrived at using AR and VR as the primary representative categories of immersive experiences and excluded Mixed Reality (MR) from available options due to the ambiguity and disagreement in the definition of MR in the literature [230]. Next, PONI prompts users to specify the category of experience (Figure 4.1.b) within AR or VR technologies [127, 230] by adopting the clustering introduced in existing approaches [265, 230, 127]. This step helps users further specify the type of desired immersive experience (e.g., a marker-based vs a location-based AR experience or a simulation vs. 360° VR experience) by browsing examples (more details can be found in Table 4.1).

Specifying the target tools and type of outcome

To address DG2, PONI also prompts users to specify their intended ways of experiencing the output. Following DG3, all tools are shown with images to provide learning opportunities and to support recognition over recall (Figure 4.1.c) [187]. Moreover, users with budget constraints are provided with an option to specify a budget range for further customization and user control. PONI provides an "*I don't know; Please assist*" choice to accommodate the decision-making process and facilitate learning through exploration if users have no particular preferences or are unsure what to choose.

Determining user skill sets and constraints

Considering differences in technical skills and motivations of the creator base of AR and VR experiences (DG1), PONI prompts users to self-define their programming experience, 3D modeling familiarity, and approximate the time they want to spend on their desired AR/VR project (Figure 4.1.d).

4.4.2 Suggestion module

Locating relevant learning materials

In keeping with DG4, PONI's results page (Figure 4.1.e) lists projects in ranked descending order of match relevance (the matching algorithm is detailed in 4.4.4). Each example project is introduced by a card containing a representative image and information about the factors

Type	Categories	Description
AR	**Object -Dependent AR**	Also known as marker-based experiences. This AR experience looks for a specific image pattern in the environment and superimposes the virtual objects on top of it.
	Object-Independent AR	Also known as marker-less experiences. Object-Independent AR detects objects or characteristic points of a scene without any prior knowledge of the environment.
	Location-Based AR	Location-based AR ties augmentation to a specific place and works by reading data from a device's camera, GPS, digital-compass, and accelerometer while predicting where the user is focusing as a trigger.
VR	**360° VR**	This is a semi-immersive experience with minimum capabilities for interacting with virtual environment and zero to low-dependency on VR-enabled technologies such as HMDs and controllers.
	VR Story Telling	While having several similarities with 360° and simulation experiences, VR storytelling mainly focuses on the content of the VR experience and how to gain the user's attention.
	VR Simulations	Simulations as fully immersive experiences requiring VR-enabled devices. This category of VR experiences encases the audio and visual perception of the user in the virtual world and cuts out all outside information to ensure a fully immersive experience.

Table 4.1: To create augmented reality experiences, users are asked to identify the method used for triggering augmentation action. PONI adopts the clustering used in [214, 24, 256] to offer representative modes of interaction in AR and VR.

such as programming level, 3D modeling level, category, output type, and estimated completion time (Figure 4.3). The suitability of each factor is portrayed through four colours from green being exact, yellow being close, red being weak, and grey being no match to reduce users' cognitive load and support recognition over recall. By clicking on each project card, users gain access to in-depth details of a given project's creation details (see 4.4.3).

Customizing suggestions

Applying DG5, PONI offers user control by allowing users to manually customize the importance of factors influencing the ordering of suggested projects (Figure 4.4). To simplify the interpretation of the numeric schema used in the background algorithm (see 4.4.4), PONI

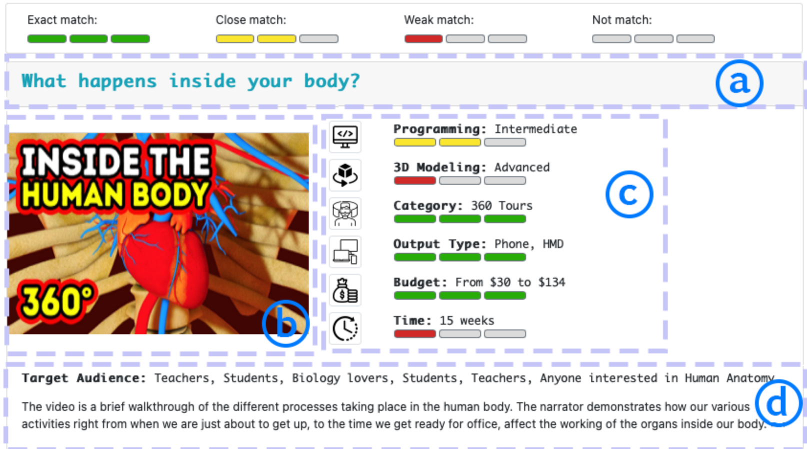


Figure 4.3: An instance of a project card that constitutes a single result retrieved by PONI. It has three main parts: (a) a header showing the project's title, (b) a preview image from the project, and (c) color-coded bars showing to what extent each project matches the user inputs (e.g., programming, 3D modeling, category, output, budget, time), and (d) a brief project description and the target audience.

presents the importance of each factor using a descriptive schema (e.i., not very important, somewhat important, and very important).

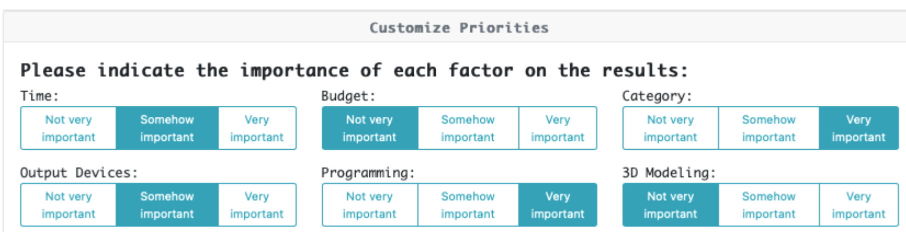


Figure 4.4: The results page contains customization options for users to define the impact of each user input on showing the ordered list of matching projects. The initial importance (explained in 4.4.4) of each factor is highlighted in blue and can be updated by the user. When the user changes the importance of any factor, the order of the results update accordingly to revised user preferences.

Filtering suggestions

By default, PONI presents all matching projects ordered from most to least relevant. However, addressing DG5, users can also apply hard filters and reduce the size of their search space (e.g., only show projects related to simulations). PONI applies these filters over the system's initial suggestions (see Figure 4.5).

Figure 4.5: The filtering tool presents the choice for users to limit recommendations to projects that strictly follow the defined criteria. The user's initial inputs are shown as highlighted to support recognition over recall.

4.4.3 Project description module

Following DG1 and DG4, each project page is personalized and designed to offer an appropriate level of information based on different user characteristics. In particular, there are two types of project pages:

1) *simplified project pages* (Figure 4.2) that target non-technical creators and minimize jargon to allow users to quickly see whether or not the project fits their needs.

2) *intermediate project pages* that target creators with intermediate to advanced skills in 3D modelling and programming. This page provides more detailed information about the recommended and optional programming languages, frameworks, 3D modeling tools, output devices, and the hardware requirements for creating a similar project. If a newcomer to programming sought more details of a given project, PONI directs them this page template.

4.4.4 Implementation

PONI is a platform-agnostic web-based application written in HTML, JavaScript, and Python in the Django framework. The goal was to create a proof-of-concept implementation of the personalized onboarding concept with real-world data that could be evaluated with real users. Once a user submits their initial preferences questionnaire (described in 4.4.1), PONI matches user preferences (described in 4.4.4) against its back-end curated database of AR/VR projects (explained in 4.4.4). In the current implementation of PONI, the database consists of a manually-curated collection of over 100 AR and VR example video projects and tutorials (this database can be made available to other researchers upon their request).

Gathering example projects and tutorials

Given the popularity of video tutorials in self-directed learning practices [123], we populated PONI's database with English-speaking YouTube videos. We aimed for diversity in the

AR/VR technology used, level of complexity in 3D modeling and programming, authoring software, and project topics (see Table 4.2). We ensured that all videos showcase at least one project and excluded general explainer videos on AR and VR.

Technology	Category	Topic	Device
Augmented Reality	Marker-based AR	Education	Oculus Quest/Quest2 examples
Virtual Reality	Marker-less AR	Industry	HTC Vive/Pro examples
Location-based AR	VR storytelling	Marketing	Google Cardboard examples
AR/VR showcase	VR simulations	Biology	Microsoft Hololens examples
AR/VR tutorial	360 videos/tours	Games	Phone AR examples

Table 4.2: Examples of the keywords used for finding AR/VR projects or tutorials on YouTube. We aimed for diversity in the technology used, level of complexity in 3D modeling and programming, hardware, software, and topics of the projects.

Curating gathered projects

Two of the authors with prior experience in AR/VR creation independently annotated videos gathered from YouTube. To construct a consistent annotation schema, these authors looked at prior work on roles of user expertise interacting with user interfaces and feature-rich software [94] and challenges of non-professional AR/VR creators [15, 184]. As a final schema (see Table 4.3), the authors annotated each project based on the type and category of experience within AR/VR, the difficulty level of 3D modeling and programming of each project, estimated budget (including physical equipment and general software licences) and time to complete a project given level of difficulty, recommended and optional creation platforms, 3D modeling software, hardware, and relevant keywords used in the creation of the project. The annotations also included a general project description and possible target audiences of each project. The authors had two rounds of discussions to ensure the consistency of annotations. In the first round, researchers looked for similarities and differences in the curation to reach agreement in annotations. In the second round, after completing the curation on both sides, an inter-rater reliability test was applied to ensure annotation consistency, achieving a Kappa score of *0.81*.

Presenting ranked list of projects

Based on the responses collected through the input module (see 4.4.1), PONI's internal algorithm starts to match user inputs against its curated database (see 4.4.4). The ranked list of matches consists of individual project cards (see 4.4.2). We defined a scoring function to sort and present a ranked list of matching projects using two criteria: *closeness* of match between user input and the corresponding curated project attributes, and the *importance* of each of the attributes (e.g., programming, 3D modeling, output, budget, time). To visually

Category	Frameworks	Modeling Tools	Output Type	HMDs
Marker-less AR	Unreal	Blender	Desktop	Google Cardboard
Marker-base AR	Unity	3Ds Max	Web	HTC vive
Location-base AR	Cry Engine	SketchUp	Smart phones	Oculus Quest2
Simulation VR	ARKit	Turbo squid	HMDs	Hololens 2
360 VR	ARCore	Free3D		Playstation VR
Storytelling VR	ARWeb	CGTrader		Oculus Rift & Go

Table 4.3: The schema used for the curation of AR/VR projects. Authors curated each project by assigning them appropriate type, category, programming and 3D modelling level, output type, and development frameworks (Types, categories, and output type are based on categories introduced in [24, 214, 256].

demonstrate the extent to which each attribute matches user input, PONI renders the closeness factor using the color schema introduced in 4.4.2 (Figure 4.3).

Closeness of Match: this considers the extent to which the user preferences match the curated project attributes (i.e., No Match, Weak Match, Close Match, and Exact Match) with respective values of 0, 1, 2, and 3. We compare eight attributes of each project's annotations and the user's answers to: type TY, time T, budget B, category of experience C, head mounted display H, output devices O, 3D modeling experience M, and programming experience P questions. If the user was unsure about any of the mentioned factors (e.g. answered "I don't know"), that factor was considered as an Exact Match in the score function to exclude its effect in the final ordering. To calculate the closeness of match, based on the data type of each eight attribute (i.e., numeric, nominal, or ordinal), we applied the following rules:

- **Numeric data (time and budget):** Each numeric attribute is assigned a *close match threshold* T_c and a *weak match threshold* T_w, which delineate the cutoff distances for close and weak matches, respectively. In the questionnaire, the user selects a range X for each numeric attribute (see 4.1.4), which is then compared with each project's corresponding attribute value y. If $y \in X$, the closeness for that attribute is equal to 3 (exact match). Otherwise, the closeness is determined by whether the distance between y and the boundary of X lies within the thresholds T_c (closeness 2), T_w (closeness 1), or neither (closeness 0).

- **Nominal data (type [AR/VR], category of experience, output device, and HMDs)**: For each project in the database, a closeness of 3 (exact match) is assigned when the user's selected value is equal to that project's corresponding attribute value. Otherwise, a closeness of 1 (weak match) is assigned.

- **Ordinal data (programming and 3D modeling experience):** Similar to nominal data, but non-exact matches yield a closeness of 2 (close match) when the user's selection is *greater* than a project's corresponding value (i.e., when the user has *more* experience than required). A user's selection that is smaller than the project's value still yields a closeness of 1 (weak match).

Importance of Attributes: PONI determines how each of the above factors are important for ordering (e.g., Low, Medium, and High with respective weights of 1, 2, and 4). The importance of each factor can also be customized by the user on the results page. The importance of each factor can be customized by the user on the recommendation page. To set default parameters for initial recommendation list we applied the following logic:

- **Low importance (3D modeling and budget):** 3D modeling skill is essential for AR/VR experiences that need customized models. However, due to the availability of online pre-made models (e.g., TurboSquid or CGTrader), users' lack of 3D modeling expertise does not affect their ability to complete a project. Also, due to the variety of available devices used for AR/VR creation, users of varying budget allowance can create most of the projects.

- **Medium importance (Output devices, HMD, and time):** AR/ VR experiences can be tested through most output systems (e.g, even desktops) since they are the initial places where the experiences are built. However, if users want to experience their creations through a specific platform (e.g., HMDs or mobiles), some limitations need to be considered before initiating a project. Moreover, newcomers may be imprecise in their estimation of the needed time commitment due to their limited grasp of the AR/VR creation landscape. Hence, time can be an important factor in the creators' initial project planning and decision-making.

- **High (programming experience and category of the experience):** Given the differences in the newcomers' skillsets (see DG1) and that coding skills are necessary for AR/VR creation, the importance of programming experience is set as high. Also, since the category of experience can define the devices and skill sets, it is important for generating accurate relevant recommendations.

The ranked list of matches consists of individual project cards (see 4.4.2). To visually demonstrate the extent to which each attribute matches user input, PONI renders the closeness factor using the color schema introduced in 4.4.2 (Figure 4.3). Project cards are sorted in a descending order based on the scoring function S calculated with the equation 4.1.

$$\begin{gathered} S = w_1.TY + w_2.T + w_3.B + w_4.C + w_5.H + w_6.O + w_7.M + w_8.P \\ w_i \in \{1, 2, 4\} \qquad TY, T, B, C, H, O, M, P \in \{0, 1, 2, 3\} \end{gathered} \tag{4.1}$$

To handle cases where users select the "I don't know" option, PONI excludes the selected factors from equation 4.1.

4.5 Usability study

To evaluate the extent to which the personalized onboarding concept introduced in PONI helps AR/VR creators find examples and tutorials, we compared PONI to a non-personalized keyword-based retrieval approach which resembles the "status-quo" of finding online materials. We ran a usability study with 16 users who were new to AR/VR creation and assessed their perceptions of usability (ease of use), utility (ease of locating relevant materials), and engagement (feel of control, confidence, and system demand) (Figure 4.6) using quantitative and qualitative methods.

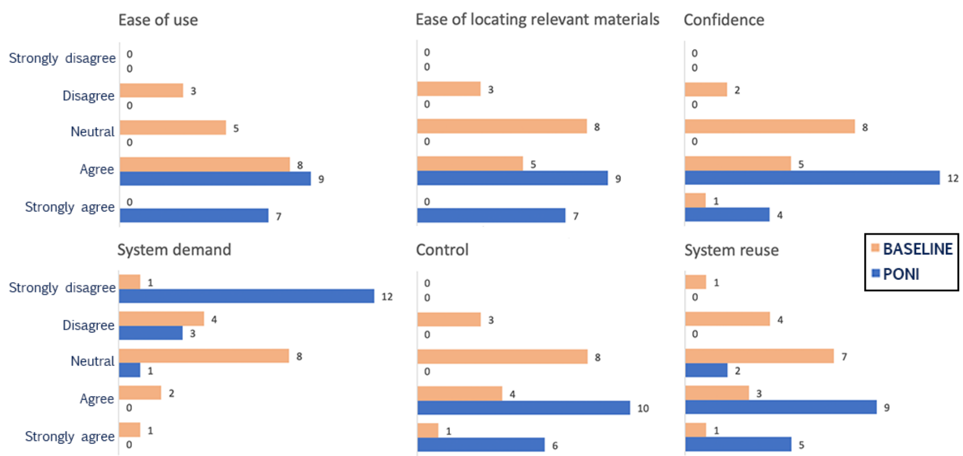


Figure 4.6: An overview of participants' responses to post-test questionnaire options assessing usability (ease of use), utility (ease of locating relevant materials), and user engagement (feeling of control, confidence, system demand, and reuse value).

4.5.1 Baseline interface used for comparison

To evaluate how the personalized learning materials suggested by PONI are perceived by users, we implemented another retrieval interface (which we will call BASELINE henceforth) that incorporated the same database of AR/VR projects used in PONI but did not personalize the retrieval. This interface (Figure 4.7) provided a keyword-based query interface and matched user queries against the curated metadata used in PONI. The formatting of the search results displayed in BASELINE was similar to PONI in that users could see each project's metadata through project cards. However, the key difference was that BASELINE did not provide any visual cues indicating to what extent each factor matches user inputs. Since no personalization was provided within this condition, all users saw the default project description page that provided full access to all of the metadata.

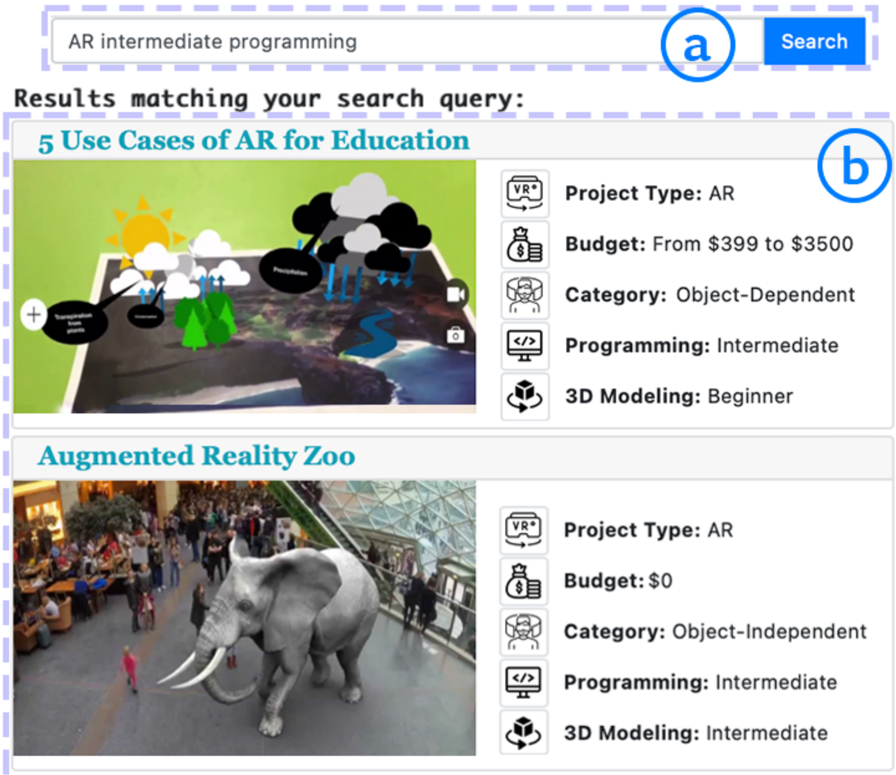


Figure 4.7: The BASELINE interface consists of a search bar (a) where users issue queries using keywords. The list of results shown only match the user's search query and there is no personalization. Similar to PONI, (b) the projects are listed in separate cards indicating metadata. But, unlike PONI, BASELINE does not provide any visual cues indicating to what extent each factor matches user inputs.

4.5.2 Participants

We focused on recruiting participants who were new to AR/VR and excluded any experienced AR/VR developers so that we could study their perceptions of the onboarding process and reduce any prior learning effects. To obtain a broad overview of AR/VR creation practices, we recruited a diverse pool of participants (9F/7M) from different backgrounds (CS, Architecture, Arts, UI/UX design, Industrial Design). Our participants were all between the ages of 18-34 and had different levels of education (7 Bachelor's, 6 Master's, and 3 PhDs). All participants reported their initial interest in AR/VR creation. The majority (12/16) of participants were completely new to AR/VR creation while a few (4/16) had tried to create an AR/VR experience in the past but still self-identified as novices. We recruited these participants mainly from university mailing lists and personal connections in the local community.

4.5.3 Study design and tasks

We used a within-subject design to minimize the impact of high variation among participants. Participants completed four tasks in total under two conditions (two using BASELINE and two using PONI) where each task asked them to select learning materials using one of the interfaces. Participants were asked to locate at least three relevant learning resources for a given AR/VR creation scenario and creator persona (an example is shown in Figure 4.8. We defined personas and tasks based on documented experiences of newcomers to AR/VR [15, 127].

Task Description
Armstrong school is planning to run a competition with the goal of exposing students to new tech trends. As a starter, the school is running a VR competition. Students are encouraged to be creative and try to build a Simulation experience through which they teach other students any topic they find interesting. The competition is going to happen in 12 weeks. Imagine you are volunteering to encourage students to enter this competition and create a VR simulation experience. A student, Emily, has been assigned to you who has never done any programming and has intermediate 3D modeling experience. To encourage her to continue this path, her parents have allocated a $600 budget for her to buy a head-mounted display device but they don't know what their options are. Your job is to help Emily find example tutorials and projects for developing a VR Simulation application. Please find 3 example projects that you think are most relevant for learning about creating VR simulation projects reflecting Emily's time and budget constraints and her programming and 3D modeling background.

Figure 4.8: One example of the scenarios used in the user study.

