

Teaching Advanced XR Development in the Vibe Coding Era

Yan Hu*

Gauri Vinsha Singh†

Blekinge Institute of Technology
Karlskrona, Sweden

ABSTRACT

Designing an “Advanced XR Development” course within a master’s program in XR presents unique challenges, as students already develop self-contained artifacts through multiple project-based courses covering topics such as interaction techniques, novel immersive interfaces, and immersive analytics. However, an analysis of the program structure reveals a critical gap: limited attention is given to integrating individual XR components into large, complex systems. We argue that this integration gap represents a key opportunity for advanced XR education. This paper explores how AI-assisted development (vibe coding) can reshape XR education by lowering implementation barriers and shifting the learning focus from building isolated components toward higher-level capstone competencies, including system design, integration, debugging, and evaluation. We propose the design of an “Advanced XR Development” course, presenting its learning objectives, teaching activities, and assessments. Rather than offering a finalized curriculum model, we position this work as a provocation for the community to discuss.

Index Terms: Course development, Vibe coding, XR development

1 INTRODUCTION

The demand for extended reality (XR) developers now surpasses the capacity of current educational pathways, resulting in a skills gap that extends beyond basic workforce shortages. Unlike conventional 2D application development, XR development is inherently more complex, requiring simultaneous consideration of 3D interaction, spatial environment, and multimodal input and output. Additionally, effective XR design depends on principles such as immersion, presence, and human perception, which do not have direct counterparts in traditional 2D interfaces [2]. Consequently, several universities have begun to establish specialized bachelor’s and master’s programs in XR and immersive technologies within the field of computer science.

On the other hand, generative artificial intelligence (GenAI) is transforming software development through the practice of vibe coding, in which developers use natural language to articulate their objectives and artificial intelligence (AI) systems generate and refine code accordingly. This development shifts the developer’s role from direct code writing to guiding and evaluating AI-generated output [8]. In educational contexts, vibe coding has the potential to increase accessibility to programming, accelerate prototyping, and enable students to concentrate on problem-solving and design. Nevertheless, it also introduces challenges related to understanding, verifying, and learning from AI-generated code [1].

This paper presents the course design for “Advanced XR Development” within an XR master’s program, emphasizing the integration of multiple XR components into a comprehensive XR system. To enhance students’ problem-solving skills in the GenAI era, the

course will incorporate the vibe coding approach. Section 2 outlines the master’s program and the challenges associated with developing this course. Section 3 details the initial course concept and its associated pedagogy. The learning objectives, teaching activities, and assignments are described in Sections 4 and 5. Section 6 discusses potential risks and raises open questions regarding course development, while Section 7 concludes the findings.

2 THE MASTER PROGRAM AND THE CHALLENGES OF THE COURSE

The Master’s program “XR and Intelligent Interactive Systems” at Blekinge Institute of Technology, Sweden, is an international program designed to prepare specialists in the design, development, and evaluation of immersive and intelligent digital systems. The program offers advanced knowledge and practical skills in XR and AI. The programme is a two-year Master’s programme comprising 120 ECTS credits. It consists of 12 courses, one big project course, and a final thesis. The curriculum covers key areas including 3D programming, interaction techniques, immersive human-machine interfaces, visual analytics, human factors, and applied artificial intelligence.

Among these courses, “Advanced XR Development” is positioned at the end of the first year and is designed to build upon the knowledge and skills acquired in previous courses. Rather than existing as an standalone course, the course is situated within a broader sequence of project-based learning activities. Several preceding and parallel courses focus on specific layers of XR system development, where students create self-contained artifacts targeting areas such as interaction techniques, novel human-computer interfaces, and immersive analytics, each representing a specialized research and development domain with its own methods, literature, and design practices [3] [4]. Therefore, the central design challenge of “Advanced XR Development” is to provide additional educational value without duplicating these existing experiences. Although the second-year “Big XR Project” course provides an opportunity for large-scale system development and integration, its primary emphasis is on delivering industry-oriented XR solutions.

3 INITIAL IDEA OF THE COURSE

Keeping the challenges in mind, “Advanced XR Development” course explicitly focuses on integration itself as a core competency, helping students understand how different XR components, technologies, and design considerations can be combined into large and complex immersive systems. Teaching XR system integration presents significant challenges, as limited project time is frequently allocated to low-level implementation tasks, including component setup and boilerplate code, which reduces opportunities for higher-level architectural analysis and design decisions [9].

The adoption of AI-assisted development addresses this issue by minimizing the effort required for component implementation. This enables students to concentrate on decomposition, interface design, subsystem integration, and evaluation. Empirical studies indicate that code-generation tools accelerate development, enhance motivation, and facilitate both implementation and exploration of unfamiliar technologies [5]. This transformation is particularly valuable

*e-mail: yan.hu@bth.se

†e-mail: gaurivinsha.singh@bth.se

for accessible XR development environments, ranging from sketching and rapid prototyping tools to high-level frameworks such as WebXR [6] and A-Frame [7], where generated components can be easily inspected and modified for integration into larger systems.

Consequently, the student's role shifts from that of a code author to a system architect. The primary learning outcome extends beyond producing a functional XR application to include articulating the rationale behind system architecture, integration decisions, trade-offs, and the critical evaluation of AI-generated solutions. Within this framework, vibe coding would include a systematic practice of specifying intent, validating outputs, and iteratively refining AI-generated components, rather than passively accepting generated code.