4.5.4 Study procedure and measures

We conducted the study both in-person and remotely through Zoom, and participants received a $15 Amazon gift card for their time. Participants were asked to log in to our web portal with pre-assigned credentials. This portal allowed them to access both test systems (PONI and BASELINE). To minimize user bias, each system was assigned a pseudonym (e.g., Green for PONI and Blue for the BASELINE condition). Next, participants filled out a pre-test questionnaire (via *SurveyMonkey*) that captured demographics and information about prior experiences in AR/VR creation learning (e.g., familiarity with AR/VR creation software and platforms, general experience in finding learning materials for AR/VR creation). We presented each of the tasks one by one in random order. After completing each task, users filled out a post-task questionnaire (via *SurveyMonkey*) to assess their perception of usability (ease of use), utility (ease of locating relevant materials), engagement (feeling of control, confidence, and system demand), and system reuse. Lastly, we carried out follow-up interviews to further probe into the strengths and weaknesses of each onboarding system design. Sessions were video and audio-recorded for transcription, and the participants were asked to share their screens through Zoom during the usability test. The usability test and follow-up interview took approximately one hour.

4.5.5 Data analysis

We used a combination of statistical tests and an inductive analysis approach [53] to analyze the study data. We ran Wilcoxon signed-rank tests with the nominal variable "system" (having two levels: PONI and BASELINE) and ordinal variable "agreement" having five levels (Strongly Disagree, Disagree, Neutral, Agree, and Strongly Agree) to quantitatively determine the significance of the results. We analyzed the participant's qualitative feedback using affinity diagrams and explored themes around users' perceptions of the personalized onboarding concept. Through discussions with two members of our research team, we categorized our findings and identified key recurring themes related to usability, utility, engagement, and areas of improvements.

4.5.6 Results

Overall most of our participants preferred PONI (15/16) over the BASELINE condition for locating relevant AR/VR learning materials. We next present users' perception of usability, utility, and engagement as they interacted with the two different conditions in our study.

Usability

Users found PONI ($Mdn = 4$) to be easier to use than BASELINE ($Mdn = 3.5$) and this difference in perceived ease of use was significant ($z = -3.14$, $p < 0.002$). Participants found the question and answer, step-by-step style of PONI to be "light" and "easy to manage" as it provided them with an "intuitive" starting point (even when these creators had limited or no understanding of what AR/VR projects they should look at). In contrast, the open-ended nature of keyword search in BASELINE felt like a "black box" and left them "uncertain" and "clueless" about what needs to be prioritized and what a meaningful query should even look like:

> *"In [BASELINE] it got a little bit tricky to know which keyword I need to search for...because, I can go through 10 results or change the keywords 6-7 times, but because I had to check the suitability of them all, it was a bit time-consuming." (P5)*

Utility

Overall PONI ($Mdn = 4$) was ranked higher than BASELINE ($Mdn = 3$) in terms of ease of locating relevant materials and this difference was significant ($z = -3.39$, $p < 0.001$)). Participants explained that the simplified terms and examples used during input served as "clues" and helped them with understanding the AR/VR nomenclature:

"It [PONI] is just intuitive;...I appreciated the question and answer method; it held my hand a bit more. Also, the features of filtering and prioritizing kinda gave you some clues about the metrics and terms you need to look for." (P3)

In addition, participants indicated that PONI could be a useful "hub" for figuring out what the requirements should be for a given project and what current standards are before they even touch an authoring tool:

"It [PONI] helped me know what I need to be considering. Because in the beginning, I do not have a set standard in my mind...I did not have to type anything, and it meant that I did not have to figure out the words." (P1)

While our quantitative analysis showed that there was significant overall user preference for PONI, some participants who were confident in their programming skills (known as *Hobbyists* in [15]), visual cues and automatic ordering of the results were not enough for locating relevant materials. These participants were ambitious with their learning goals, and found that some of the retrieved results were perhaps "too easy", "limiting" or "not challenging enough".

User engagement

Participants indicated that PONI ($Mdn = 4$) provided significantly higher control ($z = -3.13$, $p < 0.002$) compared to the BASELINE condition ($Mdn = 3$). Users found PONI helpful in making informed decisions as it allowed them to choose their priorities and maintain control over their preferences:

"...With this one [PONI] I am able to judge better whether those results fit my criteria or if they do not fit how off or different they are from my criteria." (P3)

Another aspect of PONI that provided more control to participants was the centralized way of presenting information:

"...when I was taking my class, I had little knowledge of AR/VR... I had to read Reddit forums or Google Poly and post on them because that was the only way I understood the terms and processes but still did not get a full picture." (P9)

While PONI was helpful in anchoring participants to examples that fit their skills and constraints, we observed that some users wanted to also find relevant materials based on topics of interest (e.g., all classroom-related AR examples). The current version of PONI was not able to provide this low-level of control over topic so that users could cluster projects that are thematically similar.

PONI ($Mdn = 4$) was also ranked significantly higher than the BASELINE ($Mdn = 3$) condition in helping participants feel confident about their selections ($z = -3.03, p < 0.002$).

A recurring sentiment expressed by participants was that the simple language used in PONI helped them take the appropriate next step and gave them more confidence about the criteria and keywords they should look for. With BASELINE, all participants reported frustration in formulating and re-formulating search queries from scratch as they struggled to assess the relevance of the results.

Participants ranked interacting with PONI ($Mdn = 1$) significantly less demanding than the BASELINE ($Mdn = 3$) condition ($z = -3.49$, $p < 0.001$). They also were more likely to reuse PONI ($Mdn = 4$) than the BASELINE ($Mdn = 3.5$) tool in the future ($z = -3.47$, $p < 0.001$). Participants noted that the visual cues provided in PONI made it easier to peruse through search results and explore learning resource alternatives. With BASELINE, users had to spend extra time and effort to determine relevance, usually with little success:

> *"I found it [BASELINE] cognitively demanding because I needed to memorize so many different things at the same time. I was trying to make it easier by opening multiple tabs so I did not forget the ones I liked...I felt the need to take a look at all of them, but at the same time, I could not evaluate all of them in time." (P7)*

4.5.7 Areas of improvement

Although PONI was perceived to be significantly better than the BASELINE condition for all of our measures, we did synthesize some potential areas of improvement for personalized onboarding based on the user feedback. First, some participants in the interviews reported a slight preference for using a combination of both the keyword-based and a rule-based personalized approach. These participants indicated that they could initially benefit from having PONI's exploratory approach for understanding a project's requirements, keywords, and feasibility, but would like to be able to search by keywords as they gained more experience. Some participants also shared a need for a side-by-side comparison of projects and a more in-depth analysis of how suggested projects map their individual profile. While PONI was helpful for participants in locating and recognizing suitable learning materials, there was no easy way for them to compare the pros and cons of a group of similarly relevant projects. Users also indicated that they would feel more confident about trying out a project if they could see how their skills (e.g., in 3D modeling or programming) map to the different steps of a project's creation process.

4.6 Discussion

4.6.1 Key Takeaways

We have contributed the design and evaluation of a novel personalized onboarding interface (PONI) that helps AR/VR newcomers locate relevant learning materials tailored to their programming and 3D modeling skills, development goals, and constraints, such as time or budget. Our initial evaluation indicates that for AR/VR newcomers, the personalized onboarding process offered by PONI can be more intuitive, useful, and engaging compared to exploratory keyword-based search methods. Our work complements ongoing efforts in AR/VR authoring (e.g., [183, 182, 159]) to lower the onboarding barriers for creators from different backgrounds by assisting them in understanding the landscape of AR/VR creation and getting inspiration for projects.

Although our focus in this paper was on AR/VR creation, we believe that our approach of facilitating personalized onboarding can be generalized to other complex design tasks, such as 3D modeling [123, 122], and software development tasks [154, 47, 255, 133, 55] where there is need to understand the landscape and relevant terminologies before beginning the complex authoring process. Our approach complements other innovations in personalized systems for learning and information-seeking by bringing in a human-centered lens and an interaction design approach for tackling the problem. For example, most of the research on personalized information retrieval has focused on the optimization of the underlying algorithms [179, 39, 42], and it is rare to see explicit focus on the interaction design of these systems that captures users' perceptions. Our research shows the importance of observing people using such systems and capturing their perceptions of the effectiveness and utility of personalized results. Insights from this study reveal that there is more in play than just the effectiveness of a retrieval algorithm and factors such as the UI design, user control, transparency, and flexibility all impact users' overall impressions.

We now discuss some limitations of our current work and highlight promising directions for future work in HCI to further expand the design space of personalized onboarding.

4.6.2 Limitations

Although our implementation of PONI as a proof-of-concept web-based application was useful for assessing users' initial reactions and perceptions of personalized onboarding, more research is needed to fully understand how users would interact with such tools in their actual learning tasks (for example, through a longitudinal field study). Given the scope of this research, the factors that we considered for curating existing projects in our database may not be exhaustive and it is important to keep understanding and addressing the evolving needs of AR/VR creators. However, given our current design and implementation, a natural extension for expanding the curated database would be through the use of methods such as crowd sourcing which have been successful in other learning contexts [259]. Lastly, our current scoring function used for ranking the results only considers three levels for proximity and importance, and future work can explore more granular levels to improve the distinctive power and accuracy of the retrieval in more complex onboarding scenarios.

4.6.3 Expanding personalized approaches

In our current implementation of PONI, we made use of a rule-based approach to retrieve and suggest learning materials based on user input. While this approach has been shown to be accurate in controlled scenarios, it can be blind to user context and may offer low flexibility in supporting spontaneous user interactions [216, 89] (e.g., new items or users that do not fit in any pre-defined clustering). In addition to rule-based approaches, more flexible forms of personalized recommendations can be explored that leverage information from user interaction. For example, collaborative filtering using ratings or other forms of user-generated feedback [31, 105, 222] can determine preference commonalities between groups of users and generate recommendations based on inter-user similarities, which can be particularly useful for supporting the diverse needs of AR/VR creators. Furthermore, content-based filtering [17] also can be used to generate finer-grained recommendations based on the history of a particular user's interactions. Our curated, labeled database of projects can also be used to explore more sophisticated automatic approaches using machine learning and similar techniques [212, 256]. But, some caution has to be used as retrieval and recommender systems that rely on automation and predict user behaviour based on current patterns can suffer from the cold-start problem [216, 89] when they initially lack meaningful data for creating user models.

Based on the insights from our study and prior work on hybrid use of rule-based and adaptive recommender systems [37], there could be some benefits in exploring a combination of both approaches. For example, it can be helpful to learn about user goals, skills, constraints, and context through the initial input and the personalized onboarding systems can adapt as learning progresses. It could be interesting to create experiences that support the delivery of short-term contextual recommendations (e.g., what device is appropriate for my end goal?) and higher-level long-term global recommendations (e.g., task flows and different road maps for creating a target project). Such systems can also track and progressively monitor newcomers' progress when applying recommended solutions and completing steps on authoring platforms, predict their needs, and recommend learning materials appropriate to their context. Given the impacts of personalization on how a user experiences and gets exposed to a technology, it can also be interesting to consider the effect of the *filter bubble* [186] of personalization which can isolate people from a diversity of viewpoints or content. An interesting challenge for the future research would be the investigation of interplay between learner-directed exploratory and personalized learning methods.

4.6.4 Supporting long-term engagement

Accommodate high-level and low-level user learning needs.

AR/VR creation and other digital creative processes [76, 184] often require working across a chain of different tools. Professional developers may be able to work with tools at the high end of the toolchain (e.g., Unity) since they have the relevant training and experience, but newcomers often end up wandering around and working with a large patchwork of tools or get stuck with sub-optimal solutions [184, 15, 123]. In the current design of PONI, participants perceived every suggestion by the system to be click-through to provide a reliable gateway without a need for user validation. This opens up the design space for new support tools to focus on providing a project road map joint with relevant learning materials.

Support learning through embedded communities and automated approaches.

While the current design of PONI was perceived to be effective in showing results matching a user's declared interests and backgrounds, there is opportunity to further expand the richness of the retrieved results. For example, as pointed out by participants, having a way to compare the merits of different results would help them make more informed selections. Embedding in additional comments and shared experiences from other creators within each project or tutorial (as has been explored in some recent work [35, 62, 252]) could further enhance the learning experience for new creators.

Facilitate evolving needs and situational interests.

Currently PONI only allows users to declare intents once and offers only basic customization and filtering options (see Figure 4.4). There is little support for users as their individual needs evolve or their situational interests change (e.g., if they change mind about a particular device) over time. A challenge for future work is to investigate flexible ways to provide long-term support for evolving needs of the user. One approach could be providing multiple roadmaps (similar to [134, 96]) for a given project through which users can flexibly change their pathways (e.g., change their method, devices, completion time) based on their evolving interests.

4.7 Conclusions

In this chapter, we have introduced the design of PONI, a novel personalized onboarding interface that assists new AR/VR creators in locating learning materials that are tailored to their programming and 3D modeling skills, development goals, and any constraints, such as time or budget. Users found the step-by-step question-and-answer style, color-coded suggestions, and personalized results to be intuitive, useful, and saw PONI's potential as a knowledge hub for inspiration and self-directed exploratory learning. Our findings provide an initial lens into the potential benefits of personalization for onboarding and early stages of learning about AR/VR creation and could be extended to other informal learning in technical domains.

While the focus in this and the previous chapter has been on the attitudinal feedback from new AR/VR creators, we have yet to gain a comprehensive understanding of their behavior, which is essential for a complete picture of their initial authoring and information-seeking activities. Although PONI offered insights into the early stages of new creators' journeys, it did not shed light on the subsequent steps these creators take after utilizing learning resources. In the next chapter, we will explore the practical aspects of AR creation by conducting an observational study with 12 new AR creators. This will provide us with valuable insights into their hands-on experiences and the actual implementation processes they undertake in AR development.

Chapter 5

How New Developers Approach Augmented Reality Development Using Simplified Creation Tools

My initial studies provided valuable insights into newcomers' attitudes and help-seeking behavior. These studies demonstrated that many newcomers gravitate towards traditional 3D game engines, such as *Unity* or *Unreal*, that provide a simplified platform for integration of 3D experiences with various frameworks and tools and have evolved to offer extensive support and functionalities for AR/VR development [197]. Despite the appealing and simplified AR development environments presented by modern game engines, many new creators still find it challenging to realize their immersive development projects. To understand the types of obstacles developers face in AR development, even with simplified environments, a deeper insight into how new AR developers approach programming and debugging is essential [1]. This became particularly relevant with the introduction of extended-reality tools such as the Apple Vision Pro, which, with over 600 applications at launch [13], aims to bring a diverse set of AR applications ranging from productivity to entertainment through mixed-reality capabilities. This context underscored the importance of focusing on software programmers as the new wave of AR/VR application creators.

5.1 Introduction

Designing and implementing Augmented Reality (AR) experiences is a complex, knowledge-intensive endeavour that has predominantly been carried out by specialized experts in research labs or professional game development studios. Unlike *mainstream* software development for desktop or web environments where the focus is on flat graphical interfaces and

[1] Narges Ashtari and Parmit Chilana. 2024. How New Developers Approach Augmented Reality Development Using Simplified Creation Tools: An Observational Study. Multimodal Technologies and Interaction 8, no. 4: 35. https://doi.org/10.3390/mti8040035. [16]

standard input methods, AR developers are tasked with overlaying digital content and experiences onto real-world environments via mobile applications or specialized Head-Mounted Displays (HMDs) like the Apple vision pro. Developers usually have to navigate an intricate web of development frameworks and hardware options to construct three-dimensional (3D) interactions and heighten the realism of their projects [15].

When beginning AR application development, many newcomers gravitate towards traditional 3D game engines, such as *Unity* or *Unreal*, that have evolved to offer extensive support and functionalities specifically for AR development [197]. Unity, for example, streamlines the integration of diverse AR platforms, such as ARKit for iOS and ARCore for Android, into a single, unified API. This integration enables the creation of AR applications that are compatible across various devices and platforms without necessitating platform-specific coding. Furthermore, developers can utilize additional built-in features and resources to expedite the development process and engage with an expansive community of developers [241, 83, 236].

Despite the appealing and simplified AR development environments presented by modern game engines, many new creators still find it challenging to realize their immersive development projects [15, 14]. To understand the types of obstacles developers face in AR development, even with simplified environments, a deeper insight into how new AR developers approach programming and debugging is essential. This need has become more pressing with the advent of technologies like the Apple Vision Pro, which aims to bring hundreds of AR applications, from productivity to entertainment [13], to the mass market with its mixed-reality capabilities. It is crucial to support new developers and equip them with the necessary skills to navigate the complexities of AR development. This support will empower new developers to create diverse and innovative immersive applications that fully leverage the potential of advanced AR technologies.

In this paper, we investigate how newcomers in AR approach the creation process using a simplified development environment and seek information to support their design and programming needs. We carried out detailed in-lab task-based observations and semi-structured interviews with 12 software developers who were implementing AR for the first time using the Unity development environment. This choice was made as prior research [15, 184] indicates that while new AR creators have access to a range of development tools not requiring coding skills, they inevitably turn to tools that do require coding due to the flexibility and a wide range of functions that allows developers create comprehensive experiences. By focusing on participants who already have some programming and relevant information seeking skills, this study directly explores how their foundational knowledge affects their learning curve and adaptation strategies in immersive AR development.

Among our key findings, we found that new AR developers often relied on their previous 2D development experience and sought guidance from online code examples and tutorials. However, these developers faced challenges in applying their 2D experience to the 3D realm.

Their usual sources of information, such as online forums and YouTube, were too general and failed to address the unique challenges of AR, including the prediction of 3D object behaviour and complex physics. Faced with increasing complexity in AR development, many developers turned to AI-based assistance, only to find that this approach often led to inconsistent results. A primary issue was the developers' narrow focus on finding code snippets, which caused them to overlook the challenges of 3D spatial interactions and the intricacies of AR's hardware and software. Additionally, the tendency of developers to dive into coding without a comprehensive understanding of the broader problem and its components proved ineffective in AR development.

Our paper highlights the shortcomings of popular online learning resources and approaches in preparing new developers for the unique challenges of building interactive AR applications. While there is a long history of empirical research exploring developers' work habits in various engineering tasks [142], their learning strategies [73], and online information-seeking behaviors [131, 124], our study adds new insights about how software developers tackle interactive immersive experiences and how they navigate complex programming structures and frameworks, and tackle debugging and testing tasks. The lessons learned from our work can be used to invent tailored and more effective learning tools and training programs that empower new creators to explore their own projects in AR. The main contributions of our work are as follows:

1. Providing detailed insights through observations and interviews into how developers new to AR make use of a simplified AR development environment, including their use of various online information resources and AI-assisted tools.

2. Synthesizing the common challenges encountered during AR development, especially related to navigating unfamiliar intricacies of 3D environments, and identifying gaps in their coping methods to tackle these challenges;

3. Identifying opportunities and implications for the design of learning tools and approaches to support future authors of AR applications and help them make a smoother transition from mainstream development.

5.2 Related work

This research builds upon insights from HCI and software engineering reflecting on the current landscape of AR tool development, challenges of building domain-specific software, and software developers' information-seeking activities.

5.2.1 Tool innovations is AR application development

Prior work has explored various AR-specific authoring tools tailored to creators with diverse skill levels and different fidelity stages of the resulting artifacts [184, 99]. Notable examples of such tools include Pronto [147], ProtoAR [183], GestureWiz [232], iaTAR [146, 145], ARVIKA [77], Adobe Aero [5], Microsoft Maquette [174], and Reality Composer [12]. These tools have significantly contributed to the field by focusing primarily on supporting the low to medium fidelity prototyping stages of application development. Some of these tools strive to minimize or altogether eliminate the need for extensive programming skills and reach a wider creator audiences.

On the other hand, utility of simplified authoring tools is often limited as they are overly-tailored to predefined tasks, restricting their adaptability to a wide array of platforms, frameworks, and hardware configurations [135, 15, 184]. Secondly, some of these tools are not universally accessible to end users, either due to their limited availability (beyond research labs), lack of community, or the absence of comprehensive features. Moreover, a significant drawback lies in the fact that these tools seldom cover the entire design cycle, from initial prototyping to subsequent development and testing on AR devices, leaving a critical gap in the seamless progression of the development process as demonstrated in my first study [15].

In practice, commercial AR/VR game engines and software development kits, such as Unity [243], Unreal [82], ARKit [11], ARCore [90], A-Frame [165], and WebXR [253] have emerged as the go-to choices for professionals and enthusiasts alike [15, 135, 184]. Such tools have stood out due to their robust features, providing extensive documentation, tutorials, and a supportive community for developers. Due to widespread use of Unity development platform [242], we studied AR developers use of Unity to create their first AR application. This work provides insights into developers' challenges and strategies as they start their first AR development project, complementing my prior works and other existing research [15, 14, 135, 184], offering a comprehensive analysis of real-world AR development practices.

5.2.2 Domain-specific software development

In this paper, we are presenting an observational study of developers new to the domain of AR. The challenges inherent in domain-specific software development have been well-documented, highlighting the varying needs of different user groups across different stages of design [48, 97]. For example, prior work has looked at artists using creative coding languages [207, 154]. This research has identified challenges artists face in understanding abstract representations and adapting to structured workflows [151], as well as efforts to support them through platforms tailored to domain-specific requirements [207, 149]. The discrepancy between general software engineering practices and their application in scientific programming is also noteworthy, highlighting the need for domain-specific methods and tools tailored to scientists' needs [104, 220, 120]. Additionally, game development showcases the unique aspects of specialized domains. It reveals significant differences from other creative industries [175, 8, 245, 137] and poses challenges to traditional software development models' predefined phases [234].

Studies in domain-specific challenges of building interactive applications have focused on managing the volatility of these environments, characterized by frequent changes in users, devices, and software components [218, 129, 4]. This necessitates systems that can adapt or degrade gracefully amidst changes and failures. The trend towards practical, educational, and assistive technologies mirrors the need to address challenges arising from the dynamic 3D space, affecting user interactions and application requirements. In the realm of AR/VR, my first work [15] (discussed in Chapter 3) provides insights into creators' attitudes and preferences by focusing on a broader creator population. This research is a complementary work by focusing on observing new software developers in a lab environment and providing insights into the practical challenges of building AR applications. It reveals nuanced aspects of developers' experiences transitioning from 2D to 3D environments, their information seeking patterns, and use of Generative-AI tools such as ChatGPT as an emerging assistive platform.

5.2.3 Information Seeking in Software Development Tasks

Prior studies of software developers have shed light on their work habits in writing, changing and debugging software [142], their related cognitive processes [73], and their information behavior and needs [131, 124]. Modern software development is known to be intertwined with web search today [213, 155] with developers frequently issuing search queries to seek answers about how to use an API, understand code functionalities, and troubleshoot various issues [226, 150]. Earlier studies with developers [210] indicated that that official documentation is often the first point of reference for developers when learning about a new API, but code examples, peer discussions, and hands-on experimentation with APIs have also ranked high [131].

The process of seeking relevant information presents several challenges for software developers. One key challenge is the "vocabulary problem," a term used to describe the difficulty developers face when there is a mismatch between their understanding of the problem and the language used in official documentation or help resources [78]. This can lead to considerable time being spent on sifting through large and complex sets of documentation, often resulting in reluctance to consult these resources [41, 192]. Another issue comes from the dispersion of relevant information across different sources, complicating the task of gathering all the needed details [158, 210, 211].

We note that much of the existing research has primarily focused on conventional challenges in back-end development and maintenance tasks, rarely capturing the unique challenges faced by developers working in emerging fields like AR where the focus is on creating a compelling user experience that can augment real-world activities. In these new domains, developers are often tasked with creating interactive experiences that involve user input in novel modalities, often through headsets [15]. Our study complements the existing research on developers by shedding light on how developers new to AR approach the development process using a simplified development environment and to what extent they are able to transfer their existing mainstream programming skills and information-seeking behaviors in this emerging context.

5.3 Method

To gain deeper insights into the approaches adopted by software developers new to AR, we employed a qualitative research methodology, including in-lab task-based observations and semi-structured interviews. Our main goal was to understand how newcomers make use of a simplified AR development environment, how they seek information to support their design and programming needs, and how their practices compare to mainstream programming tasks.

5.3.1 Participants and recruitment

Since we wanted to observe AR development process using a simplified creation framework and our tasks required programming, we focused on recruiting participants who had training in software development but had not worked on any AR or VR projects in the past. We employed multiple recruitment strategies, including advertising posters at local educational organizations, leveraging personal connections in industry, and utilizing snowball sampling techniques. By adopting these approaches, we aimed to ensure a diverse participant pool in terms of their backgrounds and skills in programming and design. Our recruitment efforts resulted in a total of 12 participants, with a mix of genders (5F/7M), each bringing unique backgrounds and software development roles to the study, as summarized in Table 5.1. Participants ranged from having 2-10 years of experience in programming using programming languages such as Java, JavaScript, Python, HTML, C#, and C++. Only two participants (P2 and P9) had brief experience working with Unity for 2D game creation.

5.3.2 In-Lab Observations and Task Design

Choice of platform

While platforms like A-Frame and Unreal offer similar functionalities for AR development, we opted for Unity as a representative AR development platform in our investigation. We selected Unity as a case-in-point due to its widespread use (with more than 60% of AR/VR content being made by this platform [242]) and rich feature set, including an integrated physics engine that significantly reduces the need for developers to delve into complex physics and mathematics or write intricate code.

Task selection and refinement

In constructing the task for this study, we wanted to ensure that it was doable and captured a range of competencies and complexities in creating an interactive AR experience.

ID	Gender (Age)	# Years of programming & role	Programming languages
P1	F (25-34)	7-10, Researcher (CS)	Python, C, Java, R, MATLAB
P2	M (18-24)	4-6, Software engineer	Python, C, Java, C#
P3	M (18-24)	1-3, Student (CS)	Java, Python, JavaScript, HTML
P4	M (18-24)	4-6, Student (CS)	C++, Java, Python, JavaScript
P5	M (25-34)	10+, Software engineer	Python, HTML, JavaScript
P6	F (25-34)	4-6, Researcher (CS)	Python, C++
P7	F (25-34)	4-6, Researcher (CS)	Java, Python, C++, JavaScript
P8	F (25-34)	1-3, Researcher (CS)	Python, HTML
P9	M (18-24)	1-3, Student (CS)	JavaScript, HTML, C#, Python
P10	M (25-34)	4-6, Software engineer	C++, Java, JavaScript, Python
P11	F (25-34)	4-6, Software engineer	Python, HTML, JavaScript
P12	M (25-34)	7-10, Software engineer	Python, C++, C#

Table 5.1: Participants' demographic information, years of programming experience, role, and programming language proficiencies. All participants indicated they had completed introductory courses in 3D geometry and linear algebra, either at the high school or university level.

Our goal was to assess how AR newcomers would approach the development problem and seek relevant information when using a simplified development framework. We consulted industry experts, ran three pilot studies with participants sharing the same characteristics as our main study target group, and iterated on various task configurations to gauge their feasibility.

The task was multi-layered, with each component tailored to test different skill sets necessary in AR development. We provided participants with all of the required 3D assets (see Figure 5.1). We instructed participants to create an AR model of Earth with a continuous spinning motion around its own axis. This component was intended to assess the individuals' capability to introduce and manage elementary motion dynamics in an AR environment. In addition to the spinning Earth, participants were required to incorporate a Moon that would not only revolve around the Earth but also execute a spin around its own axis. This layer of complexity ensured that participants dealt with coordinating multiple synchronized motions within the AR space. In the next part of the task, we provided a 3D model of a spaceship and asked participants to simulate a landing mission which required a realistic, physics-driven animation sequence in the AR environment. This involved common AR tasks, including collision detection and defining movement trajectories in a 3D environment. Lastly, the task asked for a simple menu embedded within the AR experience that would have two interactive buttons: "Land" and "Fly Away." Activation of the "Land" button would command the spaceship to initiate a landing sequence onto the moon's surface, while the "Fly Away" button would instigate the spaceship's departure. The inclusion of this component aimed to probe the participants' ability to combine interactivity with immersive visualization, a key proficiency in AR development.

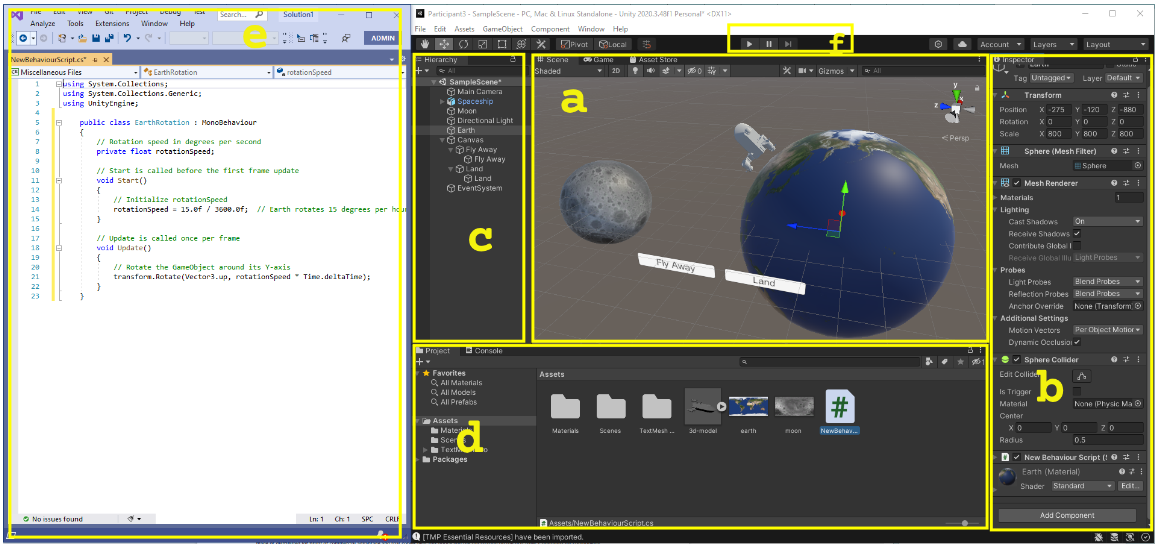


Figure 5.1: Schematic of the multi-layered AR development task for participants in Unity. (a) The "Scene View" where participants interacted with the provided 3D assets. This platform allowed participants to navigate and manipulate 3D objects. (b) The "Inspector Panel" which provides properties of the selected 3D object(s) in the scene view. Through this panel, participants were able to modify objects' attributes like position, rotation, scale, and attach components or scripts. (c) The "Hierarchy Panel" which lists all the objects in the current scene and provides an easy way to select, organize, and manage game objects. (d) The "Project Panel" which is essentially the file browser within Unity and shows all assets, scripts, prefabs, scenes, etc. (e) The "Visual Studio" integrated IDE often used in conjunction with Unity for scripting and code editing. (f) The "Game View" where participants could preview what they built as it would appear when running. Using this participants were able to play, pause, and step through frames for testing.

All studies were conducted in-person. Each session lasted around 2 hours, with participants having 90 minutes to complete the task and 20 minutes to complete post-task questionnaire and answer follow up interview questions. Participants were informed that they could proceed as far as they could within the 90-minute task time frame, and there was no requirement to complete the entire task. Furthermore, the order of implementing different parts of the task was entirely up to the participants; they were free to choose the components they felt most comfortable with or interested in. All participants were instructed to take break as needed. The study facilitator was readily available throughout the research, offering occasional hints to participants in the event of significant delays in their progress. The facilitator remained discreet to minimize any potential impact on the study outcomes while providing support.

Procedure

Prior to the study (at least five days before the study session), participants were sent comprehensive tutorials on how to use Unity, including guidelines for testing their creations, adding interactions, and navigating the interface. They were granted access to Unity along with its built-in help features and encouraged to utilize any web resources available. To capture an in-depth view of participants' actions and thoughts, their audio and screens were recorded, along with their browser histories, ChatGPT conversations if used (version 4 was set as a default option for all participants), and interactions within Unity UI. Before starting the study tasks, participants completed a questionnaire covering demographic details, educational background, proficiency in various programming languages, and preferred resources for seeking help and information. Participants then were encouraged to follow a "think-aloud" protocol to keep the facilitator updated on their logic, creative process, and any challenges encountered. Scheduled breaks were also provided to ensure participants remained focused and comfortable throughout the study.

For a more nuanced understanding of AR deployment, participants were encouraged to deploy their AR models on the HoloLens 2. This step was critical for assessing the robustness of their AR experiences in a real-world, immersive setting. To streamline this process and focus on the core objectives of the study, the actual deployment task was handled by the research team (taking approximately 10 minutes), allowing participants to concentrate on conceptualizing and building their AR models.

Post-task questionnaire

After completing the task, participants were asked to fill out a short survey questionnaire. The questions aimed to gauge the perceived difficulty of programming in a 3D environment as compared to mainstream, non-3D coding, the effectiveness of online resources in participants' information-seeking process, ease of transferring current programming skills to 3D and AR development tasks, and the extent to which participants relied on their existing programming skills to troubleshoot and solve technical problems while completing the study tasks.

5.3.3 Follow-up semi-structured interviews

To reflect on the in-lab experiences and better gauge participants' perspectives on AR development, we carried out follow-up semi-structured interviews. Acknowledging the participants' prior experiences in other domains of development, particularly 2D or other non-3D environments, was a crucial component of the post-task interview process. This served to draw comparisons and contrasts, aiming to understand how the unique complexities of AR development diverge from or align with other forms of mainstream software development. One key area that the interview focused on was whether and how the participants had to adapt their existing skills and strategies to the nuances required by AR development. In particular, semi-structured interviews explored:

- **Participants' experience and skill transferability in AR vs. 2D/non-3D environments:** This would focus on how participants' previous experiences in other environments translated to the AR task. It would also aim to identify skills that were easily transferable and those that required significant adaptation or relearning.
- **Challenges, strategies, and information-seeking behavior:** This would combine the specific challenges encountered with the resources and strategies employed to overcome them. It could explore any changes in participants' go-to platforms for assistance and how effective these were, providing insights into their evolving problem-solving process.
- **Lessons learned and future approaches:** This would capture personal reflections on what participants would do differently in future similar AR tasks, revealing data on the learning curve involved in AR development.

Each interview concluded with an opportunity where participants were encouraged to share additional thoughts, feedback, or reflections not covered by the preceding structured questions.

5.3.4 Data analysis

To analyze the participants' progress, we segmented our primary task into four distinct sub-tasks (as explained in 5.3.2), each revolving around crucial interaction components. Participants' efforts were then aligned with the reference design's sub-tasks. To gain insight into how newcomers identify and utilize various help resources, we initially examined the various phases of their information-seeking behavior in the lab. Additionally, we explored how participants perceived these help resources, drawing on their in-lab interactions while completing the task.

Analysis of help-seeking phases. For our lab-based analysis, we adapted and revised an existing theoretical model on in-person help-seeking by Nelson-Le Gall [81]. Within this framework, we categorized help-seeking behaviors into three main phases: 1) Identifying resources; 2) Assessing resource relevance; and 3) Implementing the relevant assistance to accomplish a task.

Identifying resources. In this initial phase, we evaluated how effectively participants could articulate their need for help and locate relevant resources. Our assessment tools included a query log analysis of search histories, Unity's built-in help, and an evaluation of participants' engagement with Generative AI platforms, such as ChatGPT (where applicable). We sought to identify both relevant and irrelevant resources that participants discovered. Additionally, we gauged the time required for participants to initiate their first attempt at seeking help, examining their navigation strategies and the initial moments they initiated using any available resources for assistance.

Assessing resource relevance. During this phase, we studied what transpired when participants arrived at a potentially relevant resource, as well as how well they could leverage that resource to accomplish their task. For this part of our analysis, one researcher cross-referenced data from participants' browser navigation histories and screen recordings. This helped to evaluate the relevance of the discovered resources and was further substantiated by participants' think-aloud reasoning data during the study.

Implementing help in task completion. In the final phase, we examined the degree to which participants could apply the located help to their current tasks. For this, we relied on browser navigation histories, screen recordings, and participants' attempts at task execution.

Understanding participants' perceptions of help resources. To delve deeper into participants' perceptions regarding the usefulness of various help resources they encountered, we drew upon our observations and participants' feedback from post-task questionnaires and semi-structured interviews. Using an inductive analysis method [52], we searched for recurring patterns and themes in the collected data.

5.4 Results

We present our results by first summarizing the participants' overall performance and approaches to developing an AR experience for the first time when using a simplified development platform, revealing a preference for direct implementation over problem solving. We next present key analysis of participants' information-seeking activities, highlighting their search queries, preferences for information resources and formats, and the use of generative AI tools such as ChatGPT. We then delve deeper into the challenges participants faced, which ranged from initial setup difficulties to the intricacies of 3D development and the challenges of translating 2D development knowledge to 3D environments.

5.4.1 Overview of task completion

Our analysis showed that most of our participants faced difficulty in tackling their first AR task: on average, our participants managed to complete a mere 35.8% of the primary task (25% min - 75% max) with 34% accuracy (12.5% min-62.5% max). The participants indicated that AR development was either much more difficult (8/12) or somewhat more difficult (4/12) compared to mainstream 2D development tasks (see Figure 5.3-a). At the onset of the study, participants had the option to sketch out or strategically plan their development process. However, without exception, all participants chose to dive directly into implementation, expecting to find a designated area within the Unity user interface for writing code (which existed in the Unity UI; see Figure 5.1). Despite being encouraged to access tutorials about Unity's interface and methods for adding interactions to 3D objects, none of the participants had consulted these resources prior to attending the user study session. To initiate interaction with the provided 3D objects in the Unity environment, 9 out of 12 participants relied on a trial-and-error approach, while the remaining three participants directly searched on Google. On average, it took participants 4.25 minutes to initiate their first help-seeking attempt. Although all participants were given access to the Hololens 2 headset to test their code, only 2 participants chose to do so; the remainder opted for utilizing Unity's built-in testing environment to demonstrate and test their projects.

5.4.2 Overview of information-seeking activities

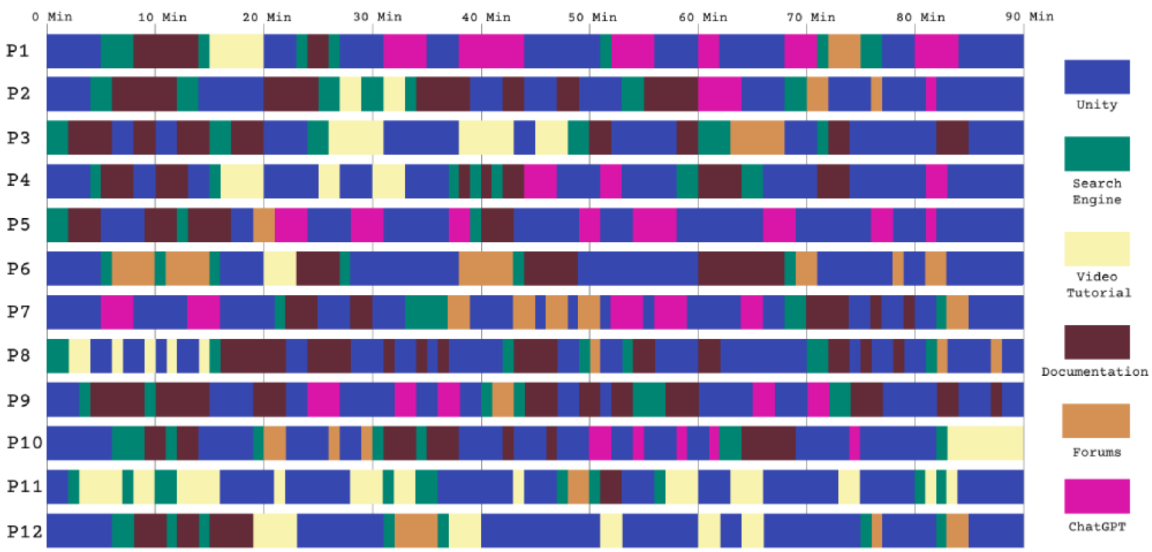


Figure 5.2: Timeline overview of participants and their information resource and development environment navigation. Participants usually began with Google searches to locate learning resources, often consulting official documentation, forums, and code snippets, with an average of 13.9 queries per session. Despite having access to the basics of Unity prior to starting the study, most participants favored learning by trial-and-error, frequently querying Unity interface specifics. On average, they switched 11 times between coding and resources, spending 44.5 minutes outside Unity. While initially favoring written documentation for quick lookup, 7 out of 12 participants eventually consulted ChatGPT after other resources proved unhelpful. But, only P1 and P5 found ChatGPT useful for streamlining their learning and workflow.

All participants relied on Google consistently while completing the subtasks. Similar to mainstream coding projects, they made use of search results that included official documentation, user forums (e.g., Stack Overflow) and snippets of code found on these platforms, followed by video tutorials (e.g., YouTube). All participants were familiar with ChatGPT during the time of the study and more than half (7/12) ended up using it in the study. On average, participants sought help 13.9 times in each session, with the frequency of these attempts ranging from 11 to 19, a variance of 5.17, and a standard deviation of 2.27 (see Figure 5.2). A variety of factors triggered these help-seeking activities (discussed below).

Even though we provided pre-study learning resources on the basics of Unity to help onboard participants, most of the participants relied on their own trial-and-error to figure out the Unity interface. This has been commonly seen in other studies of software learning [123], known as the paradox of the active user [40]. Instead of following the Unity documentation, our participants posed queries like "attaching code to 3D objects", "defining objects in Unity 3D", or generic ones like "how to instantiate an object in C#?". This DIY exploration method, while valuable for its hands-on nature, was time-consuming and often directed developers towards unnecessarily intricate solutions. On average, participants toggled 11 times between the coding interface and auxiliary resources, spending 49% (44.5 minutes) outside the development environment.

Participants' help-seeking behaviors were varied and included searching for learning how to activate specific functionalities and open-source code (55%), integrating multiple or concurrent functions into 3D objects (12%), understanding scaling and metrics (14%), and troubleshooting bugs during code compilation (19%).

In terms of choice of learning strategies and overall task completion rate and accuracy, we saw a range of be our participants. For example, P1 relied on ChatGPT as the primary learning resource, posing targeted inquiries directly related to the task. This approach facilitated the highest completion rate (75%) of the task with a high level 377 of accuracy other participants (62.5%). Conversely, P8 dedicated extensive time to reviewing official Unity documentations. However, their ability to apply the acquired knowledge to the specific task was less effective, resulting in minimal in task completion (25%). On the other hand, P11 predominantly relied on video tutorials as a learning resource to complete the tasks. Although they a higher task completion rate (50%) compared to P8, we observed that P11 was trying to recreate the instructions in the video tutorials. As a result, they struggled to implement specific adjustments 384 and had low overall accuracy (25%).

Preferences for information resources and formats.