4 LEARNING OBJECTIVES

This course is designed to deepen and broaden students' knowledge of XR development by advancing from work with individual XR components to the design and implementation of large, complex XR systems. While previous courses address components such as interaction, rendering, visual analytics, and evaluation in isolation, this course emphasizes the architectural, technical, and methodological challenges that emerge when integrating these components within a unified real-time system. Additionally, the course introduces "vibe coding" as a professional approach for prototyping and implementing XR functionality. Students will practice expressing design intent through natural language or rapid prototyping tools, iteratively refining AI-generated code, and incorporating them into a larger code base in a controlled and transparent manner. Recognizing that AI-generated code may not be inherently correct, efficient, or secure, the course also develops students' abilities to critically evaluate both system architectures and AI-generated code, enabling them to make well-founded and defensible engineering decisions regarding the use of such tools. Upon completion, students are expected to be capable of independently planning, integrating, and quality-assuring complex XR applications in the GenAI era.

5 TEACHING AND LEARNING ACTIVITIES AND ASSESSMENT

Teaching consists of lectures, laboratory sessions, and supervised project work. The lectures introduce theoretical concepts related to XR system architecture, component integration, and AI-assisted development methodologies. During the laboratory sessions, students gain practical experience in integrating XR components and applying AI-assisted development workflows under supervision. The project work requires students, individually or in small groups, to design, implement, and evaluate a large and complex XR system, with individual assessment of learning outcomes.

To support students' learning autonomy and critical thinking skills, each student submits an individual reflection report analyzing their use of AI-generated code, including decisions on which components were adopted, modified, or rejected, and the reasoning behind these choices. The final project is assessed through both system submission and an oral presentation, where students demonstrate their technical understanding, design decisions, and ownership of the development process. This assessment approach ensures that AI-assisted development remains a practice of informed evaluation and refinement rather than the uncritical acceptance of code that students cannot explain.

6 RISKS AND OPEN QUESTIONS

In developing this course, we are confronted with two primary risks. The first concerns curriculum content, given that three to four related courses already focus on different components and utilize a project-based learning approach. The second risk involves the integration of AI-assisted development into XR education. This integration requires careful evaluation of potential risks, rather than

assuming that AI support will automatically enhance learning outcomes. A significant concern is the over-reliance on AI, which may lead to skill erosion by reducing students' engagement with higher-order cognitive processes.

Regarding the first risk, it is important to consider which additional contexts can be incorporated into the course given the specific circumstances, as well as learning from current similar courses from other universities. For the second risk, the proposed approach addresses this challenge by shifting assessment from code production to integration competencies. In this manner, AI may facilitate component implementation without supplanting the core cognitive skills being developed. In the GenAI era, problem-solving and computational thinking remain essential skills across disciplines. Students must learn to define problems, decompose complex challenges, iterate solutions, and debug effectively. Vibe coding tools may serve as effective learning environments for fostering the development of students' critical thinking skills.

7 CONCLUSION

This paper presents the design of an "Advanced XR Development" course within a master's program in XR, addressing an identified integration gap in the curriculum. While students frequently construct self-contained XR artifacts, they receive limited training in integrating these components into singular complex systems. The argument is made that integration should become the core competency, and that vibe coding enables this pedagogical shift by reducing low-level implementation effort. This approach transitions the student's role from code author to system architect and shifts assessment from code production to integration competencies. Whether AI support enhances or diminishes foundational skills remains an open empirical question. This course design is offered as a provocation for community discussion, with findings from the initial course iteration to be reported in future work.

ACKNOWLEDGMENTS

This research was funded partly by the Knowledge Foundation, Sweden, through the Human-Centered Intelligent Realities (HINTS) Profile Project (contract 20220068).

REFERENCES

- [1] How Can Vibe Coding Transform Programming Education? – Communications of the ACM, June 2025. 1
- [2] D. A. Bowman, S. Coquillart, B. Froehlich, M. Hirose, Y. Kitamura, K. Kiyokawa, and W. Stuerzlinger. 3d user interfaces: New directions and perspectives. *IEEE computer graphics and applications*, 28(6):20–36, 2008. 1
- [3] B. Ens, B. Bach, M. Cordeil, U. Engelke, M. Serrano, W. Willett, A. Prouzeau, C. Anthes, W. Büschel, C. Dunne, et al. Grand challenges in immersive analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2021. 1
- [4] C. Hillmann. *UX for XR: user experience design and strategies for immersive technologies*. Springer Nature, 2025. 1
- [5] M. Kazemitabaar, J. Chow, C. K. T. Ma, B. J. Ericson, D. Weintrop, and T. Grossman. Studying the effect of ai code generators on supporting novice learners in introductory programming. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pp. 1–23, 2023. 1
- [6] B. MacIntyre and T. F. Smith. Thoughts on the future of webxr and the immersive web. In *2018 IEEE international symposium on mixed and augmented reality adjunct (ISMAR-Adjunct)*, pp. 338–342. IEEE, 2018. 2
- [7] S. Neelakantam and T. Pant. Introduction to a-frame. In *Learning Web-based Virtual Reality: Build and Deploy Web-based Virtual Reality Technology*, pp. 17–38. Springer, 2017. 2
- [8] P. P. Ray. A review on vibe coding: Fundamentals, state-of-the-art, challenges and future directions. *Authorea Preprints*, 2025. 1
- [9] J. Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285, 1988. 1