As seen in previous studies of developers [131], our participants initially sought written documentation, valuing its quick accessibility and skim-readability for coding activities. Rooted in their prior experience with programming languages like Python and Java, participants expected text-based platforms (e.g., Stack Overflow, Unity forums) to offer immediate, relevant support. However, both our observations and participants' post-task feedback (see Figure 5.3-b) indicated that these resources often fell short in addressing the specific challenges tied to crafting 3D interactions. Participants commonly described these resources as either too generic or not directly applicable to their unique requirements. For example P6 explained:

> *"As my go-to for quick fixes it felt instinctual to turn there [Stack Overflow] when I began tackling the task. But the resources out there just didn't dig deep*

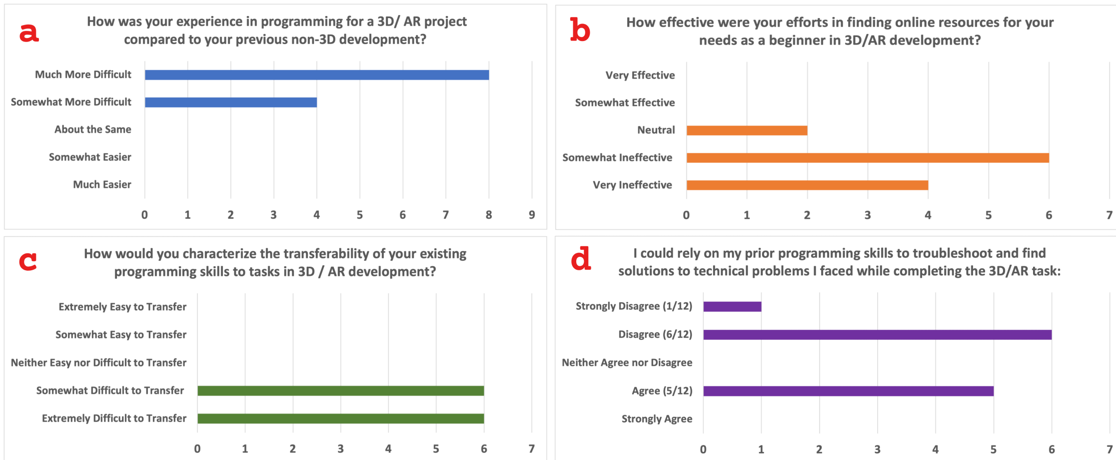


Figure 5.3: Overview of participants' responses to post-task questionnaire. (a) Participants perceived AR development as notably challenging compared to 2D tasks, with most finding it much harder; (b) Most participants found their efforts in locating online resources for 3D/AR development ineffective; (c) Most participants found it difficult to transfer their existing programming skills to 3D/AR environments. The shift to 3D added complexities in visualizing object movements, affecting not only implementation but also problem-solving approaches; (d) Over half of the participants struggled to apply their old "test-and-develop" coding habits in the new 3D/AR environment. For example, they would choose only familiar parts of the code snippets suggested by ChatGPT, but found this strategy unhelpful for debugging intricate AR interactions involving physics forces.

> *enough into the specific issues I was trying to solve or I didn't know how to cater them to my own code. It's one thing to find a code snippet that rotates a shape; it's a whole other ball game to adapt that into a moving, interactive 3D environment." (P6)*

A recurring struggle for participants, as explained by P6, was in understanding the relevance of the found example code snippets, mapping them onto their own codebases, and modifying them to reflect their specific needs.

Video tutorials, on the other hand, were especially described as being useful for visualizing the procedural steps. For example, P2 explained:

> *"So even though there is no explanation [in videos], you kind of figure out the steps...it's not really good for understanding, it's good for doing. The reason I'm jumping into this kind of video is that I've never done anything with AR. But I have 2D Unity experience. That's why I can understand what's going on here, even though I don't really know the 3D stuff." (P2)*

However, some participants conveyed that these tutorials lacked depth in elucidating the underlying principles:

"I don't like learning from videos. Since they are slow. Usually, when you learn via videos on how to create stuff, there's more bloat than there is actual content. I like learning direct...I like to jump straight to the documentation. That's how I learn a lot of new stuff in programming." (P3)

Use of novel generative AI tools for information-seeking.

Despite all participants having previous experience with ChatGPT for programming purposes, only one participant used ChatGPT as the starting point for tackling the task; the majority of participants (11/12) searched for online documentation and forums instead. When these methods failed, more than half of the participants (7/12) tried ChatGPT. However, all of these participants had mixed feelings about using ChatGPT to learn AR development and questioned its reliability based on their coding experiences. For instance, P10 mentioned:

"After trying the documentation and getting nowhere, I decided to give ChatGPT a shot. But based on my past encounters, I wasn't too hopeful about its accuracy with coding issues." (P10)

In contrast, P1 and P5 heavily used ChatGPT, benefiting from a more focused and efficient approach. Specifically, P1 used detailed prompts and follow-up questions to get accurate, context-sensitive advice from ChatGPT (Figure 5.4). This strategy helped P1 and P5 grasp foundational concepts quickly and integrate advanced features, thereby improving their workflow.

Participants who asked ChatGPT vague or context-free questions faced difficulties getting tailored answers. For instance, P4 received generic advice that did not fit their existing setup, highlighting the importance of better contextualizing the dialog (Figure 5.5).

Despite receiving complete code snippets from ChatGPT, many participants only implemented selected parts. This selective use reflected a "test-and-develop" approach based on their previous coding habits. They focused on familiar elements, avoiding more complex features to sidestep potential confusion or complications. For example, P5 used only part of the code for landing a spaceship on a rotating moon and decided to forgo the complex orientation procedures, stating, *"I'll start with what I know and build upon it."* While participants often relied on their prior experience, most of them strongly disagreed (1/12) or disagreed (6/12) that their prior skills were sufficient for debugging and tackling other technical problems in AR development (see Figure 5.3-d).

5.4.3 Challenges in information-seeking when developing AR

Navigating the intricacies of AR development while developing in a simplified environment presented our study participants with a series of nuanced challenges, ranging from difficulties in the initial setup and understanding built-in functionalities, to getting used to

User prompt: "Write Unity C# code for a moon object orbiting itself and Earth."
ChatGPT response (summarized): Delivered script and explanation; user unclear on script implementation.
User prompt: "How to attach this script to the moon object?"
ChatGPT response (summarized): Supplied step-by-step guide for script attachment; user proceeds with creation.
User prompt: "Add rotation for Earth and Moon on their axes. Provide script."
ChatGPT response (summarized): Script and logic explanation given; user uncertain about object interactions and parenting.
User prompt: "How should I run this code?"
ChatGPT response (summarized): Instructions for attaching and running the code provided; user unable to see the effects after testing.
User prompt: "On hitting play, everything resets to zero. What's the issue?"
ChatGPT response (summarized): Received Unity's troubleshooting suggestions; encountered a new problem upon trying to resolve.
User prompt: "Main camera position resets in play mode. How to fix?"
ChatGPT response (summarized): Received irrelevant response from Chat GPT; user seeks clarification.
User prompt: "Scene screen camera doesn't match Game screen. Why?"
ChatGPT response (summarized): User applies the help provided and successfully rotates moon and Earth on their axes and the moon around Earth.

Figure 5.4: Illustration of a part of P1's dialog with ChatGPT - one of the few examples in the study that shows successful use of AI assistance. P1 initiated the conversation by requesting C# code for a moon orbiting Earth. Faced with uncertainties in the script implementation, they followed up to clarify how to attach the script to the moon object. Subsequent questions involved adding axial rotation to Earth and the Moon, resolving issues related to object interactions and parenting. Additional troubleshooting involved resolving camera position resets and discrepancies between scene and game screens. This iterative, context-rich dialog resulted in more tailored guidance from ChatGPT, enabling P1 to successfully implement the desired orbital and rotational behaviors.

the complexities of 3D development and struggles with advanced physics in a 3D environment. The transition from mainstream to 3D development demanded not only a conceptual realignment but also adaptations in troubleshooting and help-seeking approaches. While y first work [15] highlights some of the approaches used by newcomers for prototyping AR/VR experiences, our study focuses on the specific approaches and challenges encountered with the use of simplified development platforms in the implementation phase. In the subsequent sections, we explore these challenges and uncover the adaptive strategies used by developers with different levels of expertise.

Challenges in getting started

A consistent challenge that we observed among participants was difficulty in identifying a reliable starting point as they were overwhelmed by the array of possibilities. For example, as P3 explains:

> *"Knowing where to start was my biggest challenge. That's why I was struggling to figure out what resources to use. Using direct resources like YouTube helped. After learning the basics, such as not needing to render lights individually, it became easier to build on my existing programming skills." (P3)*

User Prompt: "How do I move an object in Unity?"
ChatGPT Response: You can move an object using Unity's animation system. Create an animation clip with the desired movement and play it to move the object.

Figure 5.5: Illustration of a part of P4's unsuccessful interaction with ChatGPT: While P4 already had a script-managed movement system, incorporating animation (recommended by ChatGPT) proved challenging. P4's vague prompt led to generic advice, causing conflicting functions to override each other.

Strategies to "*cope with the unknowns*" varied across the board, echoing distinct developer personas as outlined in Clarke's 2007 study [49]. For instance, P1, who closely resembled the *Pragmatic Developer*, tackled the complex task by breaking it down into smaller, more manageable sub-tasks. The segmented approach improved progress and morale but had drawbacks, including frequent context-switching and difficulty in grasping the overall objectives and best use of learning materials like code examples.

This challenge of achieving a holistic understanding was a broader issue that we observed in the study. A trend among the majority of participants (10/12) was the use of a granular, bottom-up strategy—breaking down the task into basic queries like "how to move an object in Unity" without understanding the overall development task.

On the flip side, two out of the twelve participants managed to substantially complete their tasks by initially adopting a bottom-up strategy and then transitioning to a top-down approach, exhibiting traits of the *Opportunistic Developer*. In particular, P1 and P5 consulted documentations and YouTube tutorials to master foundational concepts before moving to structure their code more comprehensively. Then, leveraged ChatGPT to benefit from its dialogic interaction, tailored advice and context-specific solutions; thereby obviating the need to "reinvent the wheel." This involved turning to platforms like ChatGPT and YouTube to understand the relationships between different coding components.

Lack of awareness of development framework's built-in interactions and affordances.

Participants faced challenges in distinguishing between Unity's built-in functions and custom code when looking for help in online documentation, often leading to unnecessary debugging time. For instance, P12 said:

> *"I saw Transform.Translate and thought I needed to define it myself. Big mistake. It conflicted with Unity's built-in method and slowed my progress." (P12)*

Similarly, P8 added:

> *"When I looked at the code samples in Unity forums, they mentioned 'Quaternion' for orientation and 'Rigidbody' for physics interactions. I couldn't tell if these were built-in features or if they were custom-defined in the script. It took*

me a while to figure out that these are standard Unity components, and I didn't have to define them myself." (P8)

In a related observation, P7 spent significant time manually scripting the spaceship's movement, unaware that Unity's built-in functions could simplify the task. The participant's searches like "object direction in Unity 3D" did not lead into discovering Unity's built-in functionality (NavMesh), partly because of vocabulary issues and the advanced language used in Unity's documentation. This made it difficult for participants with non-AR related vocabulary, to find efficient solutions or identify specific Unity features in the provided code samples.

Difficulties in navigating unfamiliar 3D environments with 2D knowledge.

Our study revealed that, the shift from 2D to 3D environments required participants to adapt their learning methods and search strategies as the complexities of space and orientation take on new dimensions in AR. Overall, all participants found it either somewhat difficult (6/12) or extremely difficult (6/12) to transfer their existing programming skills to 3D/AR development tasks (see Figure 5.3-c). Furthermore, although all participants had completed courses in 3D geometry and linear algebra at the high school or university level, they found it challenging to apply this theoretical knowledge practically in the development of 3D experiences. Their familiarity with mathematical foundations did not necessarily translate into the ability to implement these principles in real-world development scenarios. In traditional 2D environments, objects typically move along two axes, simplifying the process of visualizing their motion and orientation. These complexities intensified the difficulties participants faced in visualizing object movements in the 3D space. This change in complexity influenced not only their interaction but also their methods of seeking help. Supporting this point P11 mentioned:

> *"In 2D environments, I could sketch out motions and interactions on a piece of paper…like reading a map. But now it seems like shifting from a map to a globe. Suddenly, you're accounting for depth, orientation, and multi-dimensional interactions…I'm not sure even if I can ask the right question or if my search prompts would work." (P11)*

Furthermore, the intricacies of 3D visualization were not the only aspect that perplexed participants; the shift from the right-handed coordinate system commonly found in mainstream 2D coding environments to Unity's left-handed system was particularly disorienting. Participants attributed this confusion to their previous experience with standard screen-based 2D graphics (e.g., 2D game development, web development and HTML5, etc.). Misalignment and unexpected behaviors due to this shift were evident. As P4 noted:

> *"Transitioning from my usual coding environment [web development] to Unity's left-handed coordinate system was quite disorienting. My 3D models seemed completely out of place, like they were twisted and scattered across the scene." (P4)*

In their search for solutions, participants frequently drew on terms from their prior knowledge in 2D environments. Common search queries included "Unity 3D vs 2D coordinates" and "2D to Unity 3D transition", with the prefix "2D to 3D" being recurrent.

Challenges of dealing with multiple physics forces and predicting 3D object behavior in AR

Unlike 2D programming, object interactions in 3D environments like AR often involve more sophisticated considerations. For example, we observed that participants faced unexpected behavior while coding a spaceship landing on a rotating sphere, unaware that complex physics forces were at play. As P1 stated:

> *"If it was common game dev I had my spaceship perfectly landed; Introducing rotation threw everything off not giving a hint where the problem is coming from." (P1)*

The unexpected trajectories the participants encountered made them assume coding mistakes, leading them down to assume the existence of potential bugs or development platforms inefficiencies. Typical search queries were along the lines of "unexpected physics behavior during landing" and "debugging landing sequence in Unity 3D". This hinted a potential blind spot in the learning materials: the fundamental physics at play. We observed that despite participants' attempts for finding learning materials, none explicitly mentioned or hinted at the involvement of multiple physics forces in the task at hand. This observation aligns with the findings of my first study [15], highlighting the "physical aspects of the debugging process that remain neglected in online tutorials" of AR/VR development underlining the often-overlooked yet critical aspects of practical implementation.

In addition, predicting object behavior in a 3D environment was challenging due to the complex variables at play, such as rotation speeds, initial positions, and landing trajectories. To decipher the behavior of the spaceship moving from earth to the moon, P8 opted for console-based methods (common in 2D development), and used the print function to analyze raw data such as force vectors and rotation angles. P8's attempt to apply 2D console-based methods quickly proved problematic: the data overload in the console made it tough to identify specific issues, and the numbers lacked the visual context needed for understanding 3D orientations and trajectories. P8 explained:

> *"In Pygame, I could just throw in print statements and quickly figure out what's going on...the numbers directly translated to on-screen coordinates, making it intuitive...I tried the same print-everything approach and got swamped with numbers." (P8)*

P8's conventional method of breaking down and testing the code in parts, typically effective in simpler or 2D environments, did not simplify the complexities of the 3D AR task. Instead, it added to the confusion, making it even more challenging to identify the source of the problem. As a result, P8 resorted to making ill-defined and generic queries queries using phrases "object orientation in 3D" and "understanding object movement in 3D" in search engines, that were not specific enough to direct them to relevant learning resources. This mismatch between the complexity of the task and the search queries led to an ineffective cycle of problem-solving.

5.5 Discussion

5.5.1 Key takeaways

This work can viewed as a specific case study of AR/VR. The study findings contribute insights into how software developers new to AR approach the development process using a simplified development environment. We also looked at the types of information resources they seek and how their learning experience in AR differs from mainstream software development. In particular, we shed light on the unique needs of emerging AR developers, indicating that the types of code reuse approaches that are successful in other development domains may translate poorly AR due to difficulties with unfamiliar vocabularies and intricacies of 3D environments, physics, and motion. Our observations complement and extend prior works that focus on elective, task-focused learning approaches in mainstream software development [246, 19], suggesting that the same "immediate solutions" mindset may be ill-suited for the complexities of AR development. Moreover, while emerging Generative AI tools are showing promising gains in development tasks [155, 19], our research offers an initial look at how these tools may be inadequate for AR development and further widen the skills gap.

With the advancements in mixed-reality devices such as the release of the Apple Vision Pro and the increasing interest in AR technologies, more developers are entering the field of AR development. Many of these developers are learning the necessary skills through online resources and adopting an informal approach to their education. This trend underscores the importance of understanding the challenges these new developers face as they navigate the complexities of AR development without formal training. In light of the findings from our study, it is likely that these new developers will face the same challenges that the participants in our study encountered. We now reflect on the implications of our findings for future research in HCI and the need to reconsider the design of training programs and learning resources to effectively support the growing community of informal AR developers who use simplified development frameworks. We discuss ways to enhance AR development by integrating problem-solving strategies, leveraging in-context personalized approaches and generative-AI tools, and increasing user engagement through adaptive feedback and milestone integration.

5.5.2 Enhancing AR development resources with problem solving strategies

Our study contributes novel insights into developers' challenges with programming AR for the first time using a simplified development framework. Our results build upon prior research that shows developers generally prefer selective, task-focused learning and information-seeking tactics. Previous studies have highlighted that developers often seek immediate solutions to specific problems [106, 30], focusing on getting particular API functions to work [226] or finding workarounds, rather than gaining a comprehensive understanding of the software or its underlying principles [158]. This approach is partly due to the complexity and time-intensive nature of software comprehension, leading developers to prioritize task completion over in-depth understanding.

Our study also shows that AR development presents unique challenges that extend beyond coding skills, requiring a deep understanding of the interplay between 3D elements and real-world physics. Developers often misattribute AR anomalies, like unpredictable object behavior, to coding errors, overlooking the crucial role of physics. This gap in understanding underscores deficiencies in current educational resources and debugging techniques, which are inadequate for addressing the complex interactions in AR. Our findings suggest an urgent need for specialized educational materials and tools tailored to AR development that can, for example, build off literature on problem decomposition in computer science [58, 7, 261]. These resources may include simulation environments visually representing real-world physics affecting digital elements. By offering such insights, developers can better understand the multifaceted challenges unique to AR, enhancing troubleshooting and deepening foundational understanding for success in AR development.

5.5.3 Improving learning through in-context personalized approaches

Standard debugging tools may fall short of understanding and predicting user behavior and application performance. To address this, understanding the task context [121, 29, 88] is essential. For example, the incorporation of advanced logging and machine learning techniques such as collaborative filtering [31, 105, 222] and content-based filtering [17] into development environments like Unity could be game-changing in enhancing both coding efficiency and conceptual understanding. Collaborative filtering would leverage community contributions and feedback to identify common challenges and propose vetted solutions, while content-based filtering could offer tailored recommendations based on an individual users' coding history and behavior within the platform.

The promise of these machine learning methods could be further enriched by integrating AI tools similar to GitHub Copilot into the development environment. Such tools could simultaneously suggest appropriate code snippets and dynamically link developers to contextual learning resources. For instance, if a developer struggles with object physics and encounters errors, the integrated system could suggest an optimized code snippet tailored for Unity's physics engine and couple it with a targeted tutorial that unpacks the relevant physics principles. In doing so, the developer gains not just a quick fix but also a deeper, foundational understanding of the challenge at hand.

5.5.4 Leveraging help-seeking through Generative AI platforms

Our study findings also shed light on the role of Generative AI tools, like ChatGPT, in AR development, expanding upon existing research on AI's impact on programmers. Previous studies have shown that tools like GitHub Copilot [87] are effective in generating foundational code and suggesting structures [246, 19], but their effectiveness varies based on the nature of queries and developers' ability to articulate their intent in natural language [262, 56]. Similar to prior observations [112], we also found that developers struggled to form a mental model of the underlying Large Language Models (LLMs) and to express their intentions accurately.

We also found that detailed, context-rich questions yield more useful responses from ChatGPT (e.g., by P1 and P5), but there are challenges when users lack sufficient context or the ability to validate responses. Selective code snippet implementation from ChatGPT underscores participants' cautious "test-and-develop" approach based on prior coding experiences. Initial evidence suggests that generative AI systems, like ChatGPT, in the AR development context, require reconsideration. Incorporating features like grounded utterances [155], converting vague queries into executable code, and step-by-step guides can address the contextual gap, providing educational tools for understanding the "what," "why," and "how" of coding problems. Allowing multimodal inputs such as images or videos could enhance user intent clarity, making AI more effective for those lacking specialized vocabulary. Future versions of generative AI with these features could offer targeted, adaptable, and educational support for programming tasks. Overall, our study highlights the importance of understanding the strengths and limitations of AI tools for effective use and training in domain-specific software development tasks.

5.5.5 Enhancing user engagement and learning through adaptive feedback and milestone integration

In the design of systems from an HCI perspective, system feedback and milestones are critical for user engagement and task completion [237]. Our study's findings underscore the importance of creating a progression model that not only facilitates the breaking down of complex tasks into manageable sub-tasks but also ensures that these micro-goals contribute to a holistic understanding of larger objectives of AR development. Our study indicates that breaking complex tasks into smaller, manageable sub-tasks is beneficial, particularly for the

"Pragmatic Developer" persona. However, this approach has a downside: while it boosts morale through incremental success, it can hinder a holistic understanding of overarching objectives. The challenge for HCI designers is to strike a balance between immediate, micro-level feedback and broader, macro-level insights that align with long-term goals, perhaps through a progress tracking dashboard that connects the dots between these two levels.

Furthermore, there is potential to design milestones that are both adaptive, catering to different user personas, and instructive, guiding users toward larger learning goals as they tackle more complex development tasks in AR. The challenge extends to helping users switch between bottom-up and top-down approaches when they hit learning challenges. HCI researchers could focus on developing systems capable of recognizing such learning gaps and offering guidance to navigate through them, thereby ensuring a more rounded educational experience.

5.5.6 Limitations

Although newcomers to AR can include a range of creators, such as hobbyists and domain experts with varying levels of programming expertise as demonstrated in my first study [15], in this study we focused on trained developers as our tasks required programming knowledge. Future work should consider end-user programmers who are not only new to AR but also new to programming to get a different perspective on their challenges. Furthermore, our method relied on in-lab task-based observations and interviews as this was an appropriate way to capture the emergent nature of AR development and identify key development issues in this rapidly evolving field. But, this may not capture the full range of experiences, challenges, and adaptive mechanisms that AR developers encounter in real-world settings. Longitudinal studies involving real-world tasks and larger projects could provide a more comprehensive view. Future work can also incorporate a control group of experienced AR developers for a more rigorous analysis. The current approach was taken because it offers a focused look at the challenges newcomers face, given that there is a rising tide of them entering the field, and existing literature has not yet sufficiently investigated this demographic. This focus provides foundational insights for educational initiatives aimed at supporting this growing community. Lastly, in this study we concentrated exclusively on AR development to ensure data consistency and methodological rigor; however, it is imperative for future research to broaden the scope to include other extended reality (XR) modalities like Virtual Reality (VR). This will provide a more nuanced and in-depth understanding of the challenges and learning processes associated with designing and developing activities in 3D environments.

5.6 Conclusions

In conclusion, our study demonstrates that software developers new to AR struggle in creating interactive immersive AR experiences, even when they are working within simplified development environments. They tend to rely heavily on mainstream information resources during the development process, but these resources are usually inadequate for navigating the intricacies of AR's spatial concepts, physics, and the hardware-software interplay inherent in AR development. The conventional information-seeking strategies approaches fall short in addressing the complex, 3D nature of AR. These findings highlight a pressing need for specialized AR training programs and online educational resources that focus on AR-specific problem decomposition rather than optimizing for code-level assistance. By incorporating diverse perspectives and experiences, training programs and resources can be designed to be more inclusive, thereby enriching the AR development ecosystem with a wider range of creative solutions and applications. By shedding light on the specific hurdles faced by newcomers, our work serves as a foundational step towards the creation of more targeted, user-centered learning aids that can better bridge the widening skills gap.

In the next chapter, I will step back to reflect on the broader implications and future work in the field of AR/VR development. This reflection will encompass a holistic view of the findings from previous chapters, discussing their impact on current practices and the potential they hold for shaping the future of AR/VR technologies for the new wave of the creators in this field. I will delve into how these insights can inform the design of more intuitive and inclusive development tools, foster a supportive community for creators, and address the evolving needs of an increasingly diverse creator demographic. Additionally, I will outline potential directions for future research.

Chapter 6

Reflections, implications, and future work

6.1 Key takeaways

In this dissertation, I explore the evolving landscape of AR/VR development, particularly focusing on how new creators, often self-taught or informal learners, navigate this multifaceted field. Since the start of this research in 2020, there has been a substantial increase in the guidelines provided by industry players such as Meta, Apple, and Unity. Despite the improved accessibility of these resources, the relevance of this research persists, as it offers deep insights into the unique learning trajectories and developmental strategies adopted by new AR/VR creators. This study not only highlights the critical learning paths and barriers these creators encounter but also contributes to the ongoing refinement of best practices and educational methodologies. Continuing this research approach is also crucial for comprehensively assessing the effectiveness of the emerging learning resources in AR/VR development. As the landscape of technological education evolves, it is imperative to evaluate how these tools and guidelines truly impact learner outcomes.

My work also considered the similarities and distinctive nature of AR/VR development compared to more traditional fields [131, 178]. It showed that AR/VR development stands apart as it focuses on creating immersive, interactive 3D environments, necessitating proficiency in multiple skills from prototyping and understanding of real-time user interactions to implementation and complex physics involved in building such experiences. Such interactions predominantly occur in software-driven environments and emphasize user experience aspects such as immersion and user interactions. Consequently, each development discipline not only requires a unique skill set but also a distinct approach to problem-solving and development, mirroring their different technological and user engagement priorities. On the other hand, we observed that there are similarities between AR/VR creation and traditional software development. We observed that both fields require conceptualization and requirement gathering, where developers define what needs to be created and outline the specifications. We also observed that, both AR/VR and traditional software development often follow an iterative development process. This includes stages such as designing, prototyping, and testing. In both cases, iterative refinements are essential to address issues and enhance the creation. In terms of challenges and complexities, both fields require developers to stay updated with the latest technological advancements and to continually upgrade their skills. However, AR/VR also demands a deeper understanding of user engagement in three-dimensional space, which is less prevalent in traditional software development.

In particular, key issues identified in the initial stage of this research reported in chapter 3 included difficulties in navigating the evolving AR/VR environment, specifically learning resources, understanding intricate toolchains, and applying user experience (UX) design principles while creating such immersive experiences. Through this phase of the research, a clear picture emerged of the common hurdles across different creator profiles in AR/VR development, emphasizing the differences among each individual new AR/VR creator and a need for more personalized entry points to the creation process, intuitive tools, and methodologies for AR/VR development.

Progressing from this foundational understanding, In chapter 4, I introduced PONI, which exemplifies a tailored approach to easing newcomers' entry into AR/VR development. In particular, evaluation of PONI underscored its effectiveness for AR/VR newcomers, deemed to be an intuitive, useful, and engaging platform compared to existing traditional keyword-based search methods for initiating the AR/VR learning and development process. This work showed a potential for the personalized onboarding approach to extend beyond AR/VR creation to other complex design tasks, like 3D modeling and software development, where comprehending the landscape and relevant terminologies is crucial before beginning the authoring process. This approach contributed to the field of personalized systems for learning and information-seeking, adding a human-centered perspective focus. These insights underscore the importance of considering more than just algorithmic effectiveness; emphasizing on factors like UI design, user control, transparency, and flexibility. These observations open up promising directions for future HCI research to further explore and enhance the design space of personalized onboarding.

Additionally, the empirical understanding gained from my formative observational study in Chapter 5, presented a nuanced understanding of how experienced software developers approach AR/VR development using a simplified development environment. My work examined how software developers navigate through the diverse array of AR/VR learning resources and tools, a process not previously observed in empirical detail. This study was significant in revealing how seasoned software developers, familiar with standard help resources, embark on their AR/VR learning journey, addressing a gap in our understanding of behavioral factors influencing this initiation. Additionally, this line of inquiry was instrumental in expanding our comprehension of how generative AI tools might reshape the strategies adopted by newcomers in navigating the AR/VR creation process where a new dimension is involved. The insights gained were not only pivotal in assessing the potential roles of AI in AR/VR development but also served to enrich the broader conversation in HCI about the integration of AI tools in learning and development within emerging technological domains.

In the concluding chapter, I will reflect on the broader implications and impacts of this research, considering potential enhancements to PONI's design and drawing connections to other pivotal areas of HCI research.

6.2 Implications and promising directions for future work

6.2.1 Introducing the immersive dimension into the creative tools

Our participants' experiences reported in chapter 3, as well as existing literature (e.g., [232, 182, 183]), indicate that designing 3D content on a 2D interface or screen can be often counter intuitive. However, our research findings reveal that most authoring tools currently used in practice still enforce this separation between the design and application spaces. In contrast, recent advancements in AR/VR authoring tool development, as illustrated by recent literature [147, 181], are adopting the "What You Experience is What You Get" (WYXIWYG) editor concept, originally proposed by Lee et al. [145, 146]. This concept is an immersive authoring approach that enables simultaneous content creation and validation within the application space itself.

The WYXIWYG approach, drawing from the "What You See Is What You Get" philosophy of modern graphical user interface editor tools, offers significant benefits in immersive authoring, particularly in defining spatial arrangements and behaviors [145]. With immersive authoring, designers can evaluate their creations in real time, eliminating the need to toggle between a 2D content creation space and a 3D application execution environment. This approach not only lowers the barriers to entry for novice AR/VR designers but also enhances the efficiency of AR/VR application developers who may not have programming skills. This aligns with findings in the fields of location-based experiences and ubiquitous computing, which emphasize the importance of in-situ authoring for effective ideation, reflection, and rearrangement of content [257].

Nevertheless, it is crucial to recognize that authoring immersive environments might not always be the ideal solution for every design challenge, particularly when dealing with abstract problems like programming logic [145]. In such cases, other approaches may be more suitable to address the specific needs of the design task at hand.

6.2.2 Adapting to rapid technological evolution in AR/VR development

In the dynamic field of AR/VR, where technologies and platforms are constantly evolving, it becomes crucial to develop learning resources that can adapt to these rapid changes. This need is particularly important given that version updates in AR/VR authoring tools and environments can significantly impact the development process. Newcomers and experienced creators alike often find that updates introduce changes that necessitate relearning or adjusting their workflows, which can be seen as a setback. An intriguing area of future research, inspired by my findings in the PONI study, would be to investigate automated methods for updating learning resources in tandem with these technological advancements.

The goal would be to not only maintain the relevance of educational content but also to alleviate the learning challenges posed by new versions of AR/VR tools. For example, when a major update is released for a popular AR/VR development platform, the learning resources, including tutorials and guides, could be automatically adjusted to reflect the new features and altered workflows. This could involve using AI algorithms to parse through changes in software documentation and update instructional materials accordingly.

Such an approach would significantly reduce the burden on users who need to adapt to new versions, providing them with up-to-date resources that align with the latest tool capabilities and best practices. This would be particularly beneficial in maintaining the continuity and effectiveness of personalized learning paths, as established in PONI, ensuring that learners are always working with the most current information and tools. By focusing on this aspect, future research can contribute to creating a more resilient and adaptive learning ecosystem in AR/VR development, one that is better equipped to handle the rapid pace of technological change in this field.

6.2.3 Extending personalizations for power users in AR/VR development

Reflecting on my findings from chapter 4, it is evident that while PONI excels in guiding new AR/VR creators with personalized learning paths, there is an opportunity to extend its functionality to better support power users – those with significant experience in AR/VR development. Power users often seek advanced, intricate features and deeper control over their learning and development processes [135]. For example, where a beginner might benefit from a guided module on simple VR interactions, a power user might be looking to delve into complex physics simulations or advanced spatial audio techniques in VR environments.

To cater to these needs within the PONI framework, a practical extension could involve introducing an advanced module selection feature. This feature would allow power users to customize their learning path by selecting from a range of advanced modules focused on high-level AR/VR development concepts. For instance, a module on advanced lighting techniques in VR or optimization strategies for AR applications would be directly relevant to these users. Additionally, PONI could offer project templates that include more complex scenarios and coding challenges, enabling power users to experiment with and refine sophisticated development techniques.

Another realistic extension would be the incorporation of a feedback-based adaptive learning system within PONI. This system could analyze the progress and engagement patterns of power users and suggest even more challenging projects or learning materials based on their interactions. For example, if a user demonstrates proficiency in basic AR development, PONI could recommend a project involving advanced AR features like real-time object recognition or interactive augmented environments. This adaptive approach ensures that the learning content remains challenging and relevant, thereby keeping power users engaged and continuously pushing the boundaries of their skills in AR/VR development.

6.2.4 Extending the concept of PONI to community-driven AR/VR development

Building upon PONI's success in personalizing the learning experience for AR/VR creators, the next step could involve developing a community-centric extension of PONI. This would be a platform where PONI's personalization algorithms not only recommend individualized learning materials but also connect learners with expert mentors and peer collaborators. For instance, based on a newcomer's skill level and interests identified by PONI, they could be paired with experienced developers who have complementary skills or similar project histories. This mentorship could be facilitated through integrated communication tools within the platform, allowing for seamless knowledge exchange and collaboration.

Additionally, this extended platform could host community-driven AR/VR projects, where users contribute to different aspects of a project based on their skill level and learning goals, as identified by PONI. This approach not only provides practical, hands-on experience but also creates a sense of community and shared learning [88, 92]. The platform could feature project showcases, where users can see the direct impact of their contributions and learn from the development processes of others [84].

Lastly, in chapter 4 we saw that some users with stronger skill sets in 3D modeling or programming strived for more challenging and advanced example projects to explore. Similar observations were reported in a study on development practices of professional AR/VR developers [135], as power creators of AR/VR experiences benefit from having access to experience beyond their skillset. It is crucial to enable creators to develop artifacts that are not only efficient and effective but also straightforward, goal-focused, and adaptable for various interdisciplinary roles [111]. Investigating how rewards might encourage professional AR/VR creators to share their work could be an interesting area for future research. This could involve implementing reputation-based incentives akin to those found in prominent online Q&A platforms such as StackExchange. There is emerging research on how approaches like gamification [249] and ranking systems for contributors [258] can inspire users to disseminate their knowledge in digital communities. PONI might serve as an starting point for studies aimed at creating successful knowledge-sharing communities with a focus on altruism among creators.

6.2.5 Personalizing AR/VR authoring for different domains

An intriguing direction to explore is the personalization of AR/VR authoring processes to suit the specific requirements of various domains and audiences. Given the expanding use of AR/VR in numerous fields, it is crucial to determine how a tool like PONI can be adapted to effectively meet the distinct needs of different user groups. As highlighted in chapters 3 and 4, these groups might include domain experts such as educators, scientists, subject-matter experts, UI/UX designers, and more. In our studies, we encountered biomedical engineers who desired features specific to human rehabilitation practices in VR or content developers who wanted more storytelling content integrated into the software. Attempting to accommodate all these domains in a single software could lead to an overload of features and assets. Each domain has unique requirements, workflows, and communication styles, possibly necessitating specialized features or interfaces within an AR/VR authoring tool. Traditional advanced tools for AR/VR (such as Unity or Unreal Engine) have tried to cater to various domains by incorporating all necessary features into the same interface. While this increases the tool's extensibility, it also adds complexity.

We suggest adopting a modular approach, tailoring specific features to specific users. Educators, for example, may focus on creating engaging and comprehensible educational content. They could benefit from simplified design workflows, a range of educational graphics, and features enabling easy integration of text and spoken explanations. Scientists and subject-matter experts, dealing with complex data and concepts, might need customized interactions and the ability to create custom graphical assets that accurately represent their theories. UI/UX designers would likely prioritize developing engaging, interactive digital experiences, requiring features for prototyping, animation controls for interactive design, and integration with UI design tools. By carefully addressing these individual needs, we can start to define a development roadmap [96, 134] for domain-specific functionalities in AR/VR authoring tools, thereby improving their utility and usability for a broader range of users.

6.3 Limitations of the research methodology

This thesis has advanced our knowledge in understanding the process and challenges new creators face when creating AR/VR experiences. However, it does have its limitations, and the contributions are offered cautiously. My research highlighted the significant presence of Unity as a prevalent creation tool, with limited similar alternatives as per study participants' feedback in chapter 3. In the context of AR/VR, there is an opportunity for future research to conduct detailed case studies that compare various AR/VR software tools or even implement controlled studies to assess different tool interventions and approaches.

Regarding our implementation of PONI in chapter 4 as a web-based proof-of-concept application, further research is necessary to understand how users interact with such tools in actual learning tasks, possibly through a longitudinal field study. The criteria used for curating projects in our database may not cover all factors, and it's vital to continuously adapt to the evolving needs of AR/VR creators. A potential extension for expanding the database is using crowd-sourcing methods, proven effective in other learning contexts [259]. Furthermore, our current scoring function for ranking results only considers three levels for proximity and importance; future work should investigate more nuanced levels to enhance the precision and effectiveness of retrieval in complex onboarding scenarios. Similarly, the methodology used in chapter 5, based on in-lab observations and interviews, was suitable for capturing the emergent nature of AR development using a simplified development environment and identifying key issues in this rapidly evolving field. However, it might not fully represent the range of experiences and challenges AR developers face in real-world settings. Longitudinal studies involving larger projects and real-world tasks could offer a more comprehensive understanding.

Notably, the impact of gender was not studied, which may represent an area for future inquiry. In addition, conducting research in the context of the COVID-19 pandemic posed specific challenges that may have influenced the limitations of this thesis, particularly in the field of AR/VR. The restrictions on in-person interactions necessitated a greater dependence on digital platforms for data collection, which might have constrained the depth of insights, diminished spontaneity, and introduced biases in the collected data. The inability to physically access resources and research participants likely affected the diversity in the studies. Furthermore, the global transition to remote work and the resulting changes in AR/VR software usage patterns could have impacted the interpretation and relevance of our findings. The psychological stress and uncertainty of the pandemic might have influenced the responses of participants. Future studies should consider these factors, potentially reevaluating the research design to compensate for these potential biases and influences.

Finally, although my research has revealed the promise of a novel personalized learning tool for the newcomers to AR/VR creation, a more extensive and widespread implementation might be required to fully assess the tool's effect on the performance of professional AR/VR creators.

Chapter 7

Conclusion

The primary objective of this dissertation was to characterize the emerging group of AR/VR creators as a significant segment of end-user programmers, utilizing these technologies as a novel medium for realizing their creative visions. While the current excitement around AR/VR predominantly involves professional developers with access to advanced tools like Unity or Unreal, my findings indicate that hobbyists, domain experts, and designers as different groups of new AR/VR creators have distinct requirements for prototyping, implementing, debugging, and testing in AR/VR applications. Recognizing this group of creators, this work serves as an initial step for supporting them with tailored help. My research aimed to identify and assess innovative methods to support these individuals in their initial attempts entering this complex field.

Through foundational interview-based (chapter 3) and observational empirical studies (chapter 5), I revealed that the challenges AR/VR creators encounter during implementation, debugging, and testing are intensified by the necessity to navigate numerous unknowns and keep pace with the rapidly evolving AR/VR hardware and software. A key observation was that domain experts and hobbyists often struggle to determine where to begin, frequently relying on available learning resources mostly targeted to professional creators and ad-hoc social recommendations for choosing authoring environments. This approach can lead to the selection of tools and help materials that, while suitable for the project, may not align with their experience level, resulting in various issues during implementation and testing phases.

Building on this foundation, my development and evaluation of PONI (chapter 4) showcased how a personalized approach can significantly assist new creators in the AR/VR field. By mitigating concerns related to handling numerous unknowns, PONI showed potential in empowering these creators with a practical tool for gradually learning about the AR/VR landscape. This approach facilitates the development of an intuitive understanding of this complex and multifaceted domain, enabling new creators to effectively navigate the space before delving into more general help resources and communities predominantly aimed at professional creators. My usability study of PONI particularly highlighted its effectiveness as a "hub" for inspiration before the self-directed exploratory learning phase, demonstrating its value in fostering a more approachable entry into AR/VR creation.

Finally, while much research on personalized information retrieval has concentrated on optimizing underlying algorithms, there has been limited emphasis on the interaction design of these systems and how they shape users' perceptions. This work underscores the importance of observing users interacting with such systems and capturing their views on the effectiveness and utility of personalized results. In chapter 4, I demonstrated that perceived usefulness from the vision of new AR/VR creators is influenced by more than just the algorithm's effectiveness; factors such as UI design, user control, transparency, and flexibility play crucial roles in shaping the overall user experience.

In this chapter, I conclude this dissertation by reflecting on my research contributions and how they can support further advances both in the design of educational technology for the authoring process of new AR/VR creators and more general classes of software.

7.1 Core research contributions

7.1.1 Empirical insights into how new AR/VR creators approach development process and their unique difficulties

A key contribution made in this work was by characterizing the categories of new creators in the AR/VR authoring space, delving into their motivations, needs, challenges, and workarounds. This work offered a comprehensive understanding of the varying approaches new AR/VR creators employ in designing, implementing, and testing AR/VR applications.

One of the most important findings from my study is the diverse motivations driving these creators ranging from pursuing creative expression and innovation to solving practical problems and exploring new technological frontiers. This diversity highlighted the need for AR/VR tools and resources that cater to a wide spectrum of aspirations and expertise levels. Additionally, I uncovered specific needs and challenges faced by AR/VR creators, including hurdles with creating the physical aspects of AR/VR, resource limitations, and the steep learning curve associated with AR/VR development. I documented various strategies and workarounds employed by creators to overcome these challenges, offering valuable insights into their adaptive and creative problem-solving approaches.

Furthermore, my research sheds light on the practical aspects of AR/VR creation, including how creators conceptualize their ideas, navigate the iterative development process, and utilize different tools and techniques to bring their visions to life. These insights are particularly valuable as they not only illustrate the current state of AR/VR creation but also inform the development of more user-centric AR/VR tools and methodologies.

These insights ultimately formed the basis for the design goals underling PONI and several of its key features. In the long term, these findings are well-positioned to provide a solid foundation for further design research or advances in educational technology that aim to streamline the AR/VR creation onboarding process for new creators in this field.

7.1.2 Empirical insights into how new AR/VR creators approach help-seeking and their unique challenges

An overarching theme in my work has been to understand how to assist with the information-seeking process for new AR/VR creators. In my formative study detailed in Chapter 3, I gathered rich insights about the varied practices and pain points of new AR/VR creators. This study particularly highlighted their reliance on trial and error while learning AR/VR development skills and their inclination to learn from examples of available projects as a reliable starting point where possible.

However, the study also brought to light the challenges new AR/VR creators face in articulating their intents before starting a project. They often struggle to determine the suitability of a project, development framework, or device, considering their skillsets, available tools, time budget, and specific goals. Such insights demonstrate the need for more intuitive and accessible AR/VR authoring tools and resources that can better guide creators through the initial stages of development, bridging the gap between their current capabilities and the demands of AR/VR technology.

These insights have been instrumental in forming the design goals for PONI and several of its key features. Over the long term, my findings provide a solid foundation for further design research and advances in educational technology. The aim is to streamline the process of finding relevant and personalized learning resources for new creators of AR/VR experiences, enabling a more efficient and effective onboarding experience for newcomers to this dynamic field.

7.1.3 Design of PONI as a novel personalized learning platform for locating relevant AR/VR learning resources

Another significant contribution of this dissertation is the design and implementation of the PONI platform for for getting inspiration and learning about AR/VR creation. Through successive iterations of refinement and interviews with AR/VR newcomers, I derived five main design goals:

> **DG1:** Locate targeted learning materials given the creator's background and skills.
>
> **DG2:** Locate targeted learning materials given the desired creation outcomes and constraints.
>
> **DG3:** Get inspiration and browse relevant example projects.

DG4: Assess relevance of learning materials in relation to the creator's background and desired goals.

DG5: Offer freedom and flexibility in personalizing and exploring relevant learning materials.

I utilized these goals to guide the design of PONI, ensuring it supported the use cases identified as most critical to newcomers in the AR/VR field in locating effective and relevant learning resources for starting their AR/VR learning and development journey. Specifically, PONI was designed to provide a means for discovering learning examples that are personalized to the unique needs and specifications of new AR/VR creators. This included providing rationale for why certain examples were suitable starting points for the creation process. Through my usability study, I gathered evidence indicating that PONI's design had potential to serve as a knowledge hub, aiding in the initial stages of the AR/VR authoring process, offering inspiration, and assisting in learning the necessary terminology. Although this idea has been touched upon in some past learning platforms, to my knowledge this is the first time personalizing example projects has been designed and evaluated.

While my implementation of PONI was specifically aimed at supporting the AR/VR authoring process, its design concept has broader applicability. Given the known struggles of learners in complex skill areas in locating relevant learning materials, as highlighted in [123, 76], the principles underpinning PONI's design could be generalized to support learning and development in other domains. For instance, in animation, where a variety of skills such as 3D modeling and video editing are required, and numerous feature-rich software programs are utilized, PONI's approach could be particularly beneficial.

7.1.4 Empirical insights into how software developers new to AR approach the development process and their use of AI-assisted tools

My research provides behavioral insights into the initial approaches of new software developers as they use a simplified development environment for creating AR experiences. My study sheds light on the specific information resources developers seek, and the distinctive nature of their learning experiences in AR compared to conventional software development. In chapter 5, I particularly highlight the distinct requirements of emerging AR developers, noting that code reuse strategies effective in other development areas may not be as successful in AR due to the unfamiliar vocabularies and complexities associated with 3D environments, physics, and motion in AR development. These findings build upon and broaden previous studies on elective, task-focused learning in mainstream software development, suggesting that the mindset geared towards "immediate solutions" may not adequately address the intricacies of AR development.

Additionally, my work was the first to my knowledge, that explored the use of ChatGPT as a learning aid among new creators of AR/VR applications. I reported results that underscore the platform's nuanced role in the implementations process of such immersive experiences. My work provided initial insights reporting that detailed and context-rich questions to ChatGPT and gradually adapting prompts, could potentially result in better assistance from AI assisted platforms. In contrast, less contextualized questions can result in less helpful and relevant responses from such platforms. This variability highlights the importance of user knowledge in validating the answers provided by AI platforms and the need for caution in implementing code snippets suggested by ChatGPT.

These findings indicate that the design and functionality of generative AI systems like ChatGPT in the AR development context require further consideration. Integrating such systems to track the creation progress similar to prior work [203, 107, 86, 244], or enabling multimodal inputs like images or videos could help clarify user intent and make the AI more effective for those lacking specialized vocabulary or seeking additional context. The evolution of generative AI to include such features could lead to more targeted, adaptable, and educational support for programming tasks in AR development and beyond.

Considering the significant need for specialized training and resources in AR development, it's probable that many informal learners creating AR/VR experiences will be self-taught, relying on online resources. Based on my studies in chapter 3 and 5, these new AR/VR developers are likely to encounter similar challenges as our study participants. Reflecting on these findings, I envision an imperative to reevaluate the design of training programs and learning tools to more effectively support the burgeoning community of informal AR/VR developers. Hence, I propose enhancing AR development by incorporating problem-solving strategies, utilizing in-context personalized approaches and generative-AI tools, and boosting user engagement through adaptive feedback and milestone integration in the learning process.

7.2 Secondary contributions

7.2.1 Empirical insights into challenges in designing physical aspects of AR/VR

My research, specifically in chapter 3 shed light on several nuanced aspects of designing "immersive" AR/VR experiences. One key challenge identified was in creating realistic and engaging experiences, which is often hindered by difficulties in simulating models and gestures, as well as the physical aspects of design such as user posture and fatigue. This challenge extends to AR, where maintaining user attention in an uncontrolled physical environment is particularly complex. Additionally, the use of techniques like 360° story boarding and role-playing for conceptualizing immersive experiences while helpful, was found limited in their ability to accurately portray user interactions in the unlimited nature of immersive experiences.

Another significant challenge highlighted in my study in chapter 5 was the intricacies of coding and simulating motion in 3D environments, especially within AR development. New creators with prior programming skills often attributed issues to coding mistakes or platform inefficiencies. This underlines a critical gap in existing learning materials, particularly regarding the physical aspects of debugging processes in AR/VR development. Moreover, the challenges in predicting object behavior and the application of traditional 2D programming methods in 3D environments underscore the need for more contextually relevant learning resources and tools.

These findings collectively offer a holistic view of the unique challenges faced by new creators in AR/VR environments regarding a new dimension that is involved as compared to 2D applications. This understanding is pivotal for the development of more effective and intuitive AR/VR authoring tools and resources, guiding new creators through the complexities of immersive design and development.

7.3 Closing remarks

This dissertation delves into various aspects of how new creators from diverse backgrounds and skill levels engage in the process of seeking information, designing, implementing, and testing AR/VR applications. I have delineated the motivations, needs, and challenges of such newcomers, offering a detailed understanding of a wide array of creators who are all beginners in this domain. While it is recognized that programming skills are advantageous in leveraging feature-rich tools such as Unity and Unreal for realizing creative visions in AR/VR, my research uncovers additional crucial elements. These include the importance of problem-solving strategies, the use of in-context personalized approaches, and context-appropriate help tailored to AR/VR modalities, all of which are vital in supporting new creators.

Moreover, the insights gained from my formative research have informed specific design goals, leading to the development, implementation, and evaluation of PONI (Personalized Onboarding Interface). PONI is innovatively designed to aid in acquiring inspiration and learning about AR/VR creation. The features and design of PONI represent a promising step forward in assisting new AR/VR creators to have a safer starting point for learning and locating resources, fostering an environment that encourages innovation and growth in this emerging field. I believe that the designs and insights derived from this work are well-suited to be foundational for future endeavors by HCI researchers and practitioners to support AR/VR creators, as they navigate the increasingly complex landscape of AR/VR creation.

Bibliography

[1] Cenario vr: Develop vr content with ease and speed. https://www.cenariovr.com/, 2022.

[2] Cospaces edu: Kid-friendly 3d creation and coding. https://cospaces.io/edu/, 2022.

[3] Vizor: Create 360 tours, stories, sites and webvr experiences. https://site.vizor.io/, 2022.

[4] Gregory D. Abowd. Software engineering issues for ubiquitous computing. In *Proceedings of the 21st International Conference on Software Engineering*, ICSE '99, page 75–84, New York, NY, USA, 1999. Association for Computing Machinery.

[5] Adobe. *Adobe Aero*. https://www.adobe.com/ca/products/aero.html, Accessed: 2023-11-11.

[6] David Akers, Matthew Simpson, Robin Jeffries, and Terry Winograd. Undo and erase events as indicators of usability problems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 659–668, New York, NY, USA, 2009. Association for Computing Machinery.

[7] Vicki H. Allan and Mary V. Kolesar. Teaching computer science: A problem solving approach that works. *SIGCUE Outlook*, 25(1–2):2–10, jan 1997.

[8] Apostolos Ampatzoglou and Ioannis Stamelos. Software engineering research for computer games: A systematic review. *Information and Software Technology*, 52(9):888–901, September 2010.

[9] John R. Anderson, Albert T. Corbett, Kenneth R. Koedinger, and Ray. Pelletier. Cognitive tutors: Lessons learned. *Journal of the Learning Sciences*, 4(2):167–207, April 1995.

[10] Oscar D. Andrade, Nathaniel Bean, and David G. Novick. The macro-structure of use of help. In *Proceedings of the 27th ACM International Conference on Design of Communication*, SIGDOC '09, page 143–150, New York, NY, USA, 2009. Association for Computing Machinery.

[11] Apple. *ARKit*. https://developer.apple.com/augmented-reality/, Accessed: 2023-11-11.

[12] Apple. *Reality Composer*. https://developer.apple.com/augmented-reality/tools/, Accessed: 2023-11-11.

[13] Apple. *Apple Announces More Than 600 New Apps Built for Apple Vision Pro.* https://www.apple.com/newsroom/2024/02/apple-announces-more-than-600-new-apps-built-for-apple-vision-pro/, Accessed: 2024.

[14] Narges Ashtari, Parsa Alamzadeh, Gayatri Ganapathy, and Parmit Chilana. Poni: A personalized onboarding interface for getting inspiration and learning about ar/vr creation. In *Nordic Human-Computer Interaction Conference*, NordiCHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[15] 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, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[16] Narges Ashtari and Parmit K. Chilana. How new developers approach augmented reality development using simplified creation tools: An observational study. *Multimodal Technologies and Interaction*, 8(4), 2024.

[17] Marko Balabanović and Yoav Shoham. Fab: content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72, mar 1997.

[18] Nikola Banovic, Tovi Grossman, Justin Matejka, and George Fitzmaurice. *Waken: Reverse Engineering Usage Information and Interface Structure from Software Videos*, page 83–92. Association for Computing Machinery, New York, NY, USA, 2012.

[19] Shraddha Barke, Michael B. James, and Nadia Polikarpova. Grounded copilot: How programmers interact with code-generating models, 2022.

[20] Richard Bentley and Paul Dourish. Medium versus mechanism: Supporting collaboration through customisation. In *ECSCW*, 1995.

[21] Suresh K. Bhavnani and Bonnie E. John. Delegation and circumvention: Two faces of efficiency. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '98, page 273–280, USA, 1998. ACM Press/Addison-Wesley Publishing Co.

[22] Mark Billinghurst. Rapid prototyping for ar/vr experiences. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–4, New York, NY, USA, 2020. Association for Computing Machinery.

[23] Mark Billinghurst. Grand challenges for augmented reality. *Frontiers in Virtual Reality*, 2:578080, 03 2021.

[24] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. *Trends in Human-Computer Interaction*, 8(2–3):73–272, mar 2015.

[25] Unity Blog. *Creative professionals are struggling to implement augmented reality.* https://blog.unity.com/games/creative-professionals-are-struggling-to-implement-augmented-reality, 2019.

[26] Benjamin S Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16, 1984.

[27] Barry Boehm, Bradford Clark, Ellis Horowitz, Chris Westland, Ray Madachy, and Richard Selby. Cost models for future software life cycle processes: COCOMO 2.0. *Annals of Software Engineering*, 1(1):57–94, December 1995.

[28] Paul De Bra, David Smits, Kees van der Sluijs, Alexandra I. Cristea, Jonathan Foss, Christian Glahn, and Christina M. Steiner. GRAPPLE: Learning management systems meet adaptive learning environments. In *Intelligent and Adaptive Educational-Learning Systems*, pages 133–160. Springer Berlin Heidelberg, 2013.

[29] Joel Brandt, Mira Dontcheva, Marcos Weskamp, and Scott R. Klemmer. Example-centric programming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, April 2010.

[30] Joel Brandt, Philip J. Guo, Joel Lewenstein, Mira Dontcheva, and Scott R. Klemmer. Two studies of opportunistic programming: Interleaving web foraging, learning, and writing code. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 1589–1598, New York, NY, USA, 2009. Association for Computing Machinery.

[31] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 43–52, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[32] Matthew Brehmer, Joanna McGrenere, Charlotte Tang, and Claudia Jacova. Investigating interruptions in the context of computerised cognitive testing for older adults. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2649–2658, New York, NY, USA, 2012. Association for Computing Machinery.

[33] John Bucher. *Storytelling for Virtual Reality: Methods and Principles for Crafting Immersive Narratives*. Routledge, July 2017.

[34] Andrea Bunt, Cristina Conati, and Joanna McGrenere. What role can adaptive support play in an adaptable system? In *Proceedings of the 9th international conference on Intelligent user interfaces*, IUI-CADUI04. ACM, January 2004.

[35] Andrea Bunt, Patrick Dubois, Ben Lafreniere, Michael A. Terry, and David T. Cormack. Taggedcomments: Promoting and integrating user comments in online application tutorials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 4037–4046, New York, NY, USA, 2014. Association for Computing Machinery.

[36] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[37] Robin Burke. Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408. Springer Berlin Heidelberg, 2007.

[38] Margaret Burnett, Curtis Cook, and Gregg Rothermel. End-user software engineering. *Communications of the ACM*, 47(9):53–58, September 2004.

[39] Mark J. Carman, Mark Baillie, and Fabio Crestani. Tag data and personalized information retrieval. In *Proceedings of the 2008 ACM Workshop on Search in Social Media*, SSM '08, page 27–34, New York, NY, USA, 2008. Association for Computing Machinery.

[40] John M. Carroll. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. MIT Press, Cambridge, MA, USA, 1987.

[41] John M. Carroll. *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. Technical communication and information systems. MIT Press, 1990.

[42] Pablo Castells, Miriam Fernández, David Vallet, Phivos Mylonas, and Yannis Avrithis. Self-tuning personalized information retrieval in an ontology-based framework. In *Proceedings of the 2005 OTM Confederated International Conference on On the Move to Meaningful Internet Systems*, OTM'05, page 977–986, Berlin, Heidelberg, 2005. Springer-Verlag.

[43] Yan Chen, Sang Won Lee, Yin Xie, YiWei Yang, Walter S. Lasecki, and Steve Oney. Codeon: On-demand software development assistance. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 6220–6231, New York, NY, USA, 2017. Association for Computing Machinery.

[44] Pei-Yu Chi, Sally Ahn, Amanda Ren, Mira Dontcheva, Wilmot Li, and Björn Hartmann. *MixT: Automatic Generation of Step-by-Step Mixed Media Tutorials*, page 93–102. Association for Computing Machinery, New York, NY, USA, 2012.

[45] Parmit K. Chilana, Celena Alcock, Shruti Dembla, Anson Ho, Ada Hurst, Brett Armstrong, and Philip J. Guo. Perceptions of non-cs majors in intro programming: The rise of the conversational programmer. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 251–259, 2015.

[46] Parmit K. Chilana, Amy J Ko, and Jacob O Wobbrock. Lemonaid: selection-based crowdsourced contextual help for web applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1549–1558, 2012.

[47] Parmit K. Chilana, Rishabh Singh, and Philip J. Guo. Understanding conversational programmers: A perspective from the software industry. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 1462–1472, New York, NY, USA, 2016. Association for Computing Machinery.

[48] Parmit K. Chilana, Jacob O. Wobbrock, and Amy J. Ko. Understanding usability practices in complex domains. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 2337–2346, New York, NY, USA, 2010. Association for Computing Machinery.

[49] Steven J. Clarke. What is an end user software engineer? In *End-User Software Engineering*, 2007.

[50] Cristina Conati. Modeling and scaffolding self-explanation across domains and activities. In *International Handbook of Metacognition and Learning Technologies*, pages 367–383. Springer New York, 2013.

[51] Cristina Conati. Commentary on: "toward computer-based support of MetaCognitive skills: a computational framework to coach self explanation". *International Journal of Artificial Intelligence in Education*, 26(1):183–192, November 2015.

[52] Juliet Corbin and Anselm Strauss. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications, 2014.

[53] Juliet M. Corbin and Anselm Strauss. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, 13(1):3–21, 1990.

[54] Diana I. Cordova and Mark R. Lepper. Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology*, 88(4):715–730, 1996.

[55] Kathryn Cunningham, Barbara J. Ericson, Rahul Agrawal Bejarano, and Mark Guzdial. Avoiding the turing tarpit: Learning conversational programming by starting from code's purpose. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[56] Arghavan Moradi Dakhel, Vahid Majdinasab, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, Zhen Ming, and Jiang. Github copilot ai pair programmer: Asset or liability?, 2023.

[57] Arindam Dey, Mark Billinghurst, Robert W. Lindeman, and J. Edward Swan. A systematic review of 10 years of augmented reality usability studies: 2005 to 2014. *Frontiers in Robotics and AI*, 5, April 2018.

[58] Griffin Dietz, James A. Landay, and Hyowon Gweon. Building blocks of computational thinking: Young children's developing capacities for problem decomposition. In *Annual Meeting of the Cognitive Science Society*, 2019.

[59] Andrew Dillon and Charles Watson. User analysis in hci—the historical lessons from individual differences research. *International Journal of Human-Computer Studies*, 45(6):619–637, dec 1996.

[60] Brian Dorn and Mark Guzdial. Learning on the job: Characterizing the programming knowledge and learning strategies of web designers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 703–712, New York, NY, USA, 2010. Association for Computing Machinery.

[61] Jon Dron. *Control and Constraint in E-Learning*. IGI Global, 2007.

[62] Patrick Dubois, Volodymyr Dziubak, and Andrea Bunt. Tell me more! soliciting reader contributions to software tutorials. In *Proceedings of the 43rd Graphics Interface Conference*, GI '17, page 16–23, Waterloo, CAN, 2017. Canadian Human-Computer Communications Society.

[63] Charles Dziuban, Colm Howlin, Patsy Moskal, Connie Johnson, Liza Parker, and Maria Campbell. Adaptive learning: A stabilizing influence across disciplines and universities. *Online Learning*, 22(3), September 2018.

[64] Sarah D'Angelo and Andrew Begel. Improving communication between pair programmers using shared gaze awareness. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17. ACM, May 2017.

[65] Dennis E. Egan. Individual differences in human-computer interaction. In *Handbook of Human-Computer Interaction*, pages 543–568. Elsevier, 1988.

[66] Dennis E Egan and Louis M Gomez. Assaying, isolating, and accommodating individual differences in learning a complex skill. *Individual differences in cognition*, 2:174–217, 1985.

[67] Michael Ekstrand, Wei Li, Tovi Grossman, Justin Matejka, and George Fitzmaurice. Searching for software learning resources using application context. In *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. ACM Press, 2011.

[68] Tristan C. Endsley, Kelly A. Sprehn, Ryan M. Brill, Kimberly J. Ryan, Emily C. Vincent, and James M. Martin. Augmented reality design heuristics: Designing for dynamic interactions. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 61(1):2100–2104, September 2017.

[69] Eileen Faulkenberry and Thomas Faulkenberry. Constructivism in mathematics education: A historical and personal perspective. *The Texas Science Teacher*, 35:17–21, 01 2006.

[70] Leah Findlater and Joanna McGrenere. Evaluating reduced-functionality interfaces according to feature findability and awareness. In *Lecture Notes in Computer Science*, pages 592–605. Springer Berlin Heidelberg, 2007.

[71] Leah Findlater and Joanna McGrenere. Beyond performance: Feature awareness in personalized interfaces. *International Journal of Human-Computer Studies*, 68(3):121–137, March 2010.

[72] Leah Findlater and Jacob O. Wobbrock. Personalized input: improving ten-finger touchscreen typing through automatic adaptation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012.

[73] Scott D. Fleming, Chris Scaffidi, David Piorkowski, Margaret Burnett, Rachel Bellamy, Joseph Lawrance, and Irwin Kwan. An information foraging theory perspective on tools for debugging, refactoring, and reuse tasks. *ACM Transactions on Software Engineering and Methodology*, 22(2):1–41, March 2013.

[74] Adam Fourney, Ben Lafreniere, Parmit K. Chilana, and Michael Terry. Intertwine: Creating interapplication information scent to support coordinated use of software. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 429–438, New York, NY, USA, 2014. Association for Computing Machinery.

[75] Adam Fourney, Ben Lafreniere, Richard Mann, and Michael A. Terry. “Then click ok!”: extracting references to interface elements in online documentation. In *Proceedings of the 30th ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 35–38. ACM, 2012.

[76] C. Ailie Fraser, Tricia J. Ngoon, Mira Dontcheva, and Scott Klemmer. Replay: Contextually presenting learning videos across software applications. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI ’19, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.

[77] Wolfgang Friedrich. Arvika " augmented reality for development, production and service. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality*, ISMAR ’02, page 3, USA, 2002. IEEE Computer Society.

[78] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, November 1987.

[79] Krzysztof Gajos, Katharina Reinecke, and Charles Herrmann. Accurate measurements of pointing performance from in situ observations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’12, page 3157–3166, New York, NY, USA, 2012. Association for Computing Machinery.

[80] Krzysztof Z Gajos, Katherine Everitt, Desney S. Tan, Mary Czerwinski, and Daniel S. Weld. Predictability and accuracy in adaptive user interfaces. In *CHI*, 2008.

[81] Sharon Nelson-Le Gall. Chapter 2: Help-seeking behavior in learning. *Review of Research in Education*, 12:55–90, 1985.

[82] Epic Games. *Unreal Engine*. `https://www.unrealengine.com/`, Accessed: 2023-11-11.

[83] Whimsy Games. *Godot vs Unity: All You Need to Know*. `https://whimsygames.co/blog/godot-vs-unity-all-you-need-to-know/`, 2021.

[84] Maribeth Gandy and Blair MacIntyre. Designer’s augmented reality toolkit, ten years later: implications for new media authoring tools. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, UIST ’14. ACM, October 2014.

[85] Fabio Gasparetti, Carla Limongelli, and Filippo Sciarrone. Exploiting wikipedia for discovering prerequisite relationships among learning objects. In *2015 International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–6, 2015.

[86] Arin Ghazarian and S. Majid Noorhosseini. Automatic detection of users’ skill levels using high-frequency user interface events. *User Modeling and User-Adapted Interaction*, 20(2):109–146, February 2010.

[87] GitHub and OpenAI. Github copilot, 2023.

[88] Max Goldman, Greg Little, and Robert C. Miller. Collabode: Collaborative coding in the browser. In *Proceedings of the 4th International Workshop on Cooperative and Human Aspects of Software Engineering*, CHASE '11, page 65–68, New York, NY, USA, 2011. Association for Computing Machinery.

[89] Nathaniel Good, J. Ben Schafer, Joseph A. Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, and John Riedl. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, page 439–446, USA, 1999. American Association for Artificial Intelligence.

[90] Google. *ARCore*. `https://developers.google.com/ar`, Accessed: 2023-11-11.

[91] Saul Greenberg. Using unix: Collected traces of 168 users. 1988.

[92] Saul Greenberg and Chester Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST01. ACM, November 2001.

[93] Tovi Grossman and George Fitzmaurice. Toolclips: An investigation of contextual video assistance for functionality understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 1515–1524, New York, NY, USA, 2010. Association for Computing Machinery.

[94] Tovi Grossman and George Fitzmaurice. An investigation of metrics for the in situ detection of software expertise. *Human-Computer Interaction*, 30(1):64–102, jan 2015.

[95] Tovi Grossman, George Fitzmaurice, and Ramtin Attar. A survey of software learnability: Metrics, methodologies and guidelines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 649–658, New York, NY, USA, 2009. Association for Computing Machinery.

[96] 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*, UIST '10, page 143–152, New York, NY, USA, 2010. Association for Computing Machinery.

[97] Jan Gulliksen and Bengt Sandblad. Domain-specific design of user interfaces. *International Journal of Human-Computer Interaction*, 7(2):135–151, April 1995.

[98] 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 *2010 International Symposium on Ubiquitous Virtual Reality*, pages 40–43, 2010.

[99] 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*, OZCHI '06, page 409–412, New York, NY, USA, 2006. Association for Computing Machinery.

[100] Mona Haraty, Joanna McGrenere, and Andrea Bunt. Online customization sharing ecosystems: Components, roles, and motivations. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '17, page 2359–2371, New York, NY, USA, 2017. Association for Computing Machinery.

[101] Frederick Hayes-Roth. Rule-based systems. *Commun. ACM*, 28(9):921–932, sep 1985.

[102] Marti A. Hearst. *Search User Interfaces*. Cambridge University Press, 2009.

[103] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*, CSCW '00, page 241–250, New York, NY, USA, 2000. Association for Computing Machinery.

[104] Michael A. Heroux, Roscoe A. Bartlett, Vicki E. Howle, Robert J. Hoekstra, Jonathan J. Hu, Tamara G. Kolda, Richard B. Lehoucq, Kevin R. Long, Roger P. Pawlowski, Eric T. Phipps, Andrew G. Salinger, Heidi K. Thornquist, Ray S. Tuminaro, James M. Willenbring, Alan Williams, and Kendall S. Stanley. An overview of the trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, sep 2005.

[105] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 194–201, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[106] Daqing Hou and Lin Li. Obstacles in using frameworks and apis: An exploratory study of programmers' newsgroup discussions. In *2011 IEEE 19th International Conference on Program Comprehension*, pages 91–100, 2011.

[107] Amy Hurst, Scott E. Hudson, and Jennifer Mankoff. Dynamic detection of novice vs. skilled use without a task model. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 271–280, New York, NY, USA, 2007. Association for Computing Machinery.

[108] Amy Hurst, Jennifer Mankoff, and Scott E. Hudson. Understanding pointing problems in real world computing environments. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '08, page 43–50, New York, NY, USA, 2008. Association for Computing Machinery.

[109] Snap Inc. *Snap AR Studio*. `https://ar.snap.com/lens-studio`, 2015.

[110] Anthony Jameson. Adaptive interfaces and agents. In *The human-computer interaction handbook*, pages 459–484. CRC press, 2007.

[111] Pamela Jennings. Reflections on interdisciplinary team development for the design of a platform for computer supported collaborative play. *Digital Creativity*, 15(4):209–222, December 2004.

[112] Ellen Jiang, Edwin Toh, Alejandra Molina, Kristen Olson, Claire Kayacik, Aaron Donsbach, Carrie J Cai, and Michael Terry. Discovering the syntax and strategies of natural language programming with generative language models. In *CHI Conference on Human Factors in Computing Systems*. ACM, April 2022.

[113] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, IWAR '99, page 85, USA, 1999. IEEE Computer Society.

[114] Hannes Kaufmann and Dieter Schmalstieg. Mathematics and geometry education with collaborative augmented reality. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, SIGGRAPH '02, page 37–41, New York, NY, USA, 2002. Association for Computing Machinery.

[115] Judy Kay. Learner control. *User Modeling and User-Adapted Interaction*, 11(1/2):111–127, 2001.

[116] Caitlin Kelleher and Randy Pausch. Stencils-based tutorials: Design and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, page 541–550, New York, NY, USA, 2005. Association for Computing Machinery.

[117] Fred S. Keller. Goodbye teacher. *Journal of Applied Behavior Analysis*, pages 79–89, March 1968.

[118] Fred S. Keller. Ten years of personalized instruction. *Teaching of Psychology*, pages 4–9, October 1974.

[119] Annie Kelly, R. Benjamin Shapiro, Jonathan de Halleux, and Thomas Ball. *ARcadia: A Rapid Prototyping Platform for Real-Time Tangible Interfaces*, page 1–8. Association for Computing Machinery, New York, NY, USA, 2018.

[120] Diane Kelly. Scientific software development viewed as knowledge acquisition. *J. Syst. Softw.*, 109(C):50–61, nov 2015.

[121] Mik Kersten and Gail C. Murphy. Using task context to improve programmer productivity. In *Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, SIGSOFT '06/FSE-14, page 1–11, New York, NY, USA, 2006. Association for Computing Machinery.

[122] Kimia Kiani, Parmit K. Chilana, Andrea Bunt, Tovi Grossman, and George Fitzmaurice. "i would just ask someone": Learning feature-rich design software in the modern workplace. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–10, 2020.

[123] Kimia Kiani, George Cui, Andrea Bunt, Joanna McGrenere, and Parmit K. Chilana. Beyond "one-size-fits-all": Understanding the diversity in how software newcomers discover and make use of help resources. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–14, New York, NY, USA, 2019. Association for Computing Machinery.

[124] Ada S. Kim and Amy J. Ko. A pedagogical analysis of online coding tutorials. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '17, page 321–326, New York, NY, USA, 2017. Association for Computing Machinery.

[125] Juho Kim, Philip J. Guo, Carrie J. Cai, Shang-Wen (Daniel) Li, Krzysztof Z. Gajos, and Robert C. Miller. Data-driven interaction techniques for improving navigation of educational videos. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, page 563–572, New York, NY, USA, 2014. Association for Computing Machinery.

[126] 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*, CHI '14, page 4017–4026, New York, NY, USA, 2014. Association for Computing Machinery.

[127] Kangsoo Kim, Mark Billinghurst, Gerd Bruder, Henry Been-Lirn Duh, and Gregory F. Welch. Revisiting trends in augmented reality research: A review of the 2nd decade of ismar (2008–2017). *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2947–2962, 2018.

[128] Tae Soo Kim, DaEun Choi, Yoonseo Choi, and Juho Kim. Stylette: Styling the web with natural language. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[129] Tim Kindberg, John Barton, Jeff Morgan, Gene Becker, Debbie Caswell, Philippe Debaty, Gita Gopal, Marcos Frid, Venky Krishnan, Howard Morris, John Schettino, Bill Serra, and Mirjana Spasojevic. People, places, things: Web presence for the real world. In *Proceedings Third IEEE Workshop on Mobile Computing Systems and Applications*, pages 19–28, 2000.

[130] Amy J. Ko, Robin Abraham, Laura Beckwith, Alan Blackwell, Margaret Burnett, Martin Erwig, Chris Scaffidi, Joseph Lawrance, Henry Lieberman, Brad Myers, Mary Beth Rosson, Gregg Rothermel, Mary Shaw, and Susan Wiedenbeck. The state of the art in end-user software engineering. *ACM Comput. Surv.*, 43(3), apr 2011.

[131] Amy J. Ko, Robert DeLine, and Gina Venolia. Information needs in collocated software development teams. In *Proceedings of the 29th International Conference on Software Engineering*, ICSE '07, page 344–353, USA, 2007. IEEE Computer Society.

[132] Amy J. Ko and Brad A. Myers. Designing the whyline: a debugging interface for asking questions about program behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI04. ACM, April 2004.

[133] Amy J. Ko, Brad A. Myers, and Htet Htet Aung. Six learning barriers in end-user programming systems. In *Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing*, VLHCC '04, page 199–206, USA, 2004. IEEE Computer Society.

[134] Nicholas Kong, Tovi Grossman, Björn Hartmann, Maneesh Agrawala, and George Fitzmaurice. Delta: A tool for representing and comparing workflows. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 1027–1036, New York, NY, USA, 2012. Association for Computing Machinery.

[135] 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*. ACM, May 2021.

[136] Peter G. Krogh. "interactive rooms—augmented reality in an architectural perspective". In *Proceedings of DARE 2000 on Designing Augmented Reality Environments*, DARE '00, page 135–137, New York, NY, USA, 2000. Association for Computing Machinery.

[137] Annakaisa Kultima and Kati Alha. "hopefully everything i'm doing has to do with innovation": Games industry professionals on innovation in 2009. In *2010 2nd International IEEE Consumer Electronics Society's Games Innovations Conference*, pages 1–8, 2010.

[138] Marianne LaFrance. The quality of expertise: Implications of expert-novice differences for knowledge acquisition. *SIGART Bull.*, (108):6–14, apr 1989.

[139] Benjamin Lafreniere, Andrea Bunt, John S. Whissell, Charles L. A. Clarke, and Michael Terry. Characterizing large-scale use of a direct manipulation application in the wild. In *Proceedings of Graphics Interface 2010*, GI '10, page 11–18, CAN, 2010. Canadian Information Processing Society.

[140] Benjamin Lafreniere, Tovi Grossman, and George Fitzmaurice. Community enhanced tutorials: Improving tutorials with multiple demonstrations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 1779–1788, New York, NY, USA, 2013. Association for Computing Machinery.

[141] Karim R Lakhani and Eric von Hippel. How open source software works: "free" user-to-user assistance. *Research Policy*, 32(6):923–943, June 2003.

[142] Thomas D. LaToza, Gina Venolia, and Robert DeLine. Maintaining mental models: A study of developer work habits. In *Proceedings of the 28th International Conference on Software Engineering*, ICSE '06, page 492–501, New York, NY, USA, 2006. Association for Computing Machinery.

[143] Felix Lauber, Claudius Böttcher, and Andreas Butz. Papar: Paper prototyping for augmented reality. In *Adjunct Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, AutomotiveUI '14, page 1–6, New York, NY, USA, 2014. Association for Computing Machinery.

[144] David Ledo, Steven Houben, Jo Vermeulen, Nicolai Marquardt, Lora Oehlberg, and Saul Greenberg. Evaluation strategies for HCI toolkit research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, April 2018.

[145] Gun A. Lee, Gerard J. Kim, and Mark Billinghurst. Immersive authoring: What you experience is what you get (wyxiwyg). *Commun. ACM*, 48(7):76–81, jul 2005.

[146] Gun A. Lee, Claudia Nelles, Mark Billinghurst, and Gerard Jounghyun Kim. Immersive authoring of tangible augmented reality applications. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '04, page 172–181, USA, 2004. IEEE Computer Society.

[147] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. Pronto: Rapid augmented reality video prototyping using sketches and enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery.

[148] Gilly Leshed, Eben M. Haber, Tara Matthews, and Tessa Lau. Coscripter: Automating & sharing how-to knowledge in the enterprise. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 1719–1728, New York, NY, USA, 2008. Association for Computing Machinery.

[149] Golan Levin. Essay for creative code. `https://www.flong.com/archive/texts/essays/essay_creative_code/index.html`, Accessed: 2023-11-11.

[150] Hongwei Li, Zhenchang Xing, Xin Peng, and Wenyun Zhao. What help do developers seek, when and how? In *2013 20th Working Conference on Reverse Engineering (WCRE)*, pages 142–151, 2013.

[151] Jingyi Li, Sonia Hashim, and Jennifer Jacobs. What we can learn from visual artists about software development. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[152] Wei Li, Tovi Grossman, and George Fitzmaurice. *GamiCAD: A Gamified Tutorial System for First Time Autocad Users*, page 103–112. Association for Computing Machinery, New York, NY, USA, 2012.

[153] Wei Li, Justin Matejka, Tovi Grossman, Joseph A. Konstan, and George Fitzmaurice. Design and evaluation of a command recommendation system for software applications. *ACM Trans. Comput.-Hum. Interact.*, 18(2), jul 2011.

[154] Henry Lieberman, Fabio Paternò, Markus Klann, and Volker Wulf. End-user development: An emerging paradigm. In *End User Development*, 2006.

[155] Michael Xieyang Liu, Advait Sarkar, Carina Negreanu, Benjamin Zorn, Jack Williams, Neil Toronto, and Andrew D. Gordon. "what it wants me to say": Bridging the abstraction gap between end-user programmers and code-generating large language models. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery.

[156] Wei Lo, Jianwei Yin, Shuiguang Deng, Ying Li, and Zhaohui Wu. An extended matrix factorization approach for qos prediction in service selection. In *2012 IEEE Ninth International Conference on Services Computing*, pages 162–169, 2012.

[157] Zhicong Lu. Improving viewer engagement and communication efficiency within non-entertainment live streaming. In *The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, page 162–165, New York, NY, USA, 2019. Association for Computing Machinery.

[158] Walid Maalej, Rebecca Tiarks, Tobias Roehm, and Rainer Koschke. On the comprehension of program comprehension. *ACM Trans. Softw. Eng. Methodol.*, 23(4), sep 2014.

[159] Blair MacIntyre, Maribeth Gandy, Steven Dow, and Jay David Bolter. Dart: A toolkit for rapid design exploration of augmented reality experiences. *ACM Trans. Graph.*, 24(3):932, jul 2005.

[160] Wendy E. Mackay. Patterns of sharing customizable software. In *Proceedings of the 1990 ACM Conference on Computer-Supported Cooperative Work*, CSCW '90, page 209–221, New York, NY, USA, 1990. Association for Computing Machinery.

[161] Wendy E. Mackay. Triggers and barriers to customizing software. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, page 153–160, New York, NY, USA, 1991. Association for Computing Machinery.

[162] John Maloney, Mitchel Resnick, Natalie Rusk, Brian Silverman, and Evelyn Eastmond. The scratch programming language and environment. *ACM Trans. Comput. Educ.*, 10(4), nov 2010.

[163] Lena Mamykina, Bella Manoim, Manas Mittal, George Hripcsak, and Björn Hartmann. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2857–2866, New York, NY, USA, 2011. Association for Computing Machinery.

[164] Sampada Marathe and S. Shyam Sundar. What drives customization? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, May 2011.

[165] Diego Marcos, Don McCurdy, and Kevin Ngo. *A-Frame*. `https://aframe.io/`, Accessed: 2023-11-11.

[166] Nicolai Marquardt, Robert Diaz-Marino, Sebastian Boring, and Saul Greenberg. The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11. ACM, October 2011.

[167] Mangalagouri Masarakal. Improving expertise-sensitive help systems. 2010.

[168] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Ambient help. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2751–2760, New York, NY, USA, 2011. Association for Computing Machinery.

[169] Justin Matejka, Tovi Grossman, and George Fitzmaurice. Ip-qat: In-product questions, answers, & tips. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 175–184, New York, NY, USA, 2011. Association for Computing Machinery.

[170] Jonathan F. Mattanah, Michael W. Pratt, Philip A. Cowan, and Carolyn P. Cowan. Authoritative parenting, parental scaffolding of long-division mathematics, and children's academic competence in fourth grade. *Journal of Applied Developmental Psychology*, 26(1):85–106, January 2005.

[171] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. A dose of reality: Overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15. ACM, April 2015.

[172] Joanna McGrenere, Ronald M. Baecker, and Kellogg S. Booth. A field evaluation of an adaptable two-interface design for feature-rich software. *ACM Trans. Comput.-Hum. Interact.*, 14(1):3–es, may 2007.

[173] Meta. *Meta Spark Studio*. https://spark.meta.com/, 2017.

[174] Microsoft. *Microsoft Maquette*. https://learn.microsoft.com/en-us/windows/mixed-reality/design/maquette, Accessed: 2023-11-11.

[175] Emerson Murphy-Hill, Thomas Zimmermann, and Nachiappan Nagappan. Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? In *Proceedings of the 36th International Conference on Software Engineering*, ICSE 2014, page 1–11, New York, NY, USA, 2014. Association for Computing Machinery.

[176] Rabia Murtza, Stephen Monroe, and Robert J. Youmans. Heuristic evaluation for virtual reality systems. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 61(1):2067–2071, September 2017.

[177] Brad A. Myers. Visual programming, programming by example, and program visualization: a taxonomy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI86. ACM, April 1986.

[178] Brad A. Myers, Amy J. Ko, Thomas D. LaToza, and YoungSeok Yoon. Programmers are users too: Human-centered methods for improving programming tools. *Computer*, 49(7):44–52, 2016.

[179] Phivos Mylonas, David Vallet, Pablo Castells, Miriam FernÁndez, and Yannis Avrithis. Personalized information retrieval based on context and ontological knowledge. *Knowledge Engineering Review*, 23(1):73–100, mar 2008.

[180] Avanika Narayan, Ines Chami, Laurel Orr, Simran Arora, and Christopher Ré. Can foundation models wrangle your data?, 2022.

[181] Michael Nebeling, Katy Lewis, Yu-Cheng Chang, Lihan Zhu, Michelle Chung, Piaoyang Wang, and Janet Nebeling. *XRDirector: A Role-Based Collaborative Immersive Authoring System*, page 1–12. Association for Computing Machinery, New York, NY, USA, 2020.

[182] 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, page 1–13, New York, NY, USA, 2019. Association for Computing Machinery.

[183] Michael Nebeling, Janet Nebeling, Ao Yu, and Rob Rumble. *ProtoAR: Rapid Physical-Digital Prototyping of Mobile Augmented Reality Applications*, page 1–12. Association for Computing Machinery, New York, NY, USA, 2018.

[184] 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, October 2018.

[185] Cuong Nguyen and Feng Liu. Making software tutorial video responsive. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1565–1568, New York, NY, USA, 2015. Association for Computing Machinery.

[186] Tien T. Nguyen, Pik-Mai Hui, F. Maxwell Harper, Loren Terveen, and Joseph A. Konstan. Exploring the filter bubble: The effect of using recommender systems on content diversity. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, page 677–686, New York, NY, USA, 2014. Association for Computing Machinery.

[187] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, page 152–158, New York, NY, USA, 1994. Association for Computing Machinery.

[188] Jakob Nielsen. *Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier*, page 245–272. Academic Press, Inc., USA, 1994.

[189] Jakob Nielsen and Uffe Lyngbæk. Two field studies of hypermedia usability. In Ray McAleese and Catherine Green, editors, *Hypertext: State of the Art. Papers presented at the Hypertext 2 conference, York, 1989*, pages 64–72. Intellect Ltd., Oxford, 1989.

[190] Richard E. Nisbett and Timothy D. Wilson. Telling more than we can know: Verbal reports on mental processes. *Psychological Review*, 84(3):231–259, May 1977.

[191] Donald A. Norman. The design of everyday things: Revised and expanded edition. 2013.

[192] David G. Novick, Oscar D. Andrade, and Nathaniel Bean. The micro-structure of use of help. In *Proceedings of the 27th ACM International Conference on Design of Communication*, SIGDOC '09, page 97–104, New York, NY, USA, 2009. Association for Computing Machinery.

[193] David G. Novick, Oscar D. Andrade, Nathaniel Bean, and Edith Elizalde. Help-based tutorials. page 1–8, 2008.

[194] David G. Novick and Karen Ward. Why don't people read the manual? In *Proceedings of the 24th Annual ACM International Conference on Design of Communication*, SIGDOC '06, page 11–18, New York, NY, USA, 2006. Association for Computing Machinery.

[195] Paul F. Nunes and Ajit Kambil. Personalization? no thanks. *Harvard business review*, 79:32–34, 04 2001.

[196] Reinhard Oppermann and Helmut Simm. Adaptability: user-initiated individualization. 1994.

[197] Stack Overflow. *2021 Developer Survey*. `https://insights.stackoverflow.com/survey/2021`, 2021.

[198] Cécile L. Paris. Tailoring object descriptions to a user's level of expertise. *Comput. Linguist.*, 14(3):64–78, sep 1988.

[199] Ahmed Patel, Liu Na, Rodziah Latih, Christopher Wills, Zarina Shukur, and Rabia Mulla. A study of mashup as a software application development technique with examples from an end-user programming perspective. *Journal of Computer Science*, 6(12):1406–1415, Nov 2010.

[200] Amy Pavel, Floraine Berthouzoz, and Björn Hartmann. Browsing and analyzing the command-level structure of large collections of image manipulation tutorials. 2013.

[201] Yuanzhe Peng. A survey on modern recommendation system based on big data, 2022.

[202] Perkinscoie. *Augmented and Virtual Reality Survey Report*. https://www.perkinscoie.com/en/news-insights/xr-industry-insider-2022-arvrxr-report.html, 2022.

[203] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. Pause-and-play: Automatically linking screencast video tutorials with applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 135–144, New York, NY, USA, 2011. Association for Computing Machinery.

[204] Luca Ponzanelli, Gabriele Bavota, Andrea Mocci, Massimiliano Di Penta, Rocco Oliveto, Mir Hasan, Barbara Russo, Sonia Haiduc, and Michele Lanza. Too long; didn't watch! extracting relevant fragments from software development video tutorials. In *Proceedings of the 38th International Conference on Software Engineering*, ICSE '16, page 261–272, New York, NY, USA, 2016. Association for Computing Machinery.

[205] Ivan Poupyrev, Desney S. Tan, Mark Billinghurst, Hirokazu Kato, Holger Regenbrecht, and Nobuji Tetsutani. Tiles: A mixed reality authoring interface. In *IFIP TC13 International Conference on Human-Computer Interaction*, 2001.

[206] Anna-Katharina Praetorius, Gerlinde Lenske, and Andreas Helmke. Observer ratings of instructional quality: Do they fulfill what they promise? *Learning and Instruction*, pages 387–400, 2012.

[207] Casey Reas and Ben Fry. Processing. https://processing.org/, Accessed: 2023-11-11.

[208] Marc Rettig. Nobody reads documentation. *Commun. ACM*, 34(7):19–24, jul 1991.

[209] John Rieman. A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction*, 3(3):189–218, sep 1996.

[210] Martin P. Robillard. What makes apis hard to learn? answers from developers. *IEEE Software*, 26, 2009.

[211] Martin P. Robillard and Robert Deline. A field study of api learning obstacles. *Empirical Softw. Engg.*, 16(6):703–732, dec 2011.

[212] Shahed Anzarus Sabab, Adnan Khan, Parmit K. Chilana, Joanna McGrenere, and Andrea Bunt. An automated approach to assessing an application tutorial's difficulty. In *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 1–10, 2020.

[213] Caitlin Sadowski, Kathryn T. Stolee, and Sebastian Elbaum. How developers search for code: A case study. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ESEC/FSE 2015, page 191–201, New York, NY, USA, 2015. Association for Computing Machinery.

[214] Akira Sano. An application for creating full-scale augmented reality content without 3d modeling skills. In *Proceedings of the 2011 ACM Symposium on The Role of Design in UbiComp Research & Practice*, RDURP '11, page 19–24, New York, NY, USA, 2011. Association for Computing Machinery.

[215] Christopher Scaffidi, Mary Shaw, and Brad Myers. Estimating the numbers of end users and end user programmers. In *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05)*, pages 207–214, 2005.

[216] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 253–260, New York, NY, USA, 2002. Association for Computing Machinery.

[217] Clinton B. Schertzer and Susan M. B. Schertzer. Student satisfaction and retention: A conceptual model. *Journal of Marketing for Higher Education*, 14(1):79–91, November 2004.

[218] Gregor Schiele, Marcus Handte, and Christian Becker. Pervasive computing middleware. In *Handbook of Ambient Intelligence and Smart Environments*, pages 201–227. Springer US, 2010.

[219] Avi Segal, Ziv Katzir, Ya'akov Gal, Guy Shani, and Bracha Shapira. Edurank: A collaborative filtering approach to personalization in e-learning. In *EDM*, 2014.

[220] Judith Segal. When software engineers met research scientists: A case study. *Empirical Software Engineering*, 10(4):517–536, October 2005.

[221] Hartmut Seichter, Julian Looser, and Mark Billinghurst. Composar: An intuitive tool for authoring ar applications. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '08, page 177–178, USA, 2008. IEEE Computer Society.

[222] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 210–217, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[223] Ben Shneiderman. Promoting universal usability with multi-layer interface design. *SIGCAPH Comput. Phys. Handicap.*, (73–74):1–8, jun 2002.

[224] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *Interactions*, 4(6):42–61, nov 1997.

[225] Valerie J Shute and Joseph Psotka. Intelligent tutoring systems: Past, present, and future. Technical report, ARMSTRONG LAB BROOKS AFB TX HUMAN RESOURCES DIRECTORATE, 1994.

[226] Jonathan Sillito, Gail C. Murphy, and Kris De Volder. Asking and answering questions during a programming change task. *IEEE Transactions on Software Engineering*, 34(4):434–451, 2008.

[227] Vandana Singh, Michael B. Twidale, and David M. Nichols. Users of open source software - how do they get help? In *2009 42nd Hawaii International Conference on System Sciences*, pages 1–10, 2009.

[228] Vandana Singh, Michael B. Twidale, and Dinesh Rathi. Open source technical support: A look at peer help-giving. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 6, pages 118c–118c, 2006.

[229] Ankita Nandkishor Sontakke, Manasi Patwardhan, Lovekesh Vig, Raveendra Kumar Medicherla, Ravindra Naik, and Gautam Shroff. Code summarization: Do transformers really understand code? In *Deep Learning for Code Workshop*, 2022.

[230] Maximilian Speicher, Brian D. Hall, and Michael Nebeling. What is mixed reality? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–15, New York, NY, USA, 2019. Association for Computing Machinery.

[231] 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. *Proceedings of the ACM on Human-Computer Interaction*, 2(EICS), jun 2018.

[232] Maximilian Speicher and Michael Nebeling. *GestureWiz: A Human-Powered Gesture Design Environment for User Interface Prototypes*, page 1–11. Association for Computing Machinery, New York, NY, USA, 2018.

[233] Maximilian Speicher, Kristina Tenhaft, Simon Heinen, and Harry Handorf. Enabling industry 4.0 with holobuilder. 09 2015.

[234] Patrick Stacey and Joe Nandhakumar. A temporal perspective of the computer game development process. *Information Systems Journal*, 19(5):479–497, August 2009.

[235] Suzi Stephenson. *TIGA Survey Reveals that Unity 3D Engine Dominates the UK Third Party Engine Market.* `https://tiga.org/news/tiga-survey-reveals-that-unity-3d-engine-dominates-the-uk-third-party-engine-market`, 2019.

[236] Stepico. *Why is Unity the Best Game Engine: Pros and Cons.* `https://stepico.com/blog/why-is-unity-the-best-game-engine-pros-and-cons/`, 2021.

[237] Simone Stumpf, Erin Sullivan, Erin Fitzhenry, Ian Oberst, Weng-Keen Wong, and Margaret Burnett. Integrating rich user feedback into intelligent user interfaces. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, page 50–59, New York, NY, USA, 2008. Association for Computing Machinery.

[238] Lucy Suchman. Making work visible. *Commun. ACM*, 38(9):56–64, sep 1995.

[239] Tamara Sumner and Markus Stolze. *Evolution, Not Revolution: Participatory Design in the Toolbelt Era*, page 1–26. MIT Press, Cambridge, MA, USA, 1997.

[240] Alistair Sutcliffe and Brian Gault. Heuristic evaluation of virtual reality applications. *Interacting with Computers*, 16(4):831–849, August 2004.

[241] TechCrunch. *How Unity built the world's most popular game engine.* `https://techcrunch.com/2019/10/17/how-unity-built-the-worlds-most-popular-game-engine/`, October 17, 2019.

[242] Unity Technologies. *Unity news board.* `https://unity.com/our-company/newsroom/unity-technologies-//finds-63-creatives-struggle-implementing-ar-advertising`, Accessed: 2023-11-11.

[243] Unity Technologies. *Unity Website.* `https://unity.com/`, Accessed: 2023-11-11.

[244] Dereck Toker, Ben Steichen, Matthew Gingerich, Cristina Conati, and Giuseppe Carenini. Towards facilitating user skill acquisition: Identifying untrained visualization users through eye tracking. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, IUI '14, page 105–114, New York, NY, USA, 2014. Association for Computing Machinery.

[245] Ted Tschang. Videogames as interactive experiential products and their manner of development. *International Journal of Innovation Management (ijim)*, 09:103–131, 03 2005.

[246] Priyan Vaithilingam, Tianyi Zhang, and Elena L. Glassman. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery.

[247] Janneke van de Pol, Monique Volman, Frans Oort, and Jos Beishuizen. The effects of scaffolding in the classroom: support contingency and student independent working time in relation to student achievement, task effort and appreciation of support. *Instructional Science*, 43(5):615–641, June 2015.

[248] Wim van Eck and Yolande Kolstee. The augmented painting: Playful interaction with multi-spectral images. In *2012 IEEE International Symposium on Mixed and Augmented Reality - Arts, Media, and Humanities (ISMAR-AMH)*, pages 65–69, 2012.

[249] Bogdan Vasilescu, Alexander Serebrenik, Prem Devanbu, and Vladimir Filkov. How social q&a sites are changing knowledge sharing in open source software communities. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, page 342–354, New York, NY, USA, 2014. Association for Computing Machinery.

[250] Elena Verdú, Luisa M. Regueras, María Jesús Verdú, Juan Pablo De Castro, and María Ángeles Pérez. An analysis of the research on adaptive learning: The next generation of e-learning. *WSEAS Trans. Info. Sci. and App.*, 5(6):859–868, jun 2008.

[251] Laton Vermette, Shruti Dembla, April Y. Wang, Joanna McGrenere, and Parmit K. Chilana. Social cheatsheet: An interactive community-curated information overlay for web applications. *Proceedings of the ACM on Human-Computer Interaction*, 1(CSCW), dec 2017.

[252] Laton Vermette, Joanna McGrenere, and Parmit K. Chilana. *Peek-through Customization: Example-Based In-Context Sharing for Learning Management Systems*, page 1155–1167. Association for Computing Machinery, New York, NY, USA, 2020.

[253] Vladimir Vukićević, Brandon Jones, and Boris Smus. *WebXR*. `https://en.wikipedia.org/wiki/WebXR`, Accessed: 2023-11-11.

[254] Lev S. Vygotskiĭ and Michael Cole. Mind in society : the development of higher psychological processes. 1978.

[255] April Y. Wang, Ryan Mitts, Philip J. Guo, and Parmit K. Chilana. Mismatch of expectations: How modern learning resources fail conversational programmers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA, 2018. Association for Computing Machinery.

[256] Wengran Wang, Archit Kwatra, James Skripchuk, Neeloy Gomes, Alexandra Milliken, Chris Martens, Tiffany Barnes, and Thomas Price. Novices' learning barriers when using code examples in open-ended programming. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, ITiCSE '21, page 394–400, New York, NY, USA, 2021. Association for Computing Machinery.

[257] Mark J. Weal, Eva Hornecker, Don G. Cruickshank, Danius T. Michaelides, David E. Millard, John Halloran, David C. De Roure, and Geraldine Fitzpatrick. Requirements for in-situ authoring of location based experiences. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, MobileHCI06. ACM, September 2006.

[258] Xiahua Wei, Wei Chen, and Kevin Zhu. Motivating user contributions in online knowledge communities: Virtual rewards and reputation. In *2015 48th Hawaii International Conference on System Sciences*, pages 3760–3769, 2015.

[259] Sarah Weir, Juho Kim, Krzysztof Z. Gajos, and Robert C. Miller. Learnersourcing subgoal labels for how-to videos. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '15, page 405–416, New York, NY, USA, 2015. Association for Computing Machinery.

[260] James V. Wertsch. From social interaction to higher psychological processes a clarification and application of vygotsky's theory. *Human Development*, 22(1):1–22, 1979.

[261] Brian T. Westphal, Frederick Harris, and Sami Fadali. Graphical programming: a vehicle for teaching computer problem solving. In *33rd Annual Frontiers in Education, 2003. FIE 2003.*, volume 2, pages F4C–19, 2003.

[262] Frank F. Xu, Bogdan Vasilescu, and Graham Neubig. In-ide code generation from natural language: Promise and challenges. *ACM Trans. Softw. Eng. Methodol.*, 31(2), mar 2022.

[263] Jurgen Zauner, Michael Haller, Alexander Brandl, and Werner Hartman. Authoring of a mixed reality assembly instructor for hierarchical structures. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pages 237–246, 2003.

[264] Chi Zhang and Guangzhi Zheng. Supporting adult learning: enablers, barriers, and services. In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education*, SIGITE/RIIT'13. ACM, October 2013.

[265] Feng Zhou, Henry Been-Lirn Duh, and Mark Billinghurst. Trends in augmented reality tracking, interaction and display: A review of ten years of ismar. In *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 193–202, 2008.

[266] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 493–502, New York, NY, USA, 2007. Association for Computing Machinery.

Appendix A

Study materials

In this appendix, I include study instruments and related documents that were used to recruit and collect data from participants in my studies.

A.1 Chapter 3 study materials

A.1.1 Pre-study Questionnaire

1. What is your age?
   - 18-24 years old
   - 25-34 years old
   - 35-44 years old
   - >45 years old
2. What is your gender?
   - Female
   - Male
   - Prefer not to specify
3. Background: Define the post secondary majors you have studied or you are studying:
   - Bachelor's degree: . . . .
   - Master's degree: . . . .
   - PhD degree: . . . .
   - Other: . . . .
4. What is your current position? (Sample answer: student, software developer, math teacher. Please do not specify where you work or study.)
   - . . . .

5. How many years of experience do you have in programming?
    - 0
    - 1-3
    - 4-7
    - 7-10
    - More than 10
6. List all of the languages you are familiar with, in order of experience/familiarity: (List most familiar first)
    - ....
7. Which AR/VR programming frameworks have you worked with? How much expertise do you have with each of them?
    - Unity: Expert Intermediate Beginner
    - Unreal engine: Expert Intermediate Beginner
    - WebVR: Expert Intermediate Beginner
    - ....: Expert Intermediate Beginner
    - ....: Expert Intermediate Beginner
    - Other: ....

A.1.2 Interview Questions

1. Please describe one of the AR/VR projects you've recently worked on.
    - What exactly did you make? What was it for (self-motivated, for a university project, work)? If it was a class project, did they provide you any instructions?
2. What motivated you to learn development in AR/VR?
3. How did you start to learn AR/VR development?
4. Can you tell me about the steps you took in your learning process?
    - What resources did you use?
    - How did you decide which language and tools to start developing with?
5. (If applicable) To what extent did your prior programming experience help you?
6. Can you remember any challenges or barriers you faced while working on your first AR/VR project? What were those? (By challenges I mean hardware related like headset configuration, troubleshooting, programming challenges, and modelling challenges.)
7. How did you try to resolve those challenges?
    - How did you know where to look for answers?

    - To what extent did you find those resources helpful?
    - Did your strategies work?

8. Did any of your AR/VR project experiences require you to do 3D modelling?
    - Were you familiar with 3D modelling?
9. (If applicable) How did you try to learn 3D modelling?
    - If you faced any problems, how did you try to find answers?
    - Was there any problem that you couldn't find an answer for?
10. When importing models you made separately in a 3D modelling environment, did you face any problems that you could not figure out where they came from?
11. To what extent were you satisfied with your results?
12. Did you want to continue doing AR/VR development? Why or why not?
13. What was your expectation from the modelling/programming interface and the learning resources? To what extent were you satisfied with these resources?
14. As novices are getting more motivated into AR/VR creation, what advice would you give them for getting started and being successful?
15. Is there anything else you would like to share with us?

A.2 Chapter 4 study materials

A.2.1 Pre-study Questionnaire

1. What is your age? 18-24 years old 25-34 years old 35-44 years old >45 years old
2. What is your gender? Female Male Prefer not to specify
3. Background: Define the post secondary majors you have studied or you are studying:
    - Bachelor's degree: ....
    - Master's degree: ....
    - Ph.D. degree: ....
    - Other: ....
4. What is your current position?
5. How many years of experience do you have in programming? 0 1-3 4-6 7-10 >10
6. List all of the programming languages you are familiar with, in order of experience/familiarity: (List most familiar first; if you have no programming experience, leave it blank) ....
7. Which VR programming frameworks you have worked with (if any)? How much expertise you have with each of them? ....

A.2.2 Tasks

Task 1 Scenario: A VR Simulation Experience

Armstrong school planning to run a competition with the goal of exposing students to new tech trends. As a starter, the school is running:

- A VR competition.
- Students are encouraged to be creative and try to build a Simulation experience through which they teach other students any topic they find interesting.
- The competition is going to happen in 7 weeks.

Imagine you are volunteering to encourage students to enter this competition and create a VR simulation experience. A student, Emily,has been assigned to you with the below information:

- She has never done any programming.
- She has intermediate 3D modeling experience.
- To encourage her to continue this path, her parents have allocated a $600 budget for her to buy a head-mounted display device but they don't know what their options are.

Your job is to help Emily find example tutorials and projects for developing a VR Simulation application. Use the Green tool from the home page and find 3 example projects that you think are most relevant for learning about creating VR simulation projects reflecting Emily's time and budget constraints and her programming and 3D modeling background.

Once you have finalized your 3 choices, please come back here and leave their URL in the empty fields. While you are interacting with the system and doing the task, please try to think aloud and keep us aware of your thought process.

Task 2 Scenario: An Object-Independent AR Experience

Imagine that you have started your own small business. Through some market research, you have come up with the idea of:

- Showing the objects you sell in an Augmented Reality (AR) experience.
- Through the augmented reality experience, you want your customers to be able to see the virtual 3D augmentation of the objects with any type of smartphone they own.
- The customers should be able to see the virtual objects, independent of the settings they are trying out the product (this is known as "Object-Independent AR").

- You want to sell your products as soon as possible and you have given yourself 3 weeks to wrap up creating your augmented reality experiences.

Now you want to start to know the possibilities of design and prepare a bit before starting to create this new content. Considering your, time, programming, and 3D modeling skills, use the Blue tool from the home page and find 3 example projects that you think will be most relevant and help you create an object-independent AR experience.

Once you have finalized your 3 choices, please come back here, click next, and leave their URL in the empty fields and answer some questions about your experience. While you are interacting with the system and doing the task, please try to think aloud and keep us aware of your thought process.

Task 3 Scenario: A VR 360 experience

Imagine that you are trying to become an entertainment content creator on Youtube to potentially increase your income. Through your research, you have found virtual reality (VR) experiences to be new, trending, and thriving so you have decided to create content in this area:

- Through your research, you have found that 360 experiences are the best choice.
- You want most of your audience to be able to view your 360 videos through Google Cardboard.
- You have a maximum of $100 and you want to see if you can buy a Google Cardboard headset or a similar product for yourself.
- You have given yourself 3 weeks to complete this project.

Now you want to start to know the possibilities of design and prepare a bit before starting to create this new content. Considering your budget, time, programming and 3D modeling level, use the Blue tool from the home page and find 3 example projects that you think will be most relevant and will help you create a VR 360 experience.

Once you finalized your 3 choices, please go back to the SurveyMonkey page, leave their URL in the empty fields. While you are interacting with the system and doing the task, please try to think aloud and keep us aware of your thought process.

Task 4 Scenario: An Object-Dependent AR Experience

Imagine that your friend, Matthew, is a teacher who volunteers and helps the Coquitlam high school in creating engaging educational content using new media. He wants students to be able to scan the pages of a book via the school-provided iPhones and see the virtual augmentation of the book contents through their phone screen. This would be an object-dependent augmented reality (AR) experience. Matthew has no time for research and he has asked you to suggest some tutorials and example projects related to creating object-dependent AR that suits his background:

- He is an intermediate programmer (has made 6 games that required coding)
- He is an expert 3D modeler (uses lots of CAD-related tools)
- The symposium is happening in 9 weeks
- The high school does not care about the budget as they want the best experience possible.

Use the Green tool from the home page and find 3 example projects that you think are most relevant and will help Matthew learn about creating AR object-dependent projects. These should be appropriate for his time constraint and programming and 3D modeling background.

Once you have finalized your 3 choices, please come back here, click next, and leave their URL in the empty fields and answer some questions about your experience. While you are interacting with the system and doing the task, please try to think aloud and keep us aware of your thought process.

Post-Task Questionnaire

1. I found the system easy to use
    - Strongly Agree
    - Agree
    - Neutral
    - Disagree
    - Strongly Disagree
2. With the criteria given to me in the task, I was able to easily locate relevant projects to get started.
    - Strongly Agree
    - Agree
    - Neutral
    - Disagree
    - Strongly Disagree
3. I found it demanding to evaluate the results shown in the user interface.
    - Strongly Agree
    - Agree
    - Neutral
    - Disagree
    - Strongly Disagree
4. I was frustrated while using the system:

- Strongly Agree
- Agree
- Neutral
- Disagree
- Strongly Disagree

5. I felt in control while using the system:
    - Strongly Agree
    - Agree
    - Neutral
    - Disagree
    - Strongly Disagree
6. On a scale of 1-10, how likely are you to reuse this system if you wanted to learn to create AR or VR applications?
    - ....
7. I felt confident about the projects that I selected:
    - Strongly Agree
    - Agree
    - Neutral
    - Disagree
    - Strongly Disagree
8. Answer verbally: Why did you find the 3 projects you picked relevant to this task?

A.2.3 Follow-up Interview Questions

1. What is your current process for finding new learning materials/tutorials?
2. What type of experience do you have in learning about AR or VR on your own, if any?
3. Now that you have seen both interfaces, which would you prefer to locate relevant materials?
4. Overall, how do you evaluate each of these platforms in terms of providing information for an AR/VR creation task relevant to your background?
5. Overall, how do you evaluate each of these platforms in terms of flexibility given different task criteria?
6. Overall, how confident are you with the results of each of these platforms in terms of flexibility given different search criteria?
7. If you were to start a (domain-related/hobby) task, would you consider using any of these two platforms? If yes, do you have any preferences? Why?
8. Do you have any recommendations for improvement?

A.3 Chapter 5 study materials

A.3.1 Pre-Study Questionnaire

1. Select your age group:
   - 18–24
   - 25–34
   - 35–44
   - 55–65
   - >65
2. What is your current position? (Please do not specify where you work or study.)

   (Sample answer: Student, software developer, math teacher)
3. How many years of experience do you have in programming?
   - 0
   - 1–3
   - 4–6
   - 7–10
   - More than 10
4. List all of the programming languages you are familiar with, in order of experience/familiarity (List the most familiar first).
5. Do you have any experience working with any AR/VR development frameworks? If so, please explain briefly.

A.3.2 Mid-Task Interview Questions

1. How have you progressed in your task so far?

   (Please describe your progress, including any milestones reached or objectives completed.)
2. What challenges did you face, if any?

   (Detail any obstacles encountered and how they impacted your work.)
3. What is your strategy for the rest of the session? For example, about the approach you are going to take for the rest of your development process and any particular learning resources you want to use or keep using?

   (Outline your plan for moving forward, including any changes to your methodology or resources.)
4. Describe a specific feature or functionality that you implemented so far in creating the AR experience. What was your thought process behind it? Please explain the steps you have taken so far to achieve it.

A.3.3 Post-Task Questionnaire

1. How was your experience in programming for a 3D and immersive experience compared to your previous non-3D development?
    - Much more difficult
    - Somewhat more difficult
    - About the same
    - Somewhat easier
    - Much easier
2. How effective were your efforts in finding online resources for your needs as a beginner in 3D/AR development?
    - Very effective
    - Somewhat effective
    - Neutral
    - Somewhat ineffective
    - Very ineffective
3. As a person with prior experience in programming, how would you characterize the transferability of your existing programming knowledge into creating an immersive 3D/AR experiences?
    - Extremely easy to transfer
    - Somewhat easy to transfer
    - Neither easy nor difficult to transfer
    - Somewhat difficult to transfer
    - Extremely difficult to transfer
4. As a person with prior experience in programming, I relied on my programming skills to troubleshoot and find solutions to technical problems. . .
    - Strongly disagree
    - Disagree
    - Neither agree nor disagree
    - Agree
    - Strongly agree

A.3.4 Post-Task Interview Questions

1. Show me what you created in today's session.
    (a) How was your overall experience in completing the task?
    (b) What do you think went well today as you were creating your first AR app?

(c) How did you plan and organize the development process of the AR/VR creation task?

2. Can you tell me about the most challenging parts of this task and any roadblocks you faced while completing your task?

3. What types of learning resources did you use during the creation process (if any)? For example, tutorials, videos, documentation, etc.

4. To what extent were you satisfied with the resources you used when creating your AR/VR application?

5. How did you approach the use of resources you previously mentioned in your development process? Were there any occasions in your development process in which you might have preferred using a help resource over the other options? Please explain.

6. What types of issues or bugs did you encounter during the development of interactive elements of the AR experience (if any)? How did you go about identifying and fixing them?

7. Overall, how does your experience in creating an AR experience for the first time compare to other kinds of software development that you may have done in the past?

8. While completing your task, did you find any capabilities missing from the tool you were using that might have helped you in doing your task better?

9. Reflecting on your experience, what would you do differently if you had the opportunity to start the AR development process again?

10. Is there anything else you would like to share about your experience